
European Language Grid

Release 1

ELG Technical Team

May 13, 2021

CHAPTER 1: INTRODUCTION

1	CONTENTS	3
1.1	Introduction	3
1.2	Types of catalogue items	3
1.3	Using ELG	4
1.4	Browse the catalogue	4
1.5	Register/Sign in as a user	10
1.6	Use an LT service	13
1.7	Download a resource	15
1.8	Export a metadata record	16
1.9	Contributing to ELG	17
1.10	Register as a provider	17
1.11	Publication lifecycle	17
1.12	Contribute an ELG compatible service	19
1.13	Contribute an non-ELG compatible tool or service	31
1.14	Contribute a corpus/dataset	33
1.15	Contribute a model	42
1.16	Contribute a grammar	49
1.17	Contribute a lexical/conceptual resource	52
1.18	Contribute an organization	63
1.19	Contribute a project	69
1.20	Claim an item	74
1.21	Contribute via an external repository	75
1.22	Create catalogue items	75
1.23	Manage your items	81
1.24	Edit	84
1.25	Upload and Replace content files	86
1.26	Submit for publication	86
1.27	Create new versions	88
1.28	Copy records	88
1.29	Export metadata records	94
1.30	Delete catalogue items	94
1.31	Overview	94
1.32	Access the validation forms	95
1.33	Validate an ELG compatible LT service (at technical/metadata level)	96
1.34	Validate an LRT hosted in ELG (at technical/metadata level)	100
1.35	Validate an ELG compatible LT service or an LRT hosted in ELG at legal level	103
1.36	Validate a “metadata-only record”	104
1.37	ELG operations	107
1.38	Catalogue administration	107
1.39	Feedback and Helpdesk	108

1.40	Python SDK	109
1.41	Integrate a service into ELG using the python SDK	112
1.42	Main Classes	118
1.43	Catalog	119
1.44	Authentication	121
1.45	Entity	122
1.46	Corpus	123
1.47	Service	126
1.48	Benchmark	129
1.49	Pipeline	132
1.50	FlaskService	134
1.51	Metadata schema	135
1.52	Minimal version	136
1.53	Overview	209
1.54	Internal LT Service API specification	209
1.55	Public LT API specification	221
1.56	Terms of use	223
1.57	Publications and reports	223
1.58	Instructions for the registration of metadata records	226
1.59	Indices and tables	228
	Python Module Index	229
	Index	231

The [European Language Grid \(ELG\)](#) is a platform dedicated to **Language Technology (LT)**. It is developed in the framework of the [ELG project](#) and aims to evolve into the primary platform and virtual market place for all products, services and organizations active in the LT space in Europe. The platform can be used by all stakeholders to showcase, share and distribute their products, services, tools and data resources.

CONTENTS

1.1 Introduction

The [European Language Grid \(ELG\)](#) is a scalable platform that offers access to a **multitude of assets related to Language Technology (LT)**, including commercial and non-commercial cloud LT services for all European languages, data resources such as *models, datasets, lexica, terminologies, grammars*, as well as information on *LT-related projects, organizations, and groups*.

This manual aims to guide

- **consumers:** learn how to browse the ELG catalogue and find the language resources and technologies (LRTs) you need. Also find organizations and projects to connect with.
- **providers:** learn how to contribute your language resources and technologies to the ELG catalogue and how to host and provide access to your services via the ELG cloud.
- **moderators:** if you are part of the ELG technical team, learn how to monitor and validate submitted language resources and technologies.

The current version of the manual documents the second official release (R2) of the ELG platform, launched in March 2021. More functionalities are continuously added, and this manual keeps on being updated following the evolution of the ELG platform.

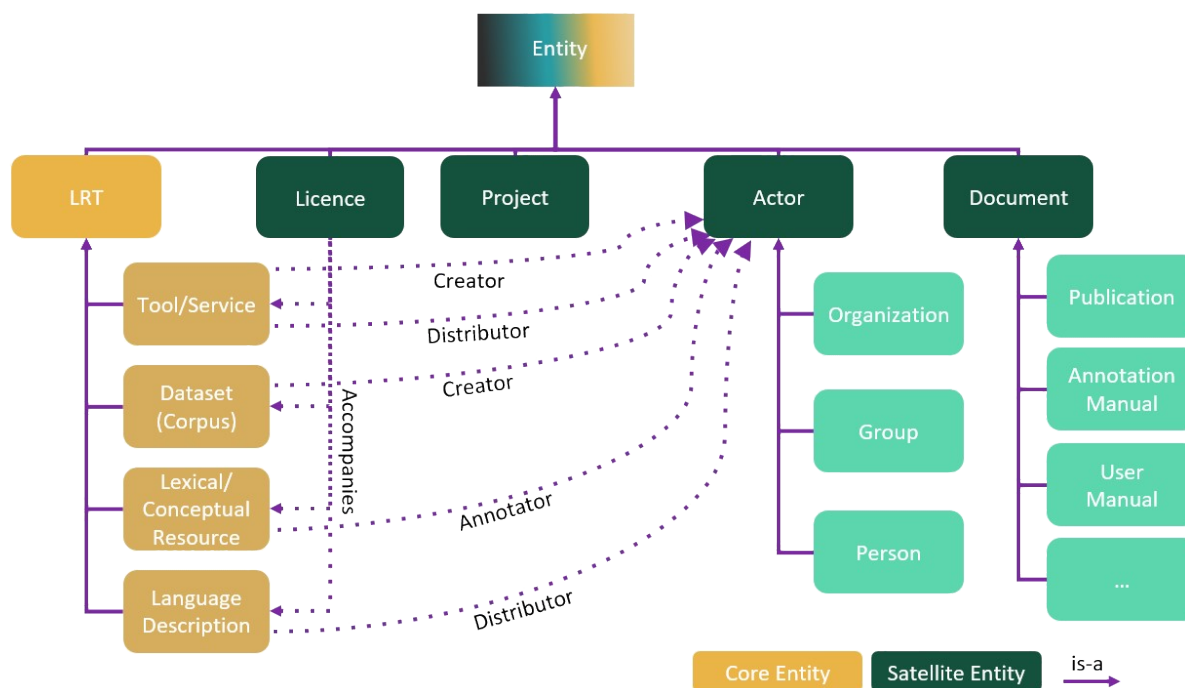
If you have any questions or want to share feedback, you can contact us through our [helpdesk](#).

1.2 Types of catalogue items

The [ELG catalogue](#) includes:

- **Language resources and technologies (LRTs)**, further classified into:
 - **tools & services:** services that run in the cloud, but also downloadable tools, source code, etc.,
 - **corpora** a.k.a datasets: collections of text documents, audio transcripts, audio and video recordings, etc.,
 - models & computational grammars, collectively referred to as **language descriptions**,
 - **lexical/conceptual resources**, comprising computational lexica, gazetteers, ontologies, term lists, etc.
- related activities and stakeholders from the wider area of language technology:
 - **projects** that have funded the development of LRTs or in which they have been deployed,
 - **organizations**, as well as **groups** and **persons** active in language technology in Europe.

The following diagram depicts the taxonomy of catalogue items:



Note: The current release doesn't display *groups* and *persons*. Moreover, *documents* and *licences* are described as separate entities, but they are not shown as main entities in the catalogue.

1.3 Using ELG

This chapter is for **consumers**, i.e. users of the European Language Grid who wish to explore the catalogue and use the provided language resources and technologies (LRTs). You will learn how to use the catalogue, how to try out and how to download LRTs.

1.4 Browse the catalogue

1.4.1 View catalogue

You can

- browse through the catalogue and see all the items,
- search for items using the free text bar,
- filter items by type, language, service function, licence, etc.
- search for items using the *Python SDK*.

By clicking on the name of an item, you can view its detailed description.

The screenshot shows the European Language Grid (ELG) interface. At the top, there is a logo for the European Language Grid and a navigation bar with links: Technologies, Resources, Community, Events, Documentation, and About ELG. Below the navigation bar is a search bar with the placeholder text "Search for services, tools, datasets, organizations..." and a "Search" button. To the left of the search results is a sidebar with a list of categories: Language resources & technologies, Service functions, Languages, Licences, Conditions of use for data, and Related entities. The search results display two items:

- "Le Monde Diplomatique" Arabic tagged corpus**: This corpus contains 102,960 vowelised, lemmatised and tagged words (58 texts from Le Monde Diplomatique Arabic, see also ELRA-W0036-04). To each text are associated 3 files : - raw text in Arabic, - vowelized text in ...
Keyword: corpus
Language: Arabic
Licences: ELRA-END-USER-COMMERCIAL-NOMEMBER-NONCOMMERCIALUSE-1.0 ...
101 views
- "Le Monde Diplomatique" Text corpus in Arabic**: Electronic archiving of "Le Monde Diplomatique" articles in Arabic from 2000. The corpus is available in HTML. Each HTML file contains one article. Number of articles available per year : • 2000: 61 articles (November a ...
28 views

1.4.2 View catalogue item

For each catalogue item, we display a set of descriptive and technical information (*metadata*), together with hyperlinks to supporting documentation and other useful material.

View tool/service

The following figure shows the catalogue item for a **tool/service**.

There are four tabs:


- **Overview**: contains the main metadata (e.g., description of basic features, function, input and output language(s) and data format(s), etc.), links to supporting documentation, contact details, resource providers, etc.
- **Download/Run**: includes the licensing terms under which the tool/service can be accessed, and relevant technical information (i.e., whether it can be downloaded and executed locally, is provided with source code, etc.).
- **Try out** (only for services running in ELG): you can provide a sample input and see the results output by the service. Depending on the type of the service, you can type in or paste some text, upload an audio file or record something, etc., and get the results rendered in a task-specific viewer.
- **Code samples** (only for services running in the cloud): you can use the code sample/template provided to test the service from the command line.

View data resource

The following figure shows the view page for a **corpus**.


There are two tabs:


- **Overview**: contains the main metadata: description, subclass, keyword(s), domain(s), etc., as well as links to supporting documentation, contact details, resource providers, etc. Some properties are grouped under the "parts" of a resource, each of which is characterised by the media type (text, audio, video, image). This allows us, for example, to describe a multimedia corpus of videos, their audio excerpts (in English), the transcriptions of the recordings (in an annotated format), and the subtitles in one or more languages (English and French, provided in plain text files), as a set of four distinct parts with the corresponding properties.



EUROPEAN
LANGUAGE
GRID

RELEASE 2

My grid 

Annika Grützner-Zahn 

Technologies


Resources


Community

Events

Documentation

About ELG

 Go to catalogue



Cogito Discover Named Entity Recognizer
ESI_NER
Version: 14.3.0

Functional service

ToolService

Overview

Download/Run

Try out

Code samples

Annotation of entities: People, Organizations, Places, Known concepts, Unknown concepts. And also tags: urls, mail addresses, phone numbers, addresses, dates, time, measures, money, percentage, file folder.

Keyword

multilingual

English

Spanish

German

French

NER

Named Entity Recognizer

Named Entity Recognition

Intended application

Named Entity Recognition

Input content resource

Language

English

Spanish; Castilian

German

French

Italian

Dutch; Flemish

Portuguese

Processing resource type

file

Data format

JSON

text/plain

HTML

XML

Character encoding

UTF-8

Media type

text

Function

Function

Named Entity Recognition

Language dependent

true

Output resource

Language

English

Processing resource type

file

Data format

JSON

Character encoding

UTF-8

Media type

text

Annotation type

Named entity

Person

Organization

Date

Location

Export

XML

Resource provider

Expert System

Website

Additional information

Landing page

Contact

Jose Manuel Gómez Pérez

Email

Documentation

Documented in

Cogito Discover Services Documentation

Resource creator


Expert System



Website

Evaluated

Evaluated: true


TRL: TRL9

**EUROPEAN
LANGUAGE
GRID**
RELEASE 2

[My grid](#)  [Annika Grützner-Zahn](#) 

[Technologies](#) [Resources](#) [Community](#) [Events](#) [Documentation](#) [About ELG](#)

[Go to catalogue](#)

**INTERA English-Slovene SVEZ ACQUIS Corpus**
Version: 1.0.0 (automatically assigned)

[Corpus](#)

[Overview](#) [Download](#) [Related LRTs](#)

The Slovene-English part of the INTERA corpus; written, domain specific (law) parallel subcorpus; 4MWs (2 MWs per language); TMX format.

Keyword
corpus

Domain
law

Intended application
machineTranslation

Corpus subclass
annotated corpus


Export
[XML](#)

Additional information
[Landing page](#)

Source of metadata record
[META-SHARE/ILSP](#)

Funded by
Integrated European language data Repository
Area
[Website](#)

Corpus part

 **TEXT**

Linguality type
bilingual

Multilinguality type
parallel

Language
[Slovenian](#) [English](#)

Modality type
written language

Original source description
The raw corpus comes from the SVEZ corpus provided by the Office of the Government of the Republic of Slovenia for European Affairs

Annotation
Annotation type
[Alignment](#)

Segmentation level
[sentence](#)

Annotation standoff
false

Annotation mode
automatic

Documentation

Documented in
Building Multilingual Terminological Resources

Documented in
Building parallel corpora for eContent professionals

Documented in
Language resources production models; the case of INTERA multilingual corpus and terminology

Documented in
[D5.2 - Report on the multilingual resources production](#)

Creation dates
01/01/2003 - 31/12/2004

Actual use

Used in application:
terminologyExtraction

Actual use details
nlpApplications

Ethics

Personal data included
false


Sensitive data included
false



1.4. Browse the catalogue

7

- **Download:** The second tab includes the licensing terms under which the resource can be accessed, and technical details on how it can be accessed (i.e., whether it can be downloaded, used via an interface, etc.), as well as details on formats and size. If the resource has been uploaded to ELG, you will also be able to download it directly; otherwise, you will be re-directed to the original access location.


The next two figures show the items for **lexical/conceptual resources** (lexica, terminologies, ontologies, etc.) and **language descriptions** respectively, with information tabs similar to those of corpora.

**EUROPEAN
LANGUAGE
GRID**
RELEASE 2

[My grid](#)  [Annika Grützner-Zahn](#) 

[Technologies](#) [Resources](#) [Community](#) [Events](#) [Documentation](#) [About ELG](#)

[Go to catalogue](#)

**CEPLeXicon**
Version: 1 (15/04/2015)
hosted at ELG


[Overview](#) [Download](#)

CEPLeXicon is a lexicon based on two different corpora of child speech – Santos corpus (Santos, 2006, Santos et al., 2014, see <http://www.dul.ul.pt/resources/546?lang=en>) and Freitas corpus (Freitas, 1997, Freitas et al. 2012). This lexicon results from the automatic tagging of the two corpora, using a tagger and the POS tag set produced in the research unit ANAGRAMA (Centro de Linguística da Universidade de Lisboa - CLUL) (Généreux, Hendrickx & Mendes, 2012). The automatic tagging was followed by a partial manual ... [Read More](#)

Keyword
lexicalconceptualresource

LCR subclass
lexicon

Lexical/ Conceptual resource part

 TEXT	Linguality type monolingual
	Language Portuguese
	Metalinguage Undetermined

Export
[XML](#)


Funded by
Complement Clauses in the Acquisition of Portuguese (PTDC/CLE-LIN/120897/2010)



Additional information
[Landing page](#)
ELRA-L0094 (other)
408-817-203-152-3 (ISLRN)

Source of metadata record
[ELRA Catalogue of Language Resources](#)

LCR details
Encoding level
syntax


Ethics
Personal data included
false
Sensitive data included
false

**EUROPEAN
LANGUAGE
GRID**
RELEASE 2

[My grid](#)  [Annika Grütznher-Zahn](#) 

[Technologies](#) [Resources](#) [Community](#) [Events](#) [Documentation](#) [About ELG](#)

[Go to catalogue](#)

**Finance English grammar**
Fin.en.grm
Version: 1.0.0 (automatically assigned)


[Overview](#) [Download](#)

Finance English abnf grammar, manually created. Created within the Portdial project

Keyword
languageDescription

LD subclass
Grammar

Language description part

 TEXT	Linguality type monolingual
	Language English
	Metalinguage Undetermined

Export
[XML](#)

Funded by
Portdial

Additional information
[Landing page](#)
[Landing page](#)

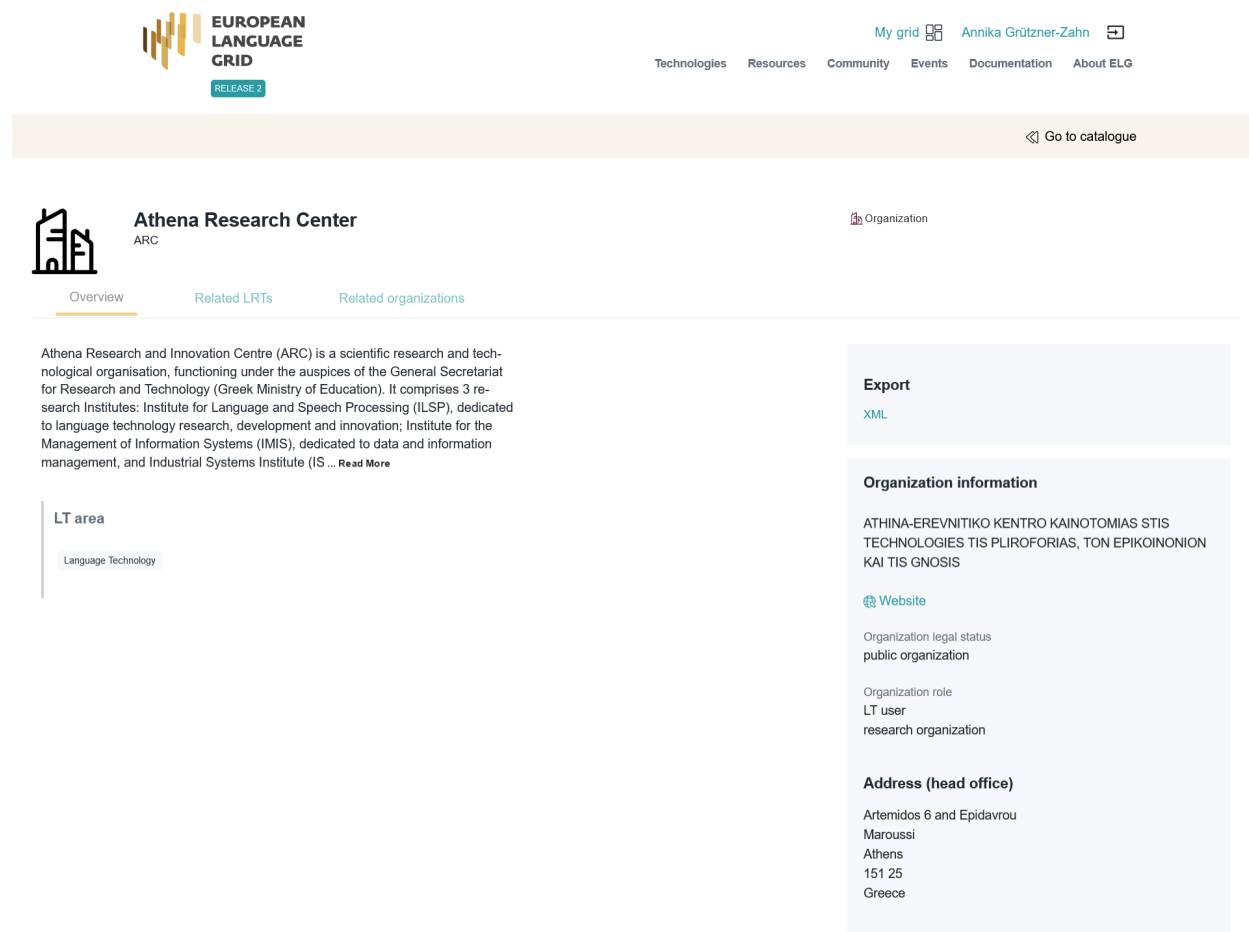
Source of metadata record
[META-SHARE/ILSP](#)

Grammar details
Encoding level
morphology

Ethics
Personal data included
false
Sensitive data included
false
null

View organization and project

The last two figures show respectively the catalogue items for an **organization** and a **project**, with contact details, funding information, links to resources, etc.



The screenshot displays the ELG website interface. At the top, the 'EUROPEAN LANGUAGE GRID' logo is visible, along with a 'RELEASE 2' badge. The navigation bar includes links for 'Technologies', 'Resources', 'Community', 'Events', 'Documentation', and 'About ELG'. A user profile for 'Annika Grützner-Zahn' is shown in the top right corner. Below the navigation bar, a 'Go to catalogue' button is present. The main content area features the 'Athena Research Center' (ARC) profile. The profile includes a logo, the name 'Athena Research Center', and the acronym 'ARC'. Below this, there are tabs for 'Overview', 'Related LRTs', and 'Related organizations'. The 'Overview' tab is selected, showing a detailed description of the ARC as a scientific research and technological organization. To the right of the description, there is an 'Export' section with a link to 'XML'. Below the description, there is a 'LT area' section with a link to 'Language Technology'. On the right side of the profile, there is a 'Organization information' section containing details about the organization's legal status, role, and address (head office).

Export

XML

Organization information

ATHINA-EREVNITIKO KENTRO KAINOTOMIAS STIS TECHNOLOGIES TIS PLIROFORIAS, TON EPIKOINONION KAI TIS GNOSIS

[Website](#)

Organization legal status
public organization

Organization role
LT user
research organization

Address (head office)

Artemidos 6 and Epidavrou
Maroussi
Athens
151 25
Greece

Some of the pages of organizations have a “Claim” button; for more information, see [Claim an item](#).

1.5 Register/Sign in as a user

Although you can browse the catalogue and access many resources without registration, a user account is required for, e.g., accessing resources with access restrictions, or making use of the try-out feature.

To register at ELG, click on the user icon at the top right of the page.

Then provide all the necessary information in the form that appears and agree to the terms of use.

After the registration, you will receive an email with a link to confirm your address; once confirmed, your account will be activated.

From now on, you can sign in to your account.


[My grid](#) [Annika Grützner-Zahn](#)
[Technologies](#) [Resources](#) [Community](#) [Events](#) [Documentation](#) [About ELG](#)
[Go to catalogue](#)


European Language Grid

ELG

[Project](#)
[Overview](#)
[Related LRTs](#)

With 24 official EU and many more additional languages, multilingualism in Europe and an inclusive Digital Single Market can only be enabled through Language Technologies (LTs). European LT business is dominated by thousands of SMEs and a few large players. Many are world-class, with technologies that outperform the global players. However, European LT business is also fragmented – by nation states, languages, verticals and sectors. Likewise, while much of European LT research is world-class, with results transferr ... [Read More](#)

Keyword

Language technology services
Multilingualism
Less-resourced languages

LT area

Language Technology

Export

[XML](#)

Project information

[Website](#)

Project start date
01/01/2019

Project end date
31/12/2021

Coordinator



German Research Center for Artificial Intelligence

[Website](#)

Participants

Athena Research Center

[Website](#)

Charles University

[Website](#)

The University of Sheffield

[Website](#)

SAIL LABS Technology

[Website](#)

Tilde

[Website](#)

University of Edinburgh

[Website](#)

Evaluation and Language Resources Distribution Agency

[Website](#)

Expert System

[Website](#)

Funder



European Commission
[Website](#)

Funding scheme category

IA

Funding country

European Union

Funding type

EU funds

Grant number: 825627

219378 (cordis)
Status
SIGNED

Related call
H2020-ICT-2018-2

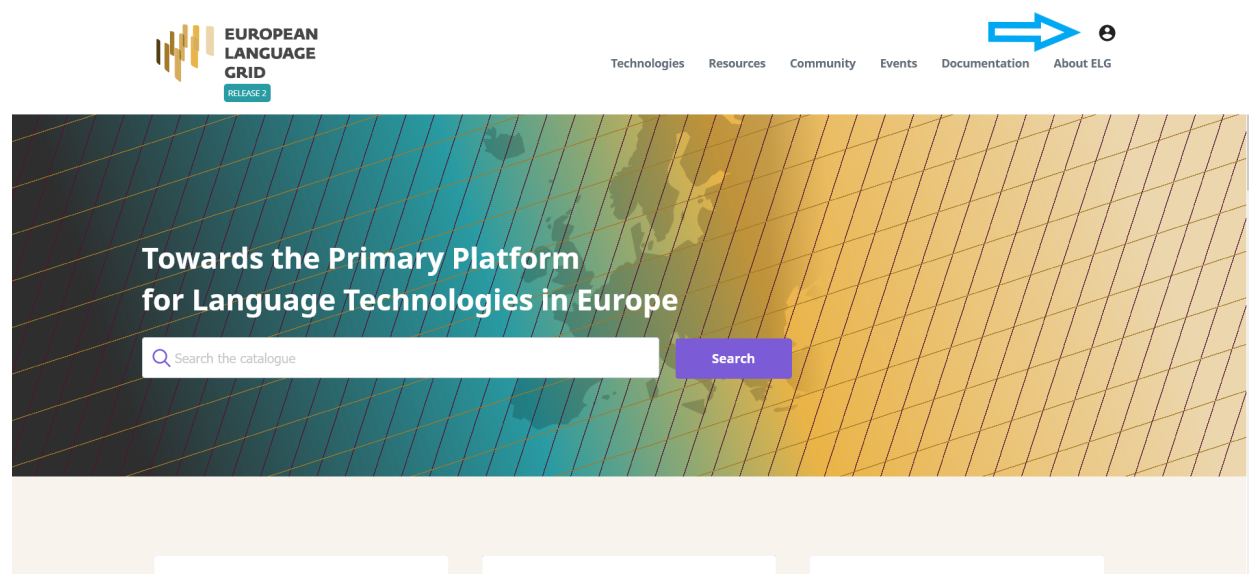
Related programme
H2020

Related subprogramme
ICT-29-2018

Amount
€7460206.25

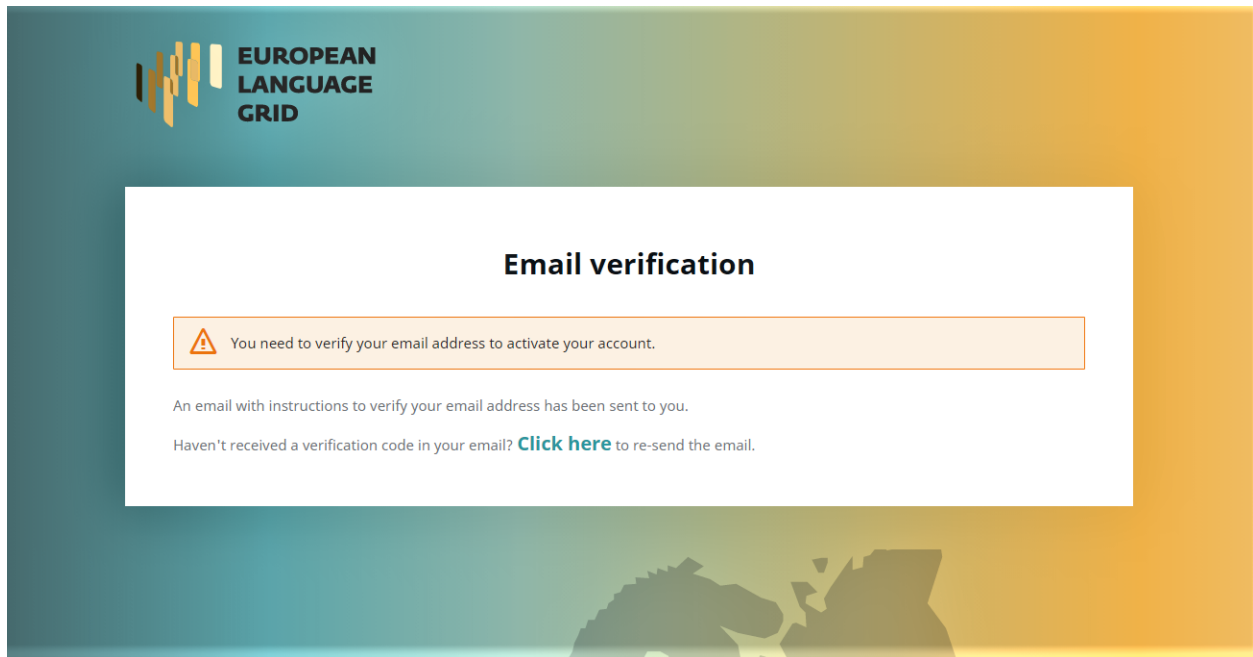
EC max contribution
€9999631.25





The screenshot shows the registration form on the ELG website. The form is titled "Register" and is set against a background with a stylized map of Europe and a grid pattern. The form fields are as follows:

- First name (*)**: A text input field.
- Last name(*)**: A text input field.
- Email(*)**: A text input field.
- Password (*)**: A text input field.
- Make sure you choose a strong password**: A note stating "Password must contain a mix of numbers, letters, and symbols. Minimum length of 8 characters."
- Confirm password (*)**: A text input field.
- Affiliation**: A text input field.
- I'm not a robot**: A checkbox and a reCAPTCHA widget.
- « Back to Login**: A link.
- Register**: A purple button.



1.6 Use an LT service

LT services that run in the cloud and follow ELG's specifications can be used via the ELG platform. You can do this via the try-out UI directly in the catalogue, via ELG's API or the *Python SDK*.

1.6.1 Try-out UI

The catalogue item of a service has a tab called **Try out**:

Try out:

Here you can provide a sample input and see the results output by the service. Depending on the type of the service, you can type in or paste some text, or upload or record audio, and get the results rendered in a task-specific viewer.


Note: In the current release, only registered users can try out services.

1.6.2 Call a service via the API

The catalogue item of a service has a tab called **Code samples**:


You can copy and modify the provided example commands to call the service from your command line.


More information on how to call and test services is given in the *Public LT API specification*.




**EUROPEAN
LANGUAGE
GRID**

RELEASE 2

[My grid](#)


[Annika Grützner-Zahn](#)


[Technologies](#)
[Resources](#)
[Community](#)
[Events](#)
[Documentation](#)
[About ELG](#)

[Go to catalogue](#)


If you want to send a processing request from your command line to ILSP Named Entity Recognizer for English you can use the following code sample/template

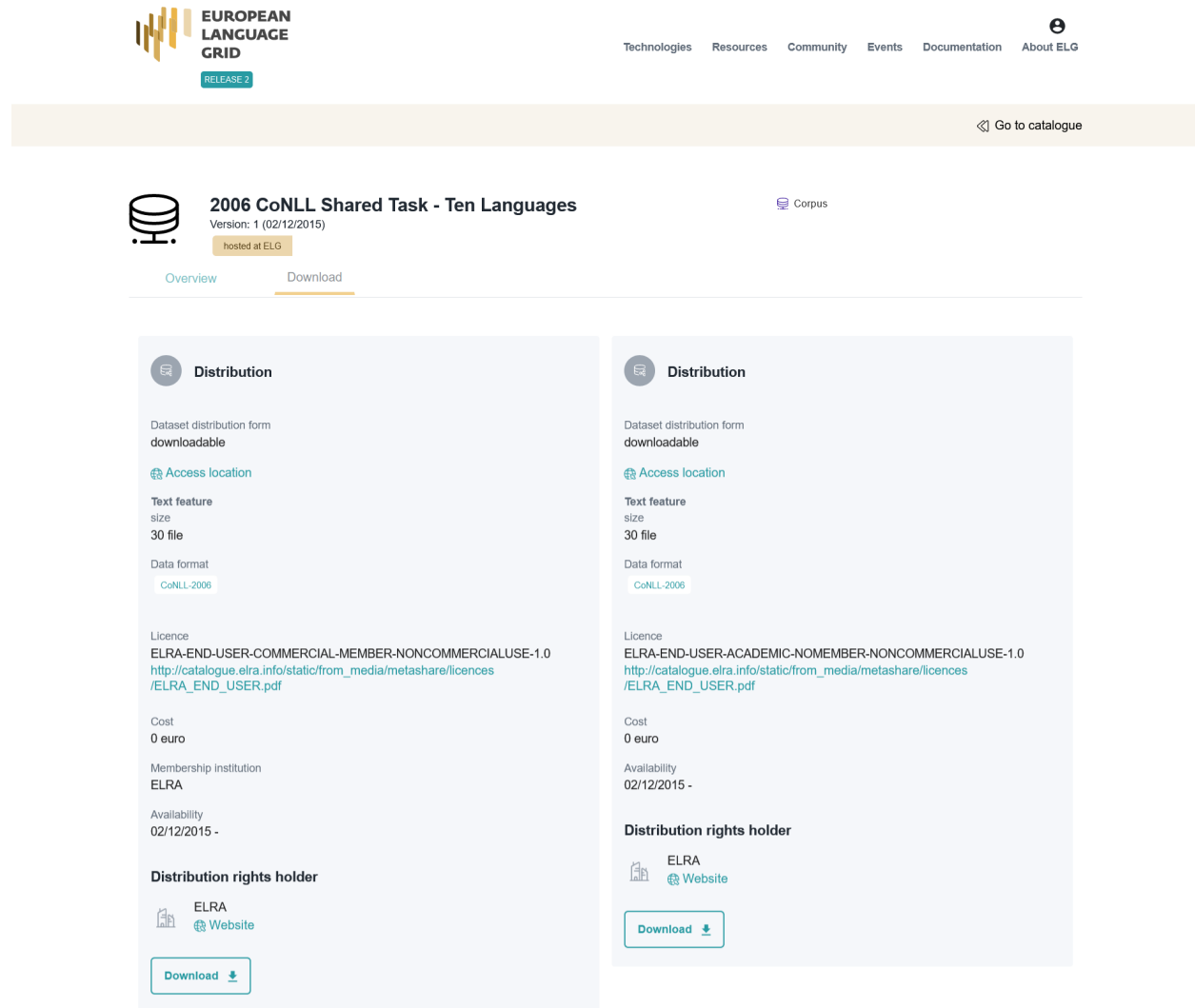
```
curl -H "Authorization: Bearer $TOKEN" -H "Content-Type: text/plain" --data-binary "@./path/somefile" https://live.european-language-grid.eu/execution/processText/lspnrcen
```

Token:

[illegible]

1.7 Download a resource

You can download a resource hosted at ELG (i.e., uploaded at ELG by the provider) through the Download tab, as shown below, or via the *Python SDK*.



The screenshot displays the European Language Grid (ELG) interface. At the top, the ELG logo is shown with the text 'EUROPEAN LANGUAGE GRID' and 'RELEASE 2'. Navigation links include Technologies, Resources, Community, Events, Documentation, and About ELG. A 'Go to catalogue' button is located in the top right. Below this, the '2006 CoNLL Shared Task - Ten Languages' resource is featured, with a version of 1 (02/12/2015) and a 'hosted at ELG' badge. The 'Download' tab is selected, showing the following details:

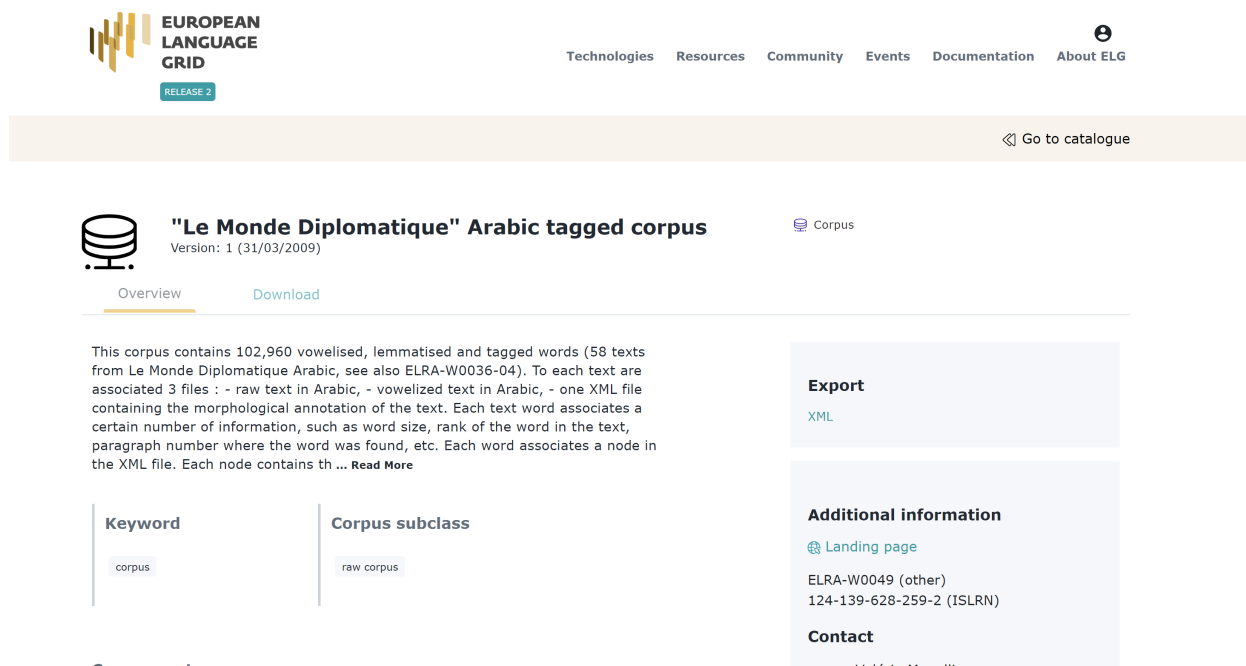
- Distribution:** Dataset distribution form downloadable. Access location is provided.
- Text feature:** size 30 file.
- Data format:** CoNLL-2006.
- Licence:** ELRA-END-USER-COMMERCIAL-MEMBER-NONCOMMERCIALUSE-1.0. The license URL is http://catalogue.elra.info/static/from_media/metashare/licences/ELRA_END_USER.pdf.
- Cost:** 0 euro.
- Membership institution:** ELRA.
- Availability:** 02/12/2015 -
- Distribution rights holder:** ELRA. A 'Download' button is present.

Downloading a resource is subject to the licensing terms under which it is provided. Resources with open access licences can be downloaded by all users. Depending on the terms, download may be restricted to registered users or require specific actions (e.g. accepting the terms):



1.8 Export a metadata record

You can export the metadata of all published catalogue items (except for resources marked as “work in progress”) in XML format, by clicking on the button as shown in the figure below.



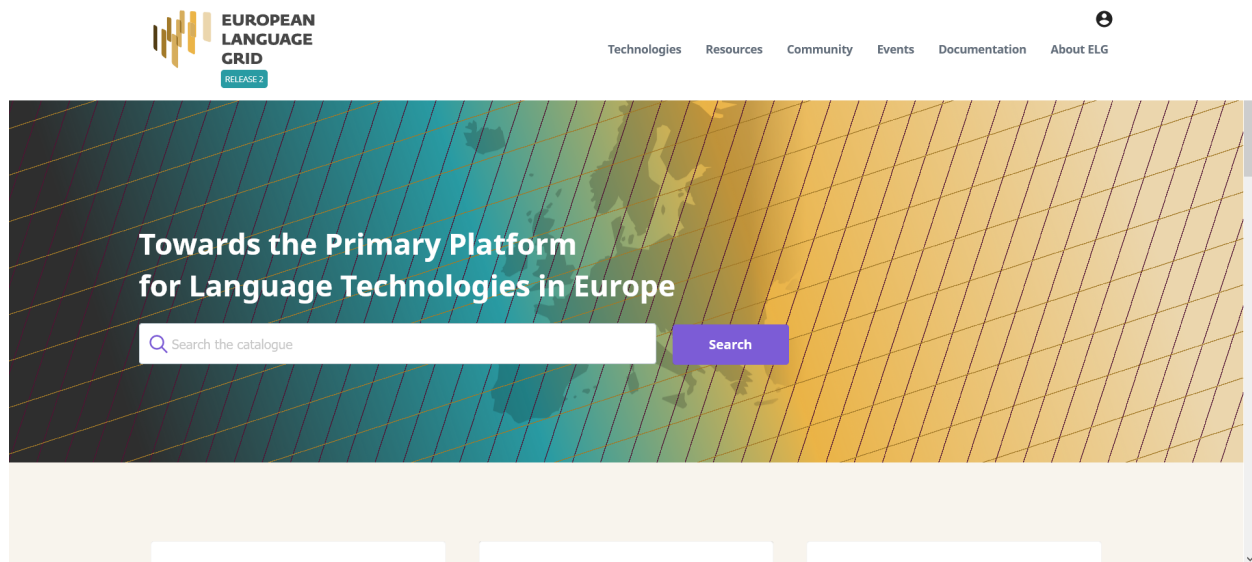
Note: Metadata records are licensed with a [CC-BY-NC 4.0](#) licence.

1.9 Contributing to ELG

This chapter is for **providers**, i.e. for users who wish to contribute *language resources and technologies* as well as information about *organizations* and *projects* to ELG. You will learn how to register as a provider and how to contribute each type of entity.

1.10 Register as a provider

In order to contribute to ELG you must have an *active account* at ELG. If you have one, sign in and click on your user name at top right of the page to access your profile.



From there you can request to become a “provider”.


When you do so you will be informed that your request will be reviewed and an answer will be provided to you soon.


1.11 Publication lifecycle

An item (i.e., a metadata record and, optionally, content files uploaded with it) contributed in ELG goes through a set of states before its publication on the catalogue (“ELG publication lifecycle”), as depicted in the following figure:

The states are:

- **new item:** A provider creates an item, by creating a metadata record through the interactive editor or by uploading a metadata file and, optionally, content files.
- **draft:** When using the interactive editor, the provider can save the metadata record, even without filling all mandatory elements; only compliance as to the data type of the elements is checked (e.g. elements that take URL must be filled in with the accepted pattern).
- **syntactically valid:** The metadata record complies with the ELG metadata schema and all mandatory elements are filled in. The provider can still continue to edit it until satisfied with the description and can then submit it for publication.




Annika Grützner-Zahn 

TechnologiesResourcesCommunityEventsDocumentationAbout ELG

User profile

User information



First name:
Annika

Last name:
Grützner-Zahn

Email:
[REDACTED]

Affiliation:
DFKI

Roles:
consumer


Change password


Edit profile

Have an amazing language tool?

ELG is primary platform for Language Technology in Europe. Your data resources, tools and technologies can be shared here and meet their users. Become a provider, and you will get acknowledgment in academic circles, increase your market, and improve the recognition of your resource. Find out more information [here](#).

Become a provider



Annika Grützner-Zahn 

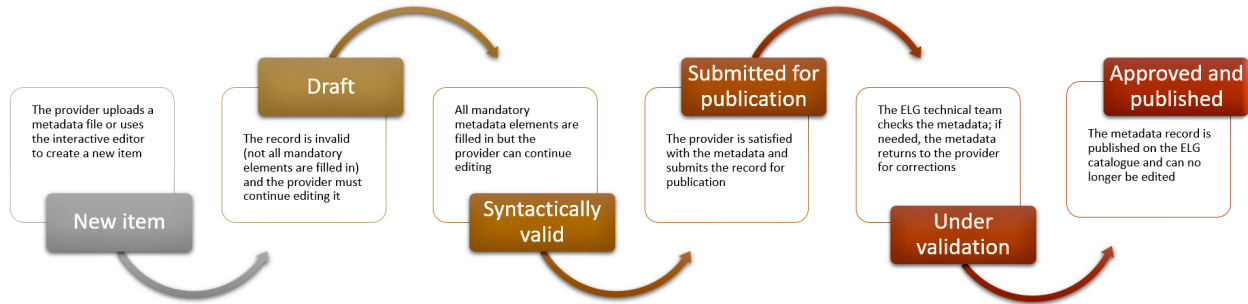
TechnologiesResourcesCommunityEventsDocumentationAbout ELG

Provider role requested succesfully!

We will review your request and get back to you as soon as we can. It might take a couple of days, meanwhile check out our [Technologies page](#).

18

Chapter 1. CONTENTS



- **submitted for publication:** The record is no longer editable. Depending on the item type and the source (see table below), the item is validated at the metadata, technical and legal level. The validation aims to check the consistency of the description and, where required, the technical compliance of the item to the ELG specifications; it doesn't include any qualitative evaluation. The validation is currently performed by the ELG consortium members. When validators identify a problem, they contact the provider for further information and may ask the provider to edit the metadata; in such cases, the status of the item is changed to "syntactically valid" and the provider is notified to make the appropriate amendments.
- **published:** When the validator(s) have approved an item, it is automatically visible via the ELG public catalogue. Once approved by the human validators and subsequently published, a metadata record cannot be edited any more.

Note: The "draft" state is available only for metadata records created via the interactive editor. Metadata files that are uploaded to the platform must be "syntactically valid".

The following table shows the validation operations foreseen for each item type / source of metadata.

Type of records	Validation type		
	Metadata	Technical	Legal
Harvested metadata	N/A	N/A	N/A
Metadata records uploaded by ELG admin	Yes	N/A	N/A
"Metadata only" records	Yes	N/A	N/A
ELG compatible services	Yes	Yes	Yes
LRTs uploaded (hosted) in ELG	Yes	Yes	Yes

"Metadata only" records are records for projects, organizations but also for LRTs that are not uploaded in ELG.

1.12 Contribute an ELG compatible service

This page describes how to contribute a language technology *service* to run on the cloud platform of the European Language Grid.

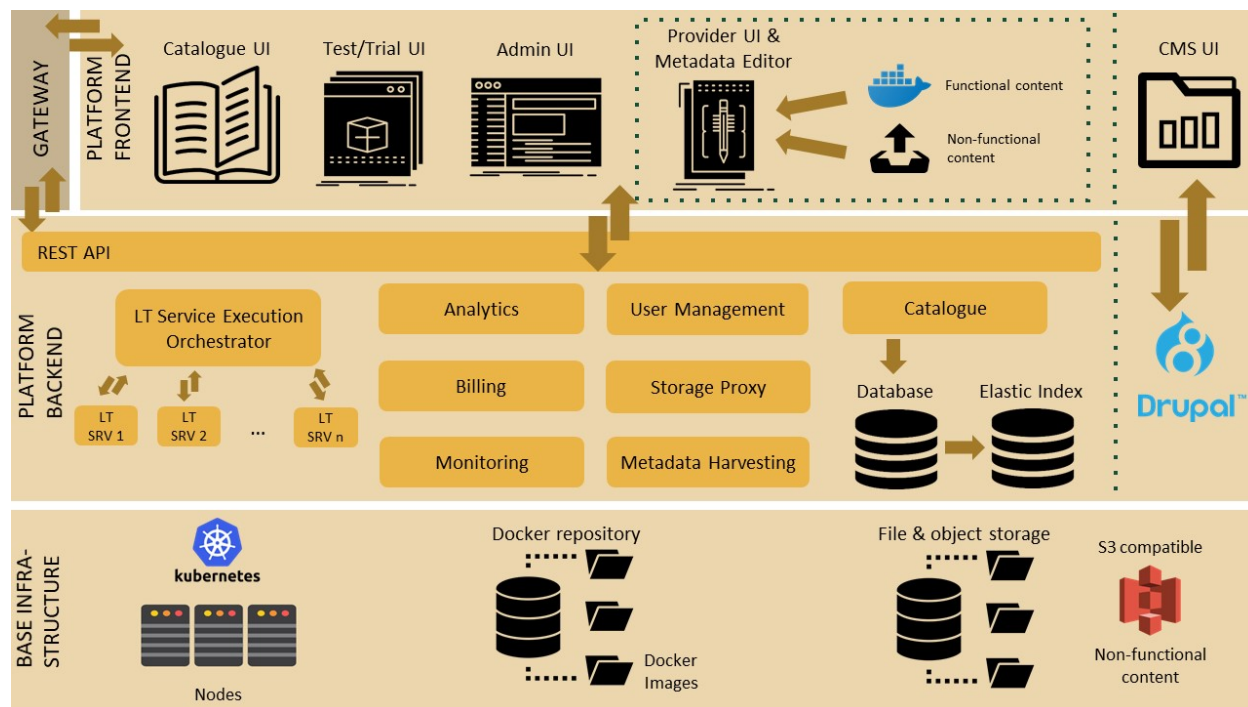
Currently, ELG supports the integration of tools/services that fall into one of the following broad categories:

- **Information Extraction (IE)** : Services that take text and annotate it with metadata on specific segments, e.g. *Named Entity Recognition (NER)*, the task of extracting persons, locations, and organizations from a given text.
- **Text Classification (TC)** : Services that take text and return a classification for the given text from a finite set of classes, e.g. *Text Categorization* which is the task of categorizing text into (usually labelled) organized categories.

- **Machine Translation (MT)** : Services that take text in one language and translate it into text in another language, possibly with additional metadata associated with each segment (sentence, phrase, etc.).
- **Automatic Speech Recognition (ASR)** : Services that take audio as input and produce text (e.g., a transcription) as output, possibly with metadata associated with each segment.
- **Text-to-Speech Generation (TTS)** : Services that take text as input and produce audio as output.

1.12.1 Overview: How an LT Service is integrated to ELG

An overview of the ELG platform is depicted below.



The following bullets summarize how LT services are deployed and invoked in ELG.

- All LT Services (as well as all the other ELG components) are deployed (run as containers) on a *Kubernetes (k8s)* cluster; k8s is a system for automating deployment, scaling, and management of containerised applications.
- All LT Services are integrated into ELG via the LT Service Execution Orchestrator/Server. This server exposes a **common public REST API (Representational state transfer)** used for invoking any of the deployed backend LT Services. The public API is used from ELG's Trial UIs that are embedded in the ELG Catalogue; it can also be invoked from the command line or any programming language; see [Use an LT service](#) section for more information. Some of the HTTP endpoints that are offered in the API are given below; for more information see [Public LT API specification](#).

Endpoint	Type	Consumes	Produces
https://{domain}/execution/processText/{ltServiceId}	POST	'application/json'	'application/json'
https://{domain}/execution/processText/{ltServiceId}	POST	'text/plain' or 'text/html'	'application/json'
https://{domain}/execution/processAudio/{ltServiceId}	POST	'audio/x-wav' or 'audio/wav'	'application/json'
https://{domain}/execution/processAudio/{ltServiceId}	POST	'audio/mpeg'	'application/json'

{domain} is 'live.european-language-grid.eu' and {ltServiceID} is the ID of the backend LT service. This ID is assigned/configured during registration; see section 3. *Register the service at ELG* - 'LT Service is deployed to ELG and configured' step.

Note: The REST API that is exposed from an LT Service X (see above) is for the communication between the LT Service Execution Orchestrator Server and X (ELG internal API - see *Internal LT Service API specification*).

- When the LT Service Execution Orchestrator receives a processing request for service X, it retrieves from the database X's k8s REST endpoint and sends a request to it. This endpoint is configured/specified during the registration process; see section 3. *Register the service at ELG* - 'LT Service is deployed to ELG and configured' step. When the Orchestrator gets the response from the LT Service, it returns it to the application/client that sent the initial call.

1.12.2 0. Before you start

- Please make sure that the service you want to contribute complies with our *terms of use*.
- Please make sure you have *registered* and been assigned the *provider role*.
- Please make sure that your service meets the technical requirements below, and choose one of the three integration options.

Technical requirements and integration options

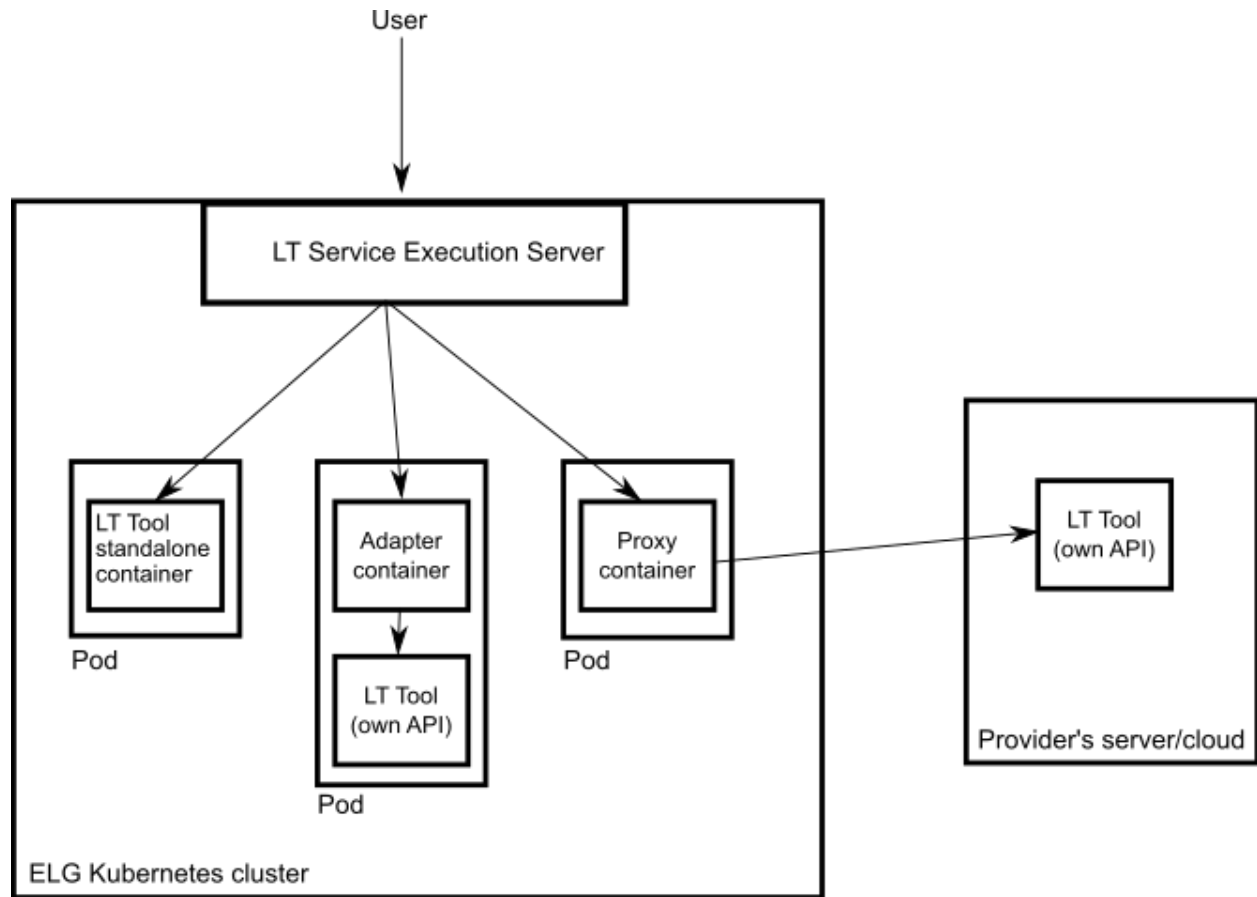
The requirements for integrating an LT tool/service to ELG are the following:

Expose an ELG compatible endpoint: You MUST create an application that exposes an HTTP endpoint for the provided LT tool(s). The application MUST consume (via the aforementioned HTTP endpoint) requests that follow the ELG JSON format, call the underlying LT tool and produce responses again in the ELG JSON format. For a detailed description of the JSON-based HTTP protocol (*ELG Internal LT API*) that you have to implement, see *the Internal LT API specification*.

Dockerisation: You MUST dockerise the application and upload the respective image(s) in a *Docker* Registry, such as *GitLab*, *DockerHub*, *Azure Container Registry* etc. You MAY select out of the three following options, the one that best fits your needs:

- **LT tools packaged in one standalone image:** One docker image is created that contains the application that exposes the ELG-compatible endpoint and the actual LT tool.
- **LT tools running remotely outside the ELG infrastructure:** For these tools, one *proxy* image is created that exposes one (or more) ELG-compatible endpoints; the proxy container communicates with the actual LT service that runs outside the ELG infrastructure.
- **LT tools requiring an adapter:** For tools that already offer an image that exposes a non-ELG compatible endpoint (HTTP-based or other), a second *adapter* image SHOULD be created that exposes an ELG-compatible endpoint and acts as proxy to the container that hosts the actual LT tool.

In the following diagram the three different options for integrating a LT tool are shown:



1.12.3 1. Dockerize your service

Build/Store Docker images

Ideally, the source code of your LT tool/service already resides on [GitLab](#) where a built-in [Continuous Integration \(CI\) Runner](#) can take care of building the image. GitLab also offers a [container registry](#) that can be used for storing the built image. For this, you need to add at the root level of your GitLab repository a `.gitlab-ci.yml` file as well as a `Dockerfile`, i.e., the recipe for building the image. [Here](#) you can find an example. After each new commit, the CI Runner is automatically triggered and runs the CI pipeline that is defined in `.gitlab-ci.yml`. You can see the progress of the pipeline on the respective page in GitLab UI (“CI / CD -> Jobs”); when it completes successfully, you can also find the image at “Packages -> Container Registry”.

Your image can also be built and tagged in your machine by running the `docker build` command. Then it can be uploaded (with `docker push`) to the GitLab registry, [DockerHub](#) (which is a public Docker registry) or any other Docker registry.

For instance, for [this](#) GitLab hosted project, the commands would be:

- `docker login registry.gitlab.com`

for logging in and be allowed to push an image

- `docker build -t registry.gitlab.com/european-language-grid/dfki/elg-jtok`

for building an image (locally) for the project - please note that before running `docker build` you have to download (clone) a copy of the project and be at the top-level directory (`elg-jtok`)

- `docker push registry.gitlab.com/european-language-grid/dfki/elg-jtok`

for pushing the image to GitLab.

In the following links you can find some more information on docker commands plus some examples:

- [Docker Command Line Interface](#).
- [Docker Tutorial from Stackify](#).

Dockerization of a Python-based LT tool

An example of a Python-based LT tool: [Python-based example](#).

Dockerization of a Java-based tool

A Spring Boot starter to make it as easy as possible to create ELG-compliant tools in Java is provided at: [ELG Spring Boot Starter](#).

1.12.4 2. Describe the service

Metadata overview

The service must be described according to the *ELG schema* and comply at least with the minimal version. The metadata elements that you need to provide for the service comprise a set of elements organized (for presentation purposes) into the following groups:

- *Minimal elements for all entities*
- *Minimal elements for all language resources and technologies*
- *Minimal elements for tools/services.*

You will find all the mandatory and recommended elements for all tools/services in the above sections. Please, note that for ELG integrated services, you **MUST** include in your metadata record a `SoftwareDistribution` component with the following elements:

- `SoftwareDistributionForm` (Mandatory): The medium, delivery channel or form (e.g., source code, API, web service, etc.) through which a software object is distributed. For ELG integrated services, use the value `http://w3id.org/meta-share/meta-share/dockerImage`.
- `dockerDownloadLocation` (Mandatory if applicable): A location where the the LT tool docker image is stored. Add the location from where the ELG team can download the docker image in order to test it.
- `serviceAdapterDownloadLocation` (Mandatory if applicable): he URL where the docker image of the service adapter can be downloaded from. Required only for ELG integrated services implemented with an adapter.
- `executionLocation` (Mandatory): A URL where the resource (mainly software) can be directly executed. Add here the REST endpoint at which the LT tool is exposed within the Docker image.
- `additionalHwRequirements` (Mandatory if applicable): A short text where you specify additional requirements for running the service, e.g. memory requirements, etc. The recommended format for this is: `'limits_memory: X limits_cpu: Y'`

Examples

Example 1: Information Extraction service

ANNIE's Named Entity Recognizer published at: <https://live.european-language-grid.eu/catalogue/#/resource/service/tool/512>

The Docker image for this service is stored at the GitLab registry.

```
<?xml version="1.0" encoding="UTF-8"?>
<ms:MetadataRecord xsi:schemaLocation="http://w3id.org/meta-share/meta-share/ ../
Schema/ELG-SHARE.xsd" xmlns:ms="http://w3id.org/meta-share/meta-share/" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance">
<ms:DescribedEntity>
  <ms:LanguageResource>
    <ms:entityType>LanguageResource</ms:entityType>
    <ms:resourceName xml:lang="en">GATE: English Named Entity Recognizer
  </ms:resourceName>
    <ms:resourceShortName xml:lang="en">annie-named-entity-recognizer</
  <ms:resourceShortName>
    <ms:description xml:lang="en">Identify names of <em>persons</em>,
  </em>, <em>locations</em>, <em>organizations</em>, as
  well as <em>money amounts</em>, <em>time and date expressions</em>
  in English texts automatically. </ms:description>
    <ms:LRIIdentifier ms:LRIIdentifierScheme="http://w3id.org/meta-share/
  meta-share/elg">ELG id automatically assigned</ms:LRIIdentifier>
    <ms:version>v8.6</ms:version>
    <ms:additionalInfo>
      <ms:landingPage>https://cloud.gate.ac.uk/shopfront/
  displayItem/annie-named-entity-recognizer</ms:landingPage>
    </ms:additionalInfo>
    <ms:keyword xml:lang="en">Named Entity Recognition</ms:keyword>
    <ms:keyword xml:lang="en">English</ms:keyword>
    <ms:resourceProvider>
      <ms:Group>
        <ms:actorType>Group</ms:actorType>
        <ms:organizationName xml:lang="en">GATE Team,
  University of Sheffield</ms:organizationName>
        <ms:website>https://gate.ac.uk/</ms:website>
      </ms:Group>
    </ms:resourceProvider>
    <ms:publicationDate>2020-02-25</ms:publicationDate>
    <ms:resourceCreator>
      <ms:Person>
        <ms:actorType>Person</ms:actorType>
        <ms:surname xml:lang="en">Roberts</ms:surname>
        <ms:givenName xml:lang="en">Ian</ms:givenName>
        <ms:email>username3@somedomain.com</ms:email>
      </ms:Person>
    </ms:resourceCreator>
    <ms:intendedApplication>
      <ms:LTClassRecommended>http://w3id.org/meta-share/omtd-share/
  NamedEntityRecognition</ms:LTClassRecommended>
    </ms:intendedApplication>
    <ms:LRSubclass>
      <ms:ToolService>
        <ms:lrType>ToolService</ms:lrType>
        <ms:function>
```

(continues on next page)

(continued from previous page)

```

<ms:LTCClassRecommended>http://w3id.org/meta-
↪share/omtd-share/NamedEntityRecognition</ms:LTCClassRecommended>
</ms:function>
<ms:SoftwareDistribution>
  <ms:SoftwareDistributionForm>http://w3id.org/
↪meta-share/meta-share/dockerImage</ms:SoftwareDistributionForm>
  <ms:executionLocation>http://localhost:8080/
↪process</ms:executionLocation>
  <ms:dockerDownloadLocation>registry.gitlab.
↪com/european-language-grid/usfd/gate-ie-tools/annie:8.6-0.0.3</
↪ms:dockerDownloadLocation>
  <ms:licenceTerms>
    <ms:licenceTermsName xml:lang="en">
↪GNU Lesser General Public License v3.0 only</ms:licenceTermsName>
    <ms:licenceTermsURL>https://spdx.org/
↪licenses/LGPL-3.0-only.html</ms:licenceTermsURL>
    <ms:conditionOfUse>http://w3id.org/
↪meta-share/meta-share/unspecified</ms:conditionOfUse>
    <ms:LicenceIdentifier_
↪ms:LicenceIdentifierScheme="http://w3id.org/meta-share/meta-share/SPDX">LGPL-3.0-
↪only</ms:LicenceIdentifier>
    <ms:LicenceIdentifier_
↪ms:LicenceIdentifierScheme="http://w3id.org/meta-share/meta-share/elg">ELG-ENT-LIC-
↪270220-00000199</ms:LicenceIdentifier>
  </ms:licenceTerms>
</ms:SoftwareDistribution>
<ms:languageDependent>true</ms:languageDependent>
<ms:inputContentResource>
  <ms:processingResourceType>http://w3id.org/
↪meta-share/meta-share/file1</ms:processingResourceType>
  <ms:language>
    <ms:languageTag>en</ms:languageTag>
↪<ms:languageId>en</ms:languageId>
  </ms:language>
  <ms:mediaType>http://w3id.org/meta-share/
↪meta-share/text</ms:mediaType>
  <ms:dataFormat>http://w3id.org/meta-share/
↪omtd-share/Json</ms:dataFormat>
  <ms:characterEncoding>http://w3id.org/meta-
↪share/meta-share/UTF-8</ms:characterEncoding>
  </ms:inputContentResource>
  <ms:outputResource>
    <ms:processingResourceType>http://w3id.org/
↪meta-share/meta-share/file1</ms:processingResourceType>
    <ms:language>
      <ms:languageTag>en</ms:languageTag>
↪<ms:languageId>en</ms:languageId>
    </ms:language>
    <ms:mediaType>http://w3id.org/meta-share/
↪meta-share/text</ms:mediaType>
    <ms:dataFormat>http://w3id.org/meta-share/
↪omtd-share/Json</ms:dataFormat>
    <ms:characterEncoding>http://w3id.org/meta-
↪share/meta-share/UTF-8</ms:characterEncoding>
    <!-- annotations: :Address, :Date, :Location,
↪:Organization, :Person, :Money, :Percent, :Token, :SpaceToken, :Sentence -->
    <ms:annotationType>http://w3id.org/meta-
↪share/omtd-share/Person</ms:annotationType>

```

(continues on next page)

(continued from previous page)

```

<ms:annotationType>http://w3id.org/meta-
share/omtd-share/Location</ms:annotationType>
<ms:annotationType>http://w3id.org/meta-
share/omtd-share/Organization</ms:annotationType>
<ms:annotationType>http://w3id.org/meta-
share/omtd-share/Date</ms:annotationType>
</ms:outputResource>
<ms:trl>http://w3id.org/meta-share/meta-share/trl4</
ms:trl>
<ms:evaluated>>false</ms:evaluated>
</ms:ToolService>
</ms:LRSubclass>
</ms:LanguageResource>
</ms:DescribedEntity>
</ms:MetadataRecord>

```

Example 2: Machine Translation service

Edinburgh's German to English MT engine published at: <https://live.european-language-grid.eu/catalogue/#/resource/service/tool/623>

The Docker image for this service is stored at DockerHub.

```

<?xml version="1.0" encoding="UTF-8"?>
<ms:MetadataRecord xmlns:ms="http://w3id.org/meta-share/meta-share/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://w3id.org/meta-share/
meta-share/ ../Schema/ELG-SHARE.xsd">
  <ms:DescribedEntity>
    <ms:LanguageResource>
      <ms:entityType>LanguageResource</ms:entityType>
      <ms:resourceName xml:lang="en">UEDIN Machine Translation
Service for German to English</ms:resourceName>
      <ms:resourceShortName xml:lang="en">UEDIN-MT-DeEn</
ms:resourceShortName>
      <ms:description xml:lang="en">A machine translation (MT)
service for German-to-English translation based on the Marian machine translation
framework. The translation model is a basic transformer model trained on ca 13.3M
sentence pairs using Marian NMT</ms:description>
      <ms:LRIIdentifier ms:LRIIdentifierScheme="http://w3id.org/meta-
share/meta-share/elg">ELG id automatically assigned</ms:LRIIdentifier>
      <ms:version>1.0.0</ms:version>
      <ms:additionalInfo>
        <ms:email>user@somedomain.uk</ms:email>
      </ms:additionalInfo>
      <ms:keyword xml:lang="en">Machine Translation</ms:keyword>
      <ms:keyword xml:lang="en">German</ms:keyword>
      <ms:keyword xml:lang="en">English</ms:keyword>
      <ms:keyword xml:lang="en">Neural machine translation</
ms:keyword>
      <ms:keyword xml:lang="en">Marian framework</ms:keyword>
      <ms:resourceProvider>
        <ms:Organization>
          <ms:actorType>Organization</ms:actorType>
          <ms:organizationName xml:lang="en">UEDIN</
ms:organizationName>

```

(continues on next page)

(continued from previous page)

```

<ms:OrganizationIdentifier_
↪ms:OrganizationIdentifierScheme="http://w3id.org/meta-share/meta-share/elg">ELG-ENT-
↪ORG-280220-00000397</ms:OrganizationIdentifier>
<ms:website>https://www.ed.ac.uk/informatics/
↪</ms:website>
</ms:Organization>
</ms:resourceProvider>
<ms:publicationDate>2020-02-28</ms:publicationDate>
<ms:resourceCreator>
  <ms:Person>
    <ms:actorType>Person</ms:actorType>
    <ms:surname xml:lang="en">Germann</ms:surname>
    <ms:givenName xml:lang="en">Ulrich</
↪ms:givenName>
    <ms:PersonalIdentifier_
↪ms:PersonalIdentifierScheme="http://w3id.org/meta-share/meta-share/elg">ELG-ENT-PER-
↪050320-00000787</ms:PersonalIdentifier>
    <ms:email>user@somedomain.uk</ms:email>
  </ms:Person>
</ms:resourceCreator>
<ms:intendedApplication>
  <ms:LTCClassRecommended>http://w3id.org/meta-share/
↪omtd-share/MachineTranslation</ms:LTCClassRecommended>
</ms:intendedApplication>
<ms:LRSubclass>
  <ms:ToolService>
    <ms:lrType>ToolService</ms:lrType>
    <ms:function>
      <ms:LTCClassRecommended>http://w3id.
↪org/meta-share/omtd-share/MachineTranslation</ms:LTCClassRecommended>
    </ms:function>
    <ms:SoftwareDistribution>
      <ms:SoftwareDistributionForm>http://
↪w3id.org/meta-share/meta-share/dockerImage</ms:SoftwareDistributionForm>
      <ms:executionLocation>http://
↪localhost:18080/api/elg/v1</ms:executionLocation>
      <ms:dockerDownloadLocation>mt4elg-de-
↪en</ms:dockerDownloadLocation>
      <ms:additionalHWRequirements>limits_
↪memory: 2048Mi limits_cpu: 1.5</ms:additionalHWRequirements>
      <ms:licenceTerms>
        <ms:licenceTermsName xml:lang=
↪"en">CC BY-SA 4.0</ms:licenceTermsName>
        <ms:licenceTermsURL>https://
↪creativecommons.org/licenses/by-sa/4.0/</ms:licenceTermsURL>
        <ms:conditionOfUse>http://
↪w3id.org/meta-share/meta-share/attribution</ms:conditionOfUse>
        <ms:conditionOfUse>http://
↪w3id.org/meta-share/meta-share/shareAlike</ms:conditionOfUse>
      </ms:licenceTerms>
    </ms:SoftwareDistribution>
    <ms:languageDependent>true</
↪ms:languageDependent>
    <ms:inputContentResource>
      <ms:processingResourceType>http://
↪w3id.org/meta-share/meta-share/file1</ms:processingResourceType>
      <ms:language>

```

(continues on next page)

(continued from previous page)

```

↪ms:languageTag>                                <ms:languageTag>de</
↪ms:languageId>                                <ms:languageId>de</
                                                </ms:language>
↪share/meta-share/text</ms:mediaType>          <ms:mediaType>http://w3id.org/meta-
                                                <ms:dataFormat>http://w3id.org/meta-
↪share/omtd-share/Json</ms:dataFormat>          <ms:characterEncoding>http://w3id.org/
↪meta-share/meta-share/UTF-8</ms:characterEncoding>
                                                </ms:inputContentResource>
                                                <ms:outputResource>
↪w3id.org/meta-share/meta-share/file1</ms:processingResourceType>
                                                <ms:processingResourceType>http://
                                                <ms:language>
↪ms:languageTag>                                <ms:languageTag>en</
↪ms:languageId>                                <ms:languageId>en</
                                                </ms:language>
↪share/meta-share/text</ms:mediaType>          <ms:mediaType>http://w3id.org/meta-
                                                <ms:dataFormat>http://w3id.org/meta-
↪share/omtd-share/Json</ms:dataFormat>          <ms:characterEncoding>http://w3id.org/
↪meta-share/meta-share/UTF-8</ms:characterEncoding>
                                                </ms:outputResource>
↪trl4</ms:trl>                                <ms:trl>http://w3id.org/meta-share/meta-share/
                                                <ms:evaluated>false</ms:evaluated>
                                                </ms:ToolService>
                                                </ms:LRSubclass>
                                                </ms:LanguageResource>
                                                </ms:DescribedEntity>
</ms:MetadataRecord>

```

1.12.5 3. Register the service at ELG


The current release of ELG offers two options for registering a resource:

A. the ELG interactive editor (see *Use the interactive editor*)



For services integrated at ELG, you **MUST** select the “Service or Tool” form and register it as an “ELG-compatible service”. At the start pages of the interactive editor, when prompted, select “Yes”.

B. the upload of a metadata file that conforms to the *ELG schema* (see *Create and upload metadata files*).

For ELG-compatible services, you **MUST** check the box next to “ELG-compatible” at the upload page.



RELEASE 2

My grid  Annika Grützner-Zahn 

Technologies Resources Community Events Documentation About ELG


Do you want to contribute a tool that will be integrated in ELG as an ELG-compatible service (i.e., available through the ELG APIs)?

☐ Yes


☐ No

You will find more information at: https://european-language-grid.readthedocs.io/en/latest/all/3_Contributing/Service.html



Home Technologies Resources Events Documentation About ELG Contact us

 The European Language Grid has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825627 (ELG)






© 2021 ELG Consortium [Terms of Use](#)



RELEASE 2

My grid  Annika Grützner-Zahn 

Technologies Resources Community Events Documentation About ELG

 Add items  My items  Feedback  Administration  Go to catalogue

VALIDATE XML FILES **UPLOAD SINGLE ITEM** UPLOAD MULTIPLE ITEMS

You can upload one **xml** file each time.

It is highly recommended that you **validate** your XML file against the ELG schema before you proceed.

☐ Work in progress

If the metadata record is for a resource that you plan to deliver later, please check the "work in progress" box.

☐ ELG-compatible service

If the metadata record is for a service to be integrated in ELG (https://european-language-grid.readthedocs.io/en/release1.1.1/all/3_Contributing/Service.html), please check the box "ELG-compatible service".

Drag & Drop your file or [Browse](#)

1.12.6 4. Manage and submit the service for publication

Through the “My items” page you can access your metadata record (see *Manage your items*) and edit it until you are satisfied. You can then *submit it for publication*, in line with the *publication lifecycle* defined for ELG metadata records.

At this stage, the metadata record can no longer be edited and is only visible to you and to us, the ELG platform administrators.

Before it is published, the service undergoes a validation process, which is described in detail at *CHAPTER 4: VALIDATING ITEMS*.

During this process, the service is deployed to ELG, configured and tested to ensure it conforms to the ELG technical specifications. We describe here the main steps in this process:

- **LT Service is deployed to ELG and configured:** The LT service is deployed (by the validator) to the k8s cluster by creating the appropriate configuration *YAML* file and uploading to the respective GitLab repository. The CI/CD pipeline that is responsible for deployments will automatically install the new service at the k8s cluster. If you request it, a separate dedicated k8s namespace can be created for the LT service before creating the *YAML* file. The validator of the service assigns to it:
 - the k8s REST endpoint that will be used for invoking it, according to the following template: `http://{{k8s service name for the registered LT tool}}.{{k8s namespace for the registered LT tool}}.svc.cluster.local{{the path where the REST service is running at}}`. The `{{the path where the REST service is running at}}` part can be found in the `executionLocation` field in the metadata. For instance, for the Edinburgh’s MT tool above it is `‘/api/elg/v1’`.
 - An ID that will be used to call it.
 - Which “try out” UI will be used for testing it and visualizing the returned results.
- **LT Service is tested:** On the LT landing page, there is a “Try out” tab and a “Code samples” tab, which can both be used to test the service with some input; see *Use an LT service* section. The validator can help you identify integration issues and resolve them. This process is continued until the LT service is correctly integrated to the platform. The procedure may require access to the k8s cluster for the validator (e.g., to check containers start-up/failures, logs, etc.).
- **LT Service is published:** When the LT service works as expected, the validator will approve it; the metadata record is then published and visible to all ELG users through the catalogue.

1.12.7 Frequently asked questions

Question: What is a k8s namespace and when should an LT Provider ask for one?

Answer: A k8s namespace is a virtual sub-cluster, which can be used to restrict access to the respective containers that run within it. You should ask for a dedicated namespace (in ELG k8s cluster) when you need to ensure isolation and security; i.e, limit access to your container, logs etc.

Question: The image that I have created is not publicly available. Is it possible to register it to the ELG platform?

Answer: Yes, it can be registered. A k8s secret containing the required credentials will be created for the namespace in which your image is going to be deployed. k8s will then be able to pull the image and deploy it.

Question: Are there any requirements for `executionLocation`? For example, an IE tool has to expose a specific path or use a specific port?

Answer: No, you can use any valid port or path. This holds for any kind of LT tool (IE, MT, ASR, etc.). The internal container port will be mapped (via port mapping) to port 80. Remember that the endpoint of the LT service follows this pattern: `http://{k8s service name for the registered LT tool}.{k8s namespace for the registered LT tool}.svc.cluster.local{the path where the REST service is running at}`, which assumes that the service is exposed to port 80.

Question: I have *n* different versions of the same IE LT tool; e.g., one version per language. How should I register them to the platform? I have to create one Docker image with all the different versions or one image per version?

Answer: Both are possible. In both cases you will have to provide a separate metadata record for each LT tool. However, in the case where the tools are packaged together, **all** metadata records must point to the same image location (`dockerDownloadLocation`) and each of them has to listen in a different HTTP endpoint (`executionLocation`) but on the same port (for simplicity). E.g., `"http://localhost:8080/NamedEntityRecognitionEN"`, `"http://localhost:8080/NamedEntityRecognitionDE"`.

Question: Should the Docker image that I will provide have a specific tag?

Answer: The images that are stored in GitLab or DockerHub are not immutable, even when they have been assigned a specific/custom tag; thus, it is possible that they are overwritten (by their creators). ELG (currently) does not have a private Docker registry that caches images. Therefore, when ELG will try (at some point) to spawn a new instance of an LT service, it might download (pull) and use an image that is not (any more) ELG compatible, because it has been overwritten (e.g. by accident). So, yes, it is recommended (but not enforced) to put a custom tag (dedicated for ELG) to the image that you will register, since it is usually more common to override the `:latest` one.

Question: How many resources will be allocated for my LT container in the k8s cluster?

Answer: By default, 512MB of RAM and half a CPU core. If your LT service requires more resources you have to specify it by using the `additionalHWRequirements` metadata element (see the MT example above) or by communicating with the ELG administrators.

Question: What is a YAML file and what does it contain?

Answer: Each service has a **YAML** file which contains information about the allocated resources in the k8s cluster (see question above) and the scaling parameters (whether it is readily available at all times or started on demand).

1.13 Contribute an non-ELG compatible tool or service

This page describes how to contribute *tools or services* that do not follow the ELG specifications to the European Language Grid. These include downloadable tools that run locally, web services running outside ELG, etc.

You can describe a tool or service and upload its contents at ELG or include in its description a link to the location it can be accessed from.

1.13.1 0. Before you start

- Please make sure that the software you want to contribute complies with our *terms of use*.
- Please make sure you have *registered* and been assigned the *provider role*.

1.13.2 1. Prepare the content files (for ELG hosted resources)

If you wish to upload the software (e.g. software code, downloadable executable files that run locally, docker images) at ELG, you must package it in a compressed format (currently as a .zip file).

1.13.3 2. Describe the software

Metadata overview

The software must be described according to the *ELG schema* and comply at least with the minimal version. The metadata elements that you need to provide for the software comprise a set of elements organized (for presentation purposes) into the following groups:

- *Minimal elements for all entities*
- *Minimal elements for all language resources and technologies*
- *Minimal elements for tools/services.*

1.13.4 3. Register the software at ELG

The current release of ELG offers two options for registering a catalogue item:

- the **ELG interactive editor** (see *Use the interactive editor*)
- the **upload of a metadata file** that conforms to the *ELG schema* in XML format (see *Create and upload metadata files*).

If you wish to upload the software, follow the instructions described here.

1.13.5 4. Manage and submit for publication

Through the “My items” page you can access your metadata record (see *Manage your items*) and edit it until you are satisfied. You can then *submit it for publication*, in line with the *publication lifecycle* defined for ELG metadata records.

At this stage, the metadata record can no longer be edited and is only visible to you and to us, the ELG platform administrators.

Before it is published, your submission undergoes a validation process, which is described in detail at *CHAPTER 4: VALIDATING ITEMS*.

Once approved, it will appear on the ELG catalogue and you will receive a notification email.

1.14 Contribute a corpus/dataset

This page describes how to contribute a *corpus* to the European Language Grid. You can describe a corpus and upload its contents at ELG or include in its description a link to the location it can be accessed from.

1.14.1 0. Before you start

- Please make sure that the corpus you want to contribute complies with our *terms of use*.
- Please make sure you have *registered* and been assigned the *provider role*.

1.14.2 1. Prepare the content files (for ELG hosted resources)

If you wish to upload the corpus at ELG, you must package it in a compressed format (currently as a .zip file).

If the files are available in multiple formats, (e.g. in XML, TXT and PDF formats), you are advised to package them in different zip files by data format.

For recommendations on the criteria for organizing files into corpora, you can also see *below*.

1.14.3 2. Describe the corpus

Metadata overview

The corpus must be described according to the *ELG schema* and comply at least with the minimal version. The metadata elements that you need to provide for the corpus comprise a set of elements organized (for presentation purposes) into the following groups:

- *Minimal elements for all entities*
- *Minimal elements for all language resources and technologies*
- *Minimal elements for corpora.*

Examples

Example 1: Bilingual raw corpus

Bilingual Bulgarian-English corpus from the National Revenue Agency (BG) (Processed)

Published at: <https://live.european-language-grid.eu/catalogue/#/resource/service/corpus/734>

```
<?xml version="1.0" encoding="UTF-8"?>
<ms:MetadataRecord xmlns:ms="http://w3id.org/meta-share/meta-share/"      xmlns:xsi=
↪ "http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://w3id.org/
↪ meta-share/meta-share/ ../Schema/ELG-SHARE.xsd">
<ms:DescribedEntity>
  <ms:LanguageResource>
    <ms:entityType>LanguageResource</ms:entityType>
    <ms:resourceName xml:lang="en">Bilingual Bulgarian-English corpus_
↪ from the National Revenue Agency (BG) (Processed)</ms:resourceName>
    <ms:description xml:lang="en">Bilingual Bulgarian-English corpus of_
↪ administrative documents on the Refund of Value Added Tax from the Bulgarian_
↪ National Revenue Agency.
```

(continues on next page)

(continued from previous page)

Bilingual Bulgarian-English corpus of administrative documents on the Refund of Value Added Tax from the Bulgarian National Revenue Agency. It was offered as collection of documents by the Bulgarian National Revenue Agency. Modules of the ILSP Focused Crawler was used for the normalization, cleaning, (near) deduplication and identification of parallel documents. The Maligna sentence aligner was used for extracting segment alignments from crawled parallel documents. As a post-processing step, alignments were merged into one TMX file. The following filters were applied: TMX files generated from document pairs which have been identified by non-aupidh methods were discarded ; TMX files with a zeroToOne_alignments/total_alignments ratio larger than 0.16, were discarded ; Alignments of non-[1:1] type(s) were discarded. ; Alignments with a TUV (after normalization) that has less than 1 tokens, were annotated ; Alignments with a 11/12 TUV length ratio smaller than 0.6 or larger than 1.6, were annotated ; Alignments in which different digits appear in each TUV were kept and annotated. ; Alignments with identical TUVs (after normalization) were annotated ; Alignments with only non-letters in at least one of their TUVs were annotated ; Duplicate alignments were kept and were annotated. The mean value of aligner's scores is 5.714609036504669, the std value is 1.8063256236105307. The mean value of length (in terms of characters) ratios is 1.0040012545201242 and the std value is 0.26545877788005745. There are 832 TUs with no annotation, containing 13336 words and 2604 lexical types in bul and 15010 words and 2031 lexical types in eng. The mean value of aligner's scores is 6.336834960545485, the std value is 1.53829791384023

```

<ms:LRIdentifier ms:LRIdentifierScheme="http://w3id.org/meta-share/
meta-share/other">ELRC_471</ms:LRIdentifier>
<ms:version>2.0</ms:version>
<ms:additionalInfo>
  <ms:landingPage>https://elrc-share.eu/repository/browse/
bilingual-bulgarian-english-corpus-from-the-national-revenue-agency-bg-processed/
4ed47824d04a11e7b7d400155d026706dbe4fc9f12424b5ba0a749fd6758072b/</ms:landingPage>
</ms:additionalInfo>
<ms:additionalInfo>
  <ms:email>contact@someDomain.com</ms:email>
</ms:additionalInfo>
<ms:contact>
  <ms:Person>
    <ms:actorType>Person</ms:actorType>
    <ms:surname xml:lang="en">Rusinova</ms:surname>
    <ms:givenName xml:lang="en">Annie</ms:givenName>
    <ms:email>contact@someDomain.com</ms:email>
  </ms:Person>
</ms:contact>
<ms:iprHolder>
  <ms:Organization>
    <ms:actorType>Organization</ms:actorType>
    <ms:organizationName xml:lang="en">National Revenue
Agency (BG)</ms:organizationName>
    <ms:website>http://www.nap.bg/en/</ms:website>
  </ms:Organization>
</ms:iprHolder>
<ms:keyword xml:lang="en">corpus</ms:keyword>
<ms:domain>
  <ms:categoryLabel xml:lang="en">FINANCE</ms:categoryLabel>
  <ms:DomainIdentifier ms:DomainClassificationScheme="http://
w3id.org/meta-share/meta-share/EUROVOC">24</ms:DomainIdentifier>
</ms:domain>
<ms:fundingProject>
  <ms:projectName xml:lang="en">European Language Resource
Coordination LOT3</ms:projectName>

```

(continues on next page)

(continued from previous page)

```

        <ms:ProjectIdentifier ms:ProjectIdentifierScheme="http://w3id.
↪org/meta-share/meta-share/other">Tools and Resources for CEF Automated Translation -
↪ LOT3 (SMART 2015/1091 - 30-CE-0816766/00-92) </ms:ProjectIdentifier>
        <ms:website>http://www.lr-coordination.eu</ms:website>
    </ms:fundingProject>
    <ms:validated>true</ms:validated>
    <ms:validation>
        <ms:validationDetails xml:lang="en">validated</
↪ms:validationDetails>
    </ms:validation>
    <ms:relation>
        <ms:relationType xml:lang="en">isAlignedVersionOf</
↪ms:relationType>
        <ms:relatedLR>
            <ms:resourceName xml:lang="en">Bilingual Bulgarian-
↪English corpus from the National Revenue Agency (BG) </ms:resourceName>
            <ms:LRIdentifier ms:LRIdentifierScheme="http://w3id.
↪org/meta-share/meta-share/other">ELRC_447</ms:LRIdentifier>
        </ms:relatedLR>
    </ms:relation>
    <ms:LRSubclass>
        <ms:Corpus>
            <ms:lrType>Corpus</ms:lrType>
            <ms:corpusSubclass>http://w3id.org/meta-share/meta-
↪share/rawCorpus</ms:corpusSubclass>
            <ms:CorpusMediaPart>
                <ms:CorpusTextPart>
                    <ms:corpusMediaType>CorpusTextPart</
↪ms:corpusMediaType>
                    <ms:mediaType>http://w3id.org/meta-
↪share/meta-share/text</ms:mediaType>
                    <ms:lingualityType>http://w3id.org/
↪meta-share/meta-share/bilingual</ms:lingualityType>
                    <ms:multilingualityType>http://w3id.
↪org/meta-share/meta-share/parallel</ms:multilingualityType>
                    <ms:language>
                        <ms:languageTag>bg</
↪ms:languageTag>
                        <ms:languageId>bg</
↪ms:languageId>
                        <ms:scriptId>Cyril</
↪ms:scriptId>
                    </ms:language>
                    <ms:language>
                        <ms:languageTag>en</
↪ms:languageTag>
                        <ms:languageId>en</
↪ms:languageId>
                    </ms:language>
                    <ms:textType>
                        <ms:categoryLabel xml:lang="en
↪">administrativeTexts</ms:categoryLabel>
                    </ms:textType>
                    <ms:TextGenre>
                        <ms:categoryLabel xml:lang="en
↪">official</ms:categoryLabel>
                    </ms:TextGenre>

```

(continues on next page)

(continued from previous page)

```

    <ms:creationMode>http://w3id.org/meta-
    ↪share/meta-share/mixed</ms:creationMode>
    <ms:hasOriginalSource>
    ↪    <ms:resourceName xml:lang="en"
    ↪    <ms:LRIdentifier_
    ↪ms:LRIdentifierScheme="http://w3id.org/meta-share/meta-share/other">ELRC_447</
    ↪ms:LRIdentifier>
    ↪    </ms:hasOriginalSource>
    ↪    <ms:creationDetails xml:lang="en">See_
    ↪description for creation details</ms:creationDetails>
    ↪    </ms:CorpusTextPart>
    ↪    </ms:CorpusMediaPart>
    ↪    <ms:DatasetDistribution>
    ↪    <ms:DatasetDistributionForm>http://w3id.org/
    ↪meta-share/meta-share/downloadable</ms:DatasetDistributionForm>
    ↪    <ms:accessLocation>https://elrc-share.eu/
    ↪repository/download/
    ↪4ed47824d04a11e7b7d400155d026706dbe4fc9f12424b5ba0a749fd6758072b/</
    ↪ms:accessLocation>
    ↪    <ms:distributionTextFeature>
    ↪    <ms:size>
    ↪    <ms:amount>1292</ms:amount>
    ↪    <ms:sizeUnit>http://w3id.org/
    ↪meta-share/meta-share/unit</ms:sizeUnit>
    ↪    </ms:size>
    ↪    <ms:dataFormat>http://w3id.org/meta-
    ↪share/omtd-share/Xml</ms:dataFormat>
    ↪    <ms:characterEncoding>http://w3id.org/
    ↪meta-share/meta-share/UTF-8</ms:characterEncoding>
    ↪    </ms:distributionTextFeature>
    ↪    <ms:licenceTerms>
    ↪    <ms:licenceTermsName xml:lang="en">
    ↪publicDomain</ms:licenceTermsName>
    ↪    <ms:licenceTermsURL>https://elrc-
    ↪share.eu/terms/publicDomain.html</ms:licenceTermsURL>
    ↪    <ms:conditionOfUse>http://w3id.org/
    ↪meta-share/meta-share/noConditions</ms:conditionOfUse>
    ↪    <ms:LicenceIdentifier_
    ↪ms:LicenceIdentifierScheme="http://w3id.org/meta-share/meta-share/elg">publicDomain
    ↪</ms:LicenceIdentifier>
    ↪    </ms:licenceTerms>
    ↪    <ms:cost>
    ↪    <ms:amount>0</ms:amount>
    ↪    <ms:currency>http://w3id.org/meta-
    ↪share/meta-share/euro</ms:currency>
    ↪    </ms:cost>
    ↪    </ms:DatasetDistribution>
    ↪    <ms:personalDataIncluded>>false</
    ↪ms:personalDataIncluded>
    ↪    <ms:sensitiveDataIncluded>>false</
    ↪ms:sensitiveDataIncluded>
    ↪    </ms:Corpus>
    ↪    </ms:LRSubclass>
    ↪    </ms:LanguageResource>
</ms:DescribedEntity>
</ms:MetadataRecord>

```


Example 2: Annotated corpus

Greek Textual Entailment corpus

Published at: <https://live.european-language-grid.eu/catalogue/#/resource/service/corpus/649>

```
<?xml version="1.0" encoding="UTF-8"?>
<ms:MetadataRecord xmlns:ms="http://w3id.org/meta-share/meta-share/" xmlns:xsi="http://
  ↪www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://w3id.org/meta-share/
  ↪meta-share/ ../Schema/ELG-SHARE.xsd">
  <ms:DescribedEntity>
    <ms:LanguageResource>
      <ms:entityType>LanguageResource</ms:entityType>
      <ms:resourceName xml:lang="en">Greek Textual Entailment Corpus</
  ↪ms:resourceName>
      <ms:resourceShortName xml:lang="en">GTEC</ms:resourceShortName>
      <ms:description xml:lang="en">GTEC consists of 600 T-H pairs manually
  ↪annotated for entailment (i.e. whether T entails H or not) by human annotators. The
  ↪dataset which is tailored to guide training and evaluation of prospect RTE systems,
  ↪is equally divided in three subsets each one representing the output of a specific
  ↪HLT application: Question Answering (QA), Comparable Documents (CD) and Machine
  ↪Translation (MT), and pertaining to specific subject fields (e.g. law, politics,
  ↪travel). T-H examples that correspond to success and failure cases of the afore-
  ↪mentioned applications have been included in the corpus. The annotations provided
  ↪are conformant to the RTE1 and RTE2 challenges.</ms:description>
      <ms:version>v1.0.0 (automatically assigned)</ms:version>
      <ms:additionalInfo>
        <ms:email>username@someDomain.com</ms:email>
      </ms:additionalInfo>
      <ms:additionalInfo>
        <ms:email>username3@someDomain.com</ms:email>
      </ms:additionalInfo>
      <ms:contact>
        <ms:Person>
          <ms:actorType>Person</ms:actorType>
          <ms:surname xml:lang="en">Giouli</ms:surname>
          <ms:givenName xml:lang="en">Voula</ms:givenName>
          <ms:email>username@someDomain.com</ms:email>
        </ms:Person>
      </ms:contact>
      <ms:contact>
        <ms:Person>
          <ms:actorType>Person</ms:actorType>
          <ms:surname xml:lang="en">Piperidis</ms:surname>
          <ms:givenName xml:lang="en">Stelios</ms:givenName>
          <ms:email>username3@someDomain.com</ms:email>
        </ms:Person>
      </ms:contact>
      <ms:keyword xml:lang="en">corpus</ms:keyword>
      <ms:domain>
        <ms:categoryLabel xml:lang="en">law</ms:categoryLabel>
      </ms:domain>
      <ms:domain>
        <ms:categoryLabel xml:lang="en">politics</ms:categoryLabel>
      </ms:domain>
      <ms:domain>
        <ms:categoryLabel xml:lang="en">travel</ms:categoryLabel>
      </ms:domain>
    </ms:DescribedEntity>
  </ms:LanguageResource>
</ms:MetadataRecord>
```

(continues on next page)

(continued from previous page)

```

    <ms:resourceCreator>
      <ms:Organization>
        <ms:actorType>Organization</ms:actorType>
        <ms:organizationName xml:lang="en">Institute for
↪Language and Speech Processing</ms:organizationName>
        <ms:website>http://www.ilsp.gr</ms:website>
      </ms:Organization>
    </ms:resourceCreator>
    <ms:intendedApplication>
      <ms:LTCClassRecommended>http://w3id.org/meta-share/omtd-share/
↪AnnotationOfTextualEntailment</ms:LTCClassRecommended>
    </ms:intendedApplication>
    <ms:actualUse>
      <ms:usedInApplication>
        <ms:LTCClassRecommended>http://w3id.org/meta-share/
↪omtd-share/AnnotationOfTextualEntailment</ms:LTCClassRecommended>
      </ms:usedInApplication>
      <ms:actualUseDetails xml:lang="en">nlpApplications</
↪ms:actualUseDetails>
    </ms:actualUse>
    <ms:isDocumentedBy>
      <ms:title xml:lang="en">Building a Greek corpus of Textual
↪Entailment</ms:title>
      <ms:DocumentIdentifier ms:DocumentIdentifierScheme="http://
↪purl.org/spar/datacite/url">http://www.lrec-conf.org/proceedings/lrec2008/pdf/427_
↪paper.pdf</ms:DocumentIdentifier>
    </ms:isDocumentedBy>
    <ms:LRSubclass>
      <ms:Corpus>
        <ms:lrType>Corpus</ms:lrType>
        <ms:corpusSubclass>http://w3id.org/meta-share/meta-
↪share/annotatedCorpus</ms:corpusSubclass>
        <ms:CorpusMediaPart>
          <ms:CorpusTextPart>
            <ms:corpusMediaType>CorpusTextPart</
↪ms:corpusMediaType>
          <ms:mediaType>http://w3id.org/meta-
↪share/meta-share/text</ms:mediaType>
          <ms:lingualityType>http://w3id.org/
↪meta-share/meta-share/monolingual</ms:lingualityType>
          <ms:language>
            <ms:languageTag>el</
↪ms:languageTag>
            <ms:languageId>el</
↪ms:languageId>
          </ms:language>
          <ms:annotation>
            <ms:annotationType>http://
↪w3id.org/meta-share/omtd-share/Lemma</ms:annotationType>
            <ms:annotationStandoff>false</
↪ms:annotationStandoff>
            <ms:annotationMode>http://
↪w3id.org/meta-share/meta-share/mixed</ms:annotationMode>
            <ms:annotationModeDetails
↪xml:lang="en">automatic annotation followed with manual disambiguation</
↪ms:annotationModeDetails>
            <ms:isAnnotatedBy>

```

(continues on next page)

(continued from previous page)

```

<ms:resourceName_
↪xml:lang="en">ILSP-Lemmatizer</ms:resourceName>
    </ms:isAnnotatedBy>
</ms:annotation>
<ms:annotation>
    <ms:annotationType>http://
↪w3id.org/meta-share/omtd-share/PartOfSpeech</ms:annotationType>
    <ms:annotationStandoff>>false</
↪ms:annotationStandoff>
    <ms:typesystem>
        <ms:resourceName_
↪xml:lang="en">ILSP/PAROLE tagset</ms:resourceName>
        </ms:typesystem>
        <ms:annotationMode>http://
↪w3id.org/meta-share/meta-share/mixed</ms:annotationMode>
        <ms:annotationModeDetails_
↪xml:lang="en">automatic annotation followed with manual disambiguation</
↪ms:annotationModeDetails>
        <ms:isAnnotatedBy>
            <ms:resourceName_
↪xml:lang="en">ILSP FBT POS tagger</ms:resourceName>
            </ms:isAnnotatedBy>
        </ms:annotation>
        <ms:annotation>
            <ms:annotationType>http://
↪w3id.org/meta-share/omtd-share/SyntacticAnnotationType</ms:annotationType>
            <ms:annotationStandoff>>false</
↪ms:annotationStandoff>
            <ms:annotationMode>http://
↪w3id.org/meta-share/meta-share/mixed</ms:annotationMode>
            </ms:annotation>
            <ms:annotation>
                <ms:annotationType>http://
↪w3id.org/meta-share/omtd-share/SyntacticAnnotationType</ms:annotationType>
                <ms:annotationStandoff>>false</
↪ms:annotationStandoff>
                <ms:annotationMode>http://
↪w3id.org/meta-share/meta-share/mixed</ms:annotationMode>
                <ms:annotationModeDetails_
↪xml:lang="en">Automatic annotation followed by manual correction</
↪ms:annotationModeDetails>
                </ms:annotation>
            <ms:annotation>
                <ms:annotationType>http://
↪w3id.org/meta-share/omtd-share/SemanticAnnotationType</ms:annotationType>
                <ms:annotationStandoff>>false</
↪ms:annotationStandoff>
                <ms:annotationMode>http://
↪w3id.org/meta-share/meta-share/manual</ms:annotationMode>
            </ms:annotation>
            <ms:creationMode>http://w3id.org/meta-
↪share/meta-share/mixed</ms:creationMode>
            <ms:originalSourceDescription_
↪xml:lang="en">web news</ms:originalSourceDescription>
            <ms:originalSourceDescription_
↪xml:lang="en">EU texts</ms:originalSourceDescription>
        </ms:CorpusTextPart>

```

(continues on next page)

(continued from previous page)

```

        </ms:CorpusMediaPart>
        <ms:DatasetDistribution>
            <ms:DatasetDistributionForm>http://w3id.org/
↪meta-share/meta-share/downloadable</ms:DatasetDistributionForm>
            <ms:accessLocation>http://metashare.ilsp.
↪gr:8080/repository/download/
↪26dca2fe63d211e29b2c842b2b6a04d7db87c85bfbe34326bb4c2e88b8c4da85</ms:accessLocation>
            <ms:distributionTextFeature>
                <ms:size>
                    <ms:amount>600</ms:amount>
                    <ms:sizeUnit>http://w3id.org/
↪meta-share/meta-share/T-HPair</ms:sizeUnit>
                </ms:size>
                <ms:dataFormat>http://w3id.org/meta-
↪share/omtd-share/Xml</ms:dataFormat>
            </ms:distributionTextFeature>
            <ms:licenceTerms>
                <ms:licenceTermsName xml:lang="en">CC-
↪BY-4.0</ms:licenceTermsName>
                <ms:licenceTermsURL>https://spdx.org/
↪licenses/CC-BY-4.0.html</ms:licenceTermsURL>
                <ms:conditionOfUse>http://w3id.org/
↪meta-share/meta-share/attribution</ms:conditionOfUse>
            </ms:licenceTerms>
            <ms:attributionText xml:lang="en">Greek_
↪Textual Entailment Corpus by Athena R.C./ILSP used under CC-BY licence</
↪ms:attributionText>
        </ms:DatasetDistribution>
        <ms:personalDataIncluded>>false</
↪ms:personalDataIncluded>
        <ms:sensitiveDataIncluded>>false</
↪ms:sensitiveDataIncluded>
        </ms:Corpus>
    </ms:LRSubclass>
</ms:LanguageResource>
</ms:DescribedEntity>
</ms:MetadataRecord>

```

Recommendations for the description and organization of corpora

Corpora are composed of files that can be organized according to different criteria. For instance, a multilingual corpus of texts from various domains can be described as a whole (one metadata record) or split into subsets (and corresponding metadata records) using the language and/or domain criteria.

In order to facilitate users, especially those accessing ELG through programmatic APIs, to automatically identify, download and use corpora as is, without having to download them and manually search among them the subsets that interest them, we include here some **recommendations**.

Providers that upload their corpora into ELG can use the following recommendations to appropriately package the files and register them as one or multiple metadata records.

Providers that grant access to corpora through hyperlinks can use as a criterion for the registration of one or multiple records the availability of the corpus through a direct link (`downloadLocation`).

The following cases are recommended:

- **multilingual corpora:** we recommend the split into bilingual pairs, so that users can easily find them and use them, for instance, to train bilingual models;
- **corpora of shared tasks:** these are usually already split into training, development, gold, and test corpus, with a direct link to each of these datasets; we suggest to use this as an established practice and register them as separate metadata records.

In all cases, providers can create a “parent” metadata record, to which the metadata records of the subsets can point, using the `isPartOf` relation.

On the other hand, the concept of `distribution` (see [ELG schema](#)) can be used to describe resources with the same metadata record in the following cases:

- **corpora available in multiple formats:** these can be described with the same metadata record, but different distributions
- corpora available with **different licensing terms:** if the same corpus is available with different licensing terms (e.g. for non-commercial use for free and for commercial use on a fee),
- corpora available **with multiple licences:** composite resources (e.g. a corpus available via an interface, a tool available with a model) may be licensed with multiple licences, one for the data and one for the tool. These can be described with the same metadata record and distribution where both licences are added.

1.14.4 3. Register the corpus at ELG

The current release of ELG offers two options for registering a catalogue item:

- the ELG **interactive editor** (see [Use the interactive editor](#))
- the **upload of a metadata file** that conforms to the [ELG schema](#) in XML format (see [Create and upload metadata files](#)).

To upload the content files for the corpus, you can follow the procedure described [here](#).

1.14.5 4. Manage and submit for publication

Through the “My items” page you can access your metadata record (see [Manage your items](#)) and edit it until you are satisfied. You can then [submit it for publication](#), in line with the [publication lifecycle](#) defined for ELG metadata records.

At this stage, the metadata record can no longer be edited and is only visible to you and to us, the ELG platform administrators.

Before it is published, your submission undergoes a validation process, which is described in detail at [CHAPTER 4: VALIDATING ITEMS](#).

Once approved, it will appear on the ELG catalogue and you will receive a notification email.

1.15 Contribute a model

This page describes how to contribute a *model* to the European Language Grid. You can describe a model and upload its contents at ELG or include in its description a link to the location it can be accessed from.

1.15.1 0. Before you start

- Please make sure that the model you want to contribute complies with our *terms of use*.
- Please make sure you have *registered* and been assigned the *provider role*.

1.15.2 1. Prepare the content files (for ELG hosted resources)

If you wish to upload the model at ELG, you must package it in a compressed format (currently as a .zip file).

If the files are available in multiple formats, (e.g. in XML, TXT and PDF formats), you are advised to package them in different zip files by data format.

1.15.3 2. Describe the model

Metadata overview

The model must be described according to the *ELG schema* and comply at least with the minimal version. The metadata elements that you need to provide for the model comprise a set of elements organized (for presentation purposes) into the following groups:

- *Minimal elements for all entities*
- *Minimal elements for all language resources and technologies*
- *Minimal elements for language descriptions*

Examples

Example 1: Machine Learning (ML) model

English Model (CoNLL-2003) for NameTag

```
<?xml version="1.0" encoding="UTF-8"?>
<ms:MetadataRecord xmlns:ms="http://w3id.org/meta-share/meta-share/" xmlns:xsi="http://
↪www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://w3id.org/meta-share/
↪meta-share/ ../Schema/ELG-SHARE.xsd">
  <ms:DescribedEntity>
    <ms:LanguageResource>
      <ms:entityType>LanguageResource</ms:entityType>
      <ms:resourceName xml:lang="en">English Model (CoNLL-2003) for_
↪NameTag</ms:resourceName>
      <ms:description xml:lang="en">English model for NameTag, a_
↪named entity recognition tool. The model is trained on CoNLL-2003 training data._
↪Recognizes PER, ORG, LOC and MISC named entities. Achieves F-measure 84.73 on CoNLL-
↪2003 test data.</ms:description>
      <ms:version>undefined</ms:version>
      <ms:additionalInfo>
```

(continues on next page)

(continued from previous page)

```

<ms:landingPage>http://hdl.handle.net/11234/1-3118</
↪ms:landingPage>
    </ms:additionalInfo>
    <ms:keyword xml:lang="en">NameTag</ms:keyword>
    <ms:keyword xml:lang="en">English</ms:keyword>
    <ms:keyword xml:lang="en">named entity recognition</
↪ms:keyword>
    <ms:resourceProvider>
        <ms:Organization>
            <ms:actorType>Organization</ms:actorType>
            <ms:organizationName xml:lang="en">Charles_
↪University, Faculty of Mathematics and Physics, Institute of Formal and Applied_
↪Linguistics (UFAL)</ms:organizationName>
        </ms:Organization>
    </ms:resourceProvider>
    <ms:publicationDate>2014-04-08</ms:publicationDate>
    <ms:resourceCreator>
        <ms:Person>
            <ms:actorType>Person</ms:actorType>
            <ms:surname xml:lang="en">Straka</ms:surname>
            <ms:givenName xml:lang="en">Milan</
↪ms:givenName>
        </ms:Person>
    </ms:resourceCreator>
    <ms:resourceCreator>
        <ms:Person>
            <ms:actorType>Person</ms:actorType>
            <ms:surname xml:lang="en">Straková</
↪ms:surname>
            <ms:givenName xml:lang="en">Jana</
↪ms:givenName>
        </ms:Person>
    </ms:resourceCreator>
    <ms:fundingProject>
        <ms:projectName xml:lang="en">LINDAT/CLARIN: Institut_
↪pro analýzu, zpracování a distribuci lingvistických dat</ms:projectName>
        <ms:grantNumber>LM2010013</ms:grantNumber>
        <ms:fundingType>http://w3id.org/meta-share/meta-share/
↪nationalFunds</ms:fundingType>
        <ms:funder>
            <ms:Organization>
                <ms:actorType>Organization</
↪ms:actorType>
                <ms:organizationName xml:lang="en">
↪Ministerstvo školství, mládeže a tělovýchovy České republiky</ms:organizationName>
            </ms:Organization>
        </ms:funder>
    </ms:fundingProject>
    <ms:fundingProject>
        <ms:projectName xml:lang="en">Integrace jazykových_
↪zdrojů za účelem extrakce informací z přirozených textů</ms:projectName>
        <ms:grantNumber>1ET101120503</ms:grantNumber>
        <ms:fundingType>http://w3id.org/meta-share/meta-share/
↪nationalFunds</ms:fundingType>
        <ms:funder>
            <ms:Organization>
                <ms:actorType>Organization</
↪ms:actorType>

```

(continues on next page)

(continued from previous page)

```

<ms:organizationName xml:lang="en">
↳Grantová agentura Akademie věd České republiky</ms:organizationName>
    </ms:Organization>
  </ms:funder>
</ms:fundingProject>
<ms:fundingProject>
  <ms:projectName xml:lang="en">Teoretické základy
↳informatiky a výpočetní lingvistiky</ms:projectName>
  <ms:grantNumber>SVV 267 314</ms:grantNumber>
  <ms:fundingType>http://w3id.org/meta-share/meta-share/
↳nationalFunds</ms:fundingType>
  <ms:funder>
    <ms:Organization>
      <ms:actorType>Organization</
↳ms:actorType>
    <ms:organizationName xml:lang="en">
↳Univerzita Karlova v Praze (mimo GAUK)</ms:organizationName>
      </ms:Organization>
    </ms:funder>
  </ms:fundingProject>
  <ms:fundingProject>
    <ms:projectName xml:lang="en">LINDAT/CLARIN: Institut
↳pro analýzu, zpracování a distribuci lingvistických dat</ms:projectName>
    <ms:grantNumber>LM2015071</ms:grantNumber>
    <ms:fundingType>http://w3id.org/meta-share/meta-share/
↳nationalFunds</ms:fundingType>
    <ms:funder>
      <ms:Organization>
        <ms:actorType>Organization</
↳ms:actorType>
      <ms:organizationName xml:lang="en">
↳Ministerstvo školství, mládeže a tělovýchovy České republiky</ms:organizationName>
        </ms:Organization>
      </ms:funder>
    </ms:fundingProject>
    <ms:fundingProject>
      <ms:projectName xml:lang="en">LINDAT/CLARIN -
↳Výzkumná infrastruktura pro jazykové technologie - rozšíření repozitáře a výpočetní
↳kapacity</ms:projectName>
      <ms:grantNumber>CZ.02.1.01/0.0/0.0/16_013/0001781</
↳ms:grantNumber>
      <ms:fundingType>http://w3id.org/meta-share/meta-share/
↳nationalFunds</ms:fundingType>
      <ms:funder>
        <ms:Organization>
          <ms:actorType>Organization</
↳ms:actorType>
        <ms:organizationName xml:lang="en">
↳Ministerstvo školství, mládeže a tělovýchovy České republiky</ms:organizationName>
          </ms:Organization>
        </ms:funder>
      </ms:fundingProject>
      <ms:LRSubclass>
        <ms:LanguageDescription>
          <ms:lrType>LanguageDescription</ms:lrType>
          <ms:LanguageDescriptionSubclass>
            <ms:MLModel>

```

(continues on next page)

(continued from previous page)

```

<ms:ldSubclassType>MlModel</
<ms:ldSubclassType>
    </ms:MLModel>
    </ms:LanguageDescriptionSubclass>
    <ms:LanguageDescriptionMediaPart>
        <ms:LanguageDescriptionTextPart>
            <ms:ldMediaType>
                <ms:mediaType>http://w3id.org/
                <ms:lingualityType>http://
                <ms:multilingualityType>http://
                <ms:language>
                    <ms:languageTag>en</
                    <ms:languageId>en</
                </ms:language>
            </ms:LanguageDescriptionTextPart>
        </ms:LanguageDescriptionMediaPart>
        <ms:DatasetDistribution>
            <ms:DatasetDistributionForm>http://
            <ms:downloadLocation>https://lindat.
            <ms:downloadLocation>
                <ms:accessLocation>http://hdl.handle.
            <ms:accessLocation>
                <ms:samplesLocation>http://lindat.mff.
            <ms:samplesLocation>
                <ms:distributionTextFeature>
                    <ms:size>
                        <ms:amount>9.1</
                    <ms:amount>
                        <ms:sizeUnit>http://
                    <ms:sizeUnit>
                        </ms:size>
                        <ms:dataFormat>http://w3id.
                    <ms:dataFormat>
                        </ms:distributionTextFeature>
                        <ms:licenceTerms>
                            <ms:licenceTermsName xml:lang=
                            <ms:licenceTermsName>
                                <ms:licenceTermsURL>http://
                                <ms:licenceTermsURL>
                                    <ms:conditionOfUse>http://
                                    <ms:conditionOfUse>http://
                                    <ms:conditionOfUse>http://
                                    <ms:conditionOfUse>http://
                                <ms:conditionOfUse>
                                    </ms:licenceTerms>
                                </ms:DatasetDistribution>
                                <ms:personalDataIncluded>false</
                            <ms:personalDataIncluded>

```

(continues on next page)

(continued from previous page)

```

<ms:sensitiveDataIncluded>false</
ms:sensitiveDataIncluded>
</ms:LanguageDescription>
</ms:LRSubclass>
</ms:LanguageResource>
</ms:DescribedEntity>
</ms:MetadataRecord>

```

Example 2: N-gram model

PANACEA Environment Corpus n-grams EL (Greek)

Published at <https://live.european-language-grid.eu/catalogue/#/resource/service/ld/900>

```

<?xml version="1.0" encoding="UTF-8"?>
<ms:MetadataRecord xmlns:ms="http://w3id.org/meta-share/meta-share/" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://w3id.org/meta-share/
meta-share/ ../Schema/ELG-SHARE.xsd">
  <ms:DescribedEntity>
    <ms:LanguageResource>
      <ms:entityType>LanguageResource</ms:entityType>
      <ms:resourceName xml:lang="en">PANACEA Environment Corpus n-
grams EL (Greek)</ms:resourceName>
      <ms:description xml:lang="en">PANACEA Environment Corpus n-
grams EL (Greek) 1.0 contains Greek word n-grams and Greek word/tag/lemma n-grams
in the "Environment" (ENV) domain. N-grams are accompanied by their observed
frequency counts. The length of the n-grams ranges from unigrams (single words) to
five-grams. The data were collected in the context of PANACEA (http://www.panacea-
lr.eu), an EU-FP7 Funded Project under Grant Agreement 248064.
The n-gram counts were generated from crawled Web pages that were automatically
detected to be in the Greek language and were automatically classified as relevant
to the ENV domain. The collection consisted of approximately 31.71 million tokens.
Data collection took place in the summer of 2011.</ms:description>
      <ms:version>1.0</ms:version>
      <ms:additionalInfo>
        <ms:landingPage>http://nlp.ilsp.gr/panacea/D4.3/data/
201209/gms/env_el/README.txt</ms:landingPage>
      </ms:additionalInfo>
      <ms:additionalInfo>
        <ms:email>contact@someDomain.com</ms:email>
      </ms:additionalInfo>
      <ms:contact>
        <ms:Person>
          <ms:actorType>Person</ms:actorType>
          <ms:surname xml:lang="en">Prokopidis</
ms:surname>
          <ms:givenName xml:lang="en">Prokopis</
ms:givenName>
          <ms:email>contact@someDomain.com</ms:email>
        </ms:Person>
      </ms:contact>
      <ms:contact>
        <ms:Person>
          <ms:actorType>Person</ms:actorType>
          <ms:surname xml:lang="en">Papavassiliou</
ms:surname>

```

(continues on next page)

(continued from previous page)

```

<ms:givenName xml:lang="en">Vassilis</
↪ms:givenName>
    <ms:email>contact@someDomain.com</ms:email>
    </ms:Person>
</ms:contact>
<ms:keyword xml:lang="en">corpus</ms:keyword>
<ms:domain>
    <ms:categoryLabel xml:lang="en">environment</
↪ms:categoryLabel>
</ms:domain>
<ms:resourceCreator>
    <ms:Organization>
        <ms:actorType>Organization</ms:actorType>
        <ms:organizationName xml:lang="en">Institute
↪for Language and Speech Processing</ms:organizationName>
        <ms:website>http://www.ilsp.gr</ms:website>
    </ms:Organization>
</ms:resourceCreator>
<ms:creationStartDate>2011-06-01</ms:creationStartDate>
<ms:creationEndDate>2011-08-31</ms:creationEndDate>
<ms:fundingProject>
    <ms:projectName xml:lang="en">Platform for Automatic,
↪Normalized Annotation and Cost-Effective Acquisition of Language Resources for
↪Human Language </ms:projectName>
    <ms:website>http://www.panacea-lr.eu</ms:website>
</ms:fundingProject>
<ms:LRS subclass>
    <ms:LanguageDescription>
        <ms:lrType>LanguageDescription</ms:lrType>
        <ms:LanguageDescriptionSubclass>
            <ms:NGramModel>
                <ms:ldSubclassType>NGramModel
↪</ms:ldSubclassType>
                <ms:baseItem>http://w3id.org/
↪meta-share/meta-share/word</ms:baseItem>
                <ms:order>5</ms:order>
            </ms:NGramModel>
        </ms:LanguageDescriptionSubclass>
        <ms:LanguageDescriptionMediaPart>
            <ms:LanguageDescriptionTextPart>
                <ms:ldMediaType>
↪LanguageDescriptionTextPart</ms:ldMediaType>
                <ms:mediaType>http://w3id.org/
↪meta-share/meta-share/text</ms:mediaType>
                <ms:lingualityType>http://
↪w3id.org/meta-share/meta-share/monolingual</ms:lingualityType>
                <ms:language>
                    <ms:languageTag>el</
↪ms:languageTag>
                    <ms:languageId>el</
↪ms:languageId>
                </ms:language>
                <ms:metalanguage>
                    <ms:languageTag>und</
↪ms:languageTag>
                    <ms:languageId>und</
↪ms:languageId>

```

(continues on next page)

(continued from previous page)

```

</ms:metallanguage>
<ms:creationDetails xml:lang=
↪ "en">automatic web crawling, automatic language detection, data preprocessing
↪ (boilerpipe filtering, lemmatization & tagging)</ms:creationDetails>
</ms:LanguageDescriptionTextPart>
</ms:LanguageDescriptionMediaPart>
<ms:DatasetDistribution>
<ms:DatasetDistributionForm>http://
↪ w3id.org/meta-share/meta-share/downloadable</ms:DatasetDistributionForm>
<ms:accessLocation>http://metashare.
↪ ilsp.gr:8080/repository/download/
↪ 490952dc1cec11e2b545842b2b6a04d78dc202de28d5421f91752610a781175e</ms:accessLocation>
<ms:distributionTextFeature>
<ms:size>
<ms:amount>435189</
↪ ms:amount>
<ms:sizeUnit>http://
↪ w3id.org/meta-share/meta-share/unigram</ms:sizeUnit>
</ms:size>
<ms:size>
<ms:amount>3.860716E6
↪ </ms:amount>
<ms:sizeUnit>http://
↪ w3id.org/meta-share/meta-share/bigram</ms:sizeUnit>
</ms:size>
<ms:size>
<ms:amount>9.767383E6
↪ </ms:amount>
<ms:sizeUnit>http://
↪ w3id.org/meta-share/meta-share/trigram</ms:sizeUnit>
</ms:size>
<ms:size>
<ms:amount>1.368394E7
↪ </ms:amount>
<ms:sizeUnit>http://
↪ w3id.org/meta-share/meta-share/four-gram</ms:sizeUnit>
</ms:size>
<ms:size>
<ms:amount>1.495402E7
↪ </ms:amount>
<ms:sizeUnit>http://
↪ w3id.org/meta-share/meta-share/five-gram</ms:sizeUnit>
</ms:size>
<ms:dataFormat>http://w3id.
↪ org/meta-share/omtd-share/Text</ms:dataFormat>
</ms:distributionTextFeature>
<ms:licenceTerms>
<ms:licenceTermsName xml:lang=
↪ "en">CC-BY-SA-4.0</ms:licenceTermsName>
<ms:licenceTermsURL>https://
↪ spdx.org/licenses/CC-BY-SA-4.0.html</ms:licenceTermsURL>
<ms:conditionOfUse>http://
↪ w3id.org/meta-share/meta-share/attribution</ms:conditionOfUse>
<ms:conditionOfUse>http://
↪ w3id.org/meta-share/meta-share/shareAlike</ms:conditionOfUse>
</ms:licenceTerms>
<ms:attributionText xml:lang="en">
↪ This LR has been created by Athena R.C./ILSP (www.ilsp.gr) and is licensed under
↪ CC-BY-SA licence</ms:attributionText>

```

(continued from previous page)

```

</ms:DatasetDistribution>
<ms:personalDataIncluded>>false</
↪ms:personalDataIncluded>
<ms:sensitiveDataIncluded>>false</
↪ms:sensitiveDataIncluded>
</ms:LanguageDescription>
</ms:LRSubclass>
</ms:LanguageResource>
</ms:DescribedEntity>
</ms:MetadataRecord>

```

1.15.4 3. Register the model at ELG

The current release of ELG offers two options for registering a catalogue item:

- the ELG **interactive editor** (see *Use the interactive editor*)
- the **upload of a metadata file** that conforms to the *ELG schema* in XML format (see *Create and upload metadata files*).

To upload the content files for the model, you can follow the procedure described here.

1.15.5 4. Manage and submit for publication

Through the “My items” page you can access your metadata record (see *Manage your items*) and edit it until you are satisfied. You can then *submit it for publication*, in line with the *publication lifecycle* defined for ELG metadata records.

At this stage, the metadata record can no longer be edited and is only visible to you and to us, the ELG platform administrators.

Before it is published, your submission undergoes a validation process, which is described in detail at *CHAPTER 4: VALIDATING ITEMS*.

Once approved, it will appear on the ELG catalogue and you will receive a notification email.

1.16 Contribute a grammar

This page describes how to contribute a *grammar* to the European Language Grid. You can describe a grammar and upload its contents at ELG or include in its description a link to the location it can be accessed from.

1.16.1 0. Before you start

- Please make sure that the grammar you want to contribute complies with our *terms of use*.
- Please make sure you have *registered* and been assigned the *provider role*.

1.16.2 1. Prepare the content files (for ELG hosted resources)

If you wish to upload the grammar at ELG, you must package it in a compressed format (currently as a .zip file).

If the files are available in multiple formats, (e.g. in XML, TXT and PDF formats), you are advised to package them in different zip files by data format.

1.16.3 2. Describe the grammar

Metadata overview

The grammar must be described according to the *ELG schema* and comply at least with the minimal version. The metadata elements that you need to provide for the grammar comprise a set of elements organized (for presentation purposes) into the following groups:

- *Minimal elements for all entities*
- *Minimal elements for all language resources and technologies*
- *Minimal elements for language descriptions*

Examples

Example: Monolingual computational grammar for a specific domain

Tourism Italian grammar

Published at: <https://live.european-language-grid.eu/catalogue/#/resource/service/ld/901>

```
<?xml version="1.0" encoding="UTF-8"?>
<ms:MetadataRecord xmlns:ms="http://w3id.org/meta-share/meta-share/" xmlns:xsi="http://
↪www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://w3id.org/meta-share/
↪meta-share/ ../Schema/ELG-SHARE.xsd">
  <ms:DescribedEntity>
    <ms:LanguageResource>
      <ms:entityType>LanguageResource</ms:entityType>
      <ms:resourceName xml:lang="en">Tourism Italian grammar</
↪ms:resourceName>
      <ms:resourceShortName xml:lang="en">Tour.ita.grm</
↪ms:resourceShortName>
      <ms:description xml:lang="en">Tourism Italian abnf grammar,
↪manually created. Created within the Portdial project</ms:description>
      <ms:version>1.0.0 (automatically assigned)</ms:version>
      <ms:additionalInfo>
        <ms:landingPage>https://sites.google.com/site/
↪portdial2</ms:landingPage>
      </ms:additionalInfo>
      <ms:additionalInfo>
        <ms:email>contact@someDomain.com</ms:email>
      </ms:additionalInfo>
      <ms:contact>
        <ms:Person>
          <ms:actorType>Person</ms:actorType>
          <ms:surname xml:lang="en">Potamianos</
↪ms:surname>
          <ms:givenName xml:lang="en">Alex</
↪ms:givenName>
```

(continues on next page)

(continued from previous page)

```

        <ms:email>contact@someDomain.com</ms:email>
      </ms:Person>
    </ms:contact>
    <ms:keyword xml:lang="en">languageDescription</ms:keyword>
    <ms:fundingProject>
      <ms:projectName xml:lang="en">Portdial</
↪ms:projectName>
    </ms:fundingProject>
    <ms:LRS subclass>
      <ms:LanguageDescription>
        <ms:lrType>LanguageDescription</ms:lrType>
        <ms:LanguageDescriptionSubclass>
          <ms:Grammar>
            <ms:ldSubclassType>Grammar</
↪ms:ldSubclassType>
            <ms:encodingLevel>http://w3id.
↪org/meta-share/meta-share/morphology</ms:encodingLevel>
          </ms:Grammar>
        </ms:LanguageDescriptionSubclass>
        <ms:LanguageDescriptionMediaPart>
          <ms:LanguageDescriptionTextPart>
            <ms:ldMediaType>
↪LanguageDescriptionTextPart</ms:ldMediaType>
            <ms:mediaType>http://w3id.org/
↪meta-share/meta-share/text</ms:mediaType>
            <ms:lingualityType>http://
↪w3id.org/meta-share/meta-share/monolingual</ms:lingualityType>
            <ms:language>
              <ms:languageTag>it</
↪ms:languageTag>
              <ms:languageId>it</
↪ms:languageId>
            </ms:language>
            <ms:metalanguage>
              <ms:languageTag>und</
↪ms:languageTag>
              <ms:languageId>und</
↪ms:languageId>
            </ms:metalanguage>
          </ms:LanguageDescriptionTextPart>
        </ms:LanguageDescriptionMediaPart>
        <ms:DatasetDistribution>
          <ms:DatasetDistributionForm>http://
↪w3id.org/meta-share/meta-share/downloadable</ms:DatasetDistributionForm>
          <ms:accessLocation>http://accessURL</
↪ms:accessLocation>
          <ms:licenceTerms>
            <ms:licenceTermsName xml:lang=
↪"en">CC-BY-SA-4.0</ms:licenceTermsName>
            <ms:licenceTermsURL>https://
↪spdx.org/licenses/CC-BY-SA-4.0.html</ms:licenceTermsURL>
            <ms:conditionOfUse>http://
↪w3id.org/meta-share/meta-share/attribution</ms:conditionOfUse>
            <ms:conditionOfUse>http://
↪w3id.org/meta-share/meta-share/shareAlike</ms:conditionOfUse>
          </ms:licenceTerms>
        </ms:DatasetDistribution>

```

(continues on next page)

(continued from previous page)

```

<ms:personalDataIncluded>>false</
↪ms:personalDataIncluded>
<ms:sensitiveDataIncluded>>false</
↪ms:sensitiveDataIncluded>
</ms:LanguageDescription>
</ms:LRS subclass>
</ms:LanguageResource>
</ms:DescribedEntity>
</ms:MetadataRecord>

```

1.16.4 3. Register the grammar at ELG

The current release of ELG offers two options for registering a catalogue item:

- the ELG **interactive editor** (see *Use the interactive editor*)
- the **upload of a metadata file** that conforms to the *ELG schema* in XML format (see *Create and upload metadata files*).

To upload the content files for the grammar, you can follow the procedure described here.

1.16.5 4. Manage and submit for publication

Through the “My items” page you can access your metadata record (see *Manage your items*) and edit it until you are satisfied. You can then *submit it for publication*, in line with the *publication lifecycle* defined for ELG metadata records.

At this stage, the metadata record can no longer be edited and is only visible to you and to us, the ELG platform administrators.

Before it is published, your submission undergoes a validation process, which is described in detail at *CHAPTER 4: VALIDATING ITEMS*.

Once approved, it will appear on the ELG catalogue and you will receive a notification email.

1.17 Contribute a lexical/conceptual resource

This page describes how to contribute a *lexical/conceptual resource* to the European Language Grid. You can describe it and upload its contents at ELG or include in its description a link to the location it can be accessed from.

1.17.1 0. Before you start

- Please make sure that the lexical/conceptual resource you want to contribute complies with our *terms of use*.
- Please make sure you have *registered* and been assigned the *provider role*.

1.17.2 1. Prepare the content files (for ELG hosted resources)

If you wish to upload the resource at ELG, you must package it in a compressed format (currently as a .zip file).

If the files are available in multiple formats, (e.g. in XML, TXT and PDF formats), you are advised to package them in different zip files by data format.

1.17.3 2. Describe the lexical/conceptual resource

Metadata overview

The lexical/conceptual resource must be described according to the *ELG schema* and comply at least with the minimal version. The metadata elements that you need to provide for the resource comprise a set of elements organized (for presentation purposes) into the following groups:

- *Minimal elements for all entities*
- *Minimal elements for all language resources and technologies*
- *Minimal elements for lexical/conceptual resources*

Examples

Example 1: Terminological lexicon

INTERA Corpus - the Bulgarian-English terms from the BG-EN pair

Published at: <https://live.european-language-grid.eu/catalogue/#/resource/service/lcr/694>

```
<?xml version="1.0" encoding="UTF-8"?>
<ms:MetadataRecord xmlns:ms="http://w3id.org/meta-share/meta-share/" xmlns:xsi="http://
↪www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://w3id.org/meta-share/
↪meta-share/ ../Schema/ELG-SHARE.xsd">
  <ms:DescribedEntity>
    <ms:LanguageResource>
      <ms:entityType>LanguageResource</ms:entityType>
      <ms:resourceName xml:lang="en">INTERA Corpus - the Bulgarian-
↪English terms from the BG-EN pair</ms:resourceName>
      <ms:description xml:lang="en">The Bulgarian-English terms
↪from the BG-EN pair of the INTERA corpus; written language, domain specific (law,
↪education).</ms:description>
      <ms:version>1.0.0 (automatically assigned)</ms:version>
      <ms:additionalInfo>
        <ms:email>contact@somedomain.com</ms:email>
      </ms:additionalInfo>
      <ms:contact>
        <ms:Person>
          <ms:actorType>Person</ms:actorType>
          <ms:surname xml:lang="en">Gavrilidou</
↪ms:surname>
          <ms:givenName xml:lang="en">Maria</
↪ms:givenName>
          <ms:email>contact@somedomain.com</ms:email>
        </ms:Person>
      </ms:contact>
      <ms:keyword xml:lang="en">lexicalconceptualresource</
↪ms:keyword>
```

(continues on next page)

(continued from previous page)

```

        <ms:domain>
            <ms:categoryLabel xml:lang="en">education</
↪ms:categoryLabel>
        </ms:domain>
        <ms:domain>
            <ms:categoryLabel xml:lang="en">law</ms:categoryLabel>
        </ms:domain>
        <ms:creationStartDate>2003-01-01</ms:creationStartDate>
        <ms:creationEndDate>2004-12-31</ms:creationEndDate>
        <ms:fundingProject>
            <ms:projectName xml:lang="en">Integrated European_
↪language data Repository Area</ms:projectName>
            <ms:website>http://www.elda.org/intera</ms:website>
        </ms:fundingProject>
        <ms:intendedApplication>
            <ms:LTCClassOther>machineTranslation</ms:LTCClassOther>
        </ms:intendedApplication>
        <ms:actualUse>
            <ms:usedInApplication>
                <ms:LTCClassOther>terminologyExtraction</
↪ms:LTCClassOther>
            </ms:usedInApplication>
            <ms:actualUseDetails xml:lang="en">nlpApplications</
↪ms:actualUseDetails>
        </ms:actualUse>
        <ms:isDocumentedBy>
            <ms:title xml:lang="en">Building Multilingual_
↪Terminological Resources</ms:title>
        </ms:isDocumentedBy>
        <ms:isDocumentedBy>
            <ms:title xml:lang="en">Building parallel corpora for_
↪eContent professionals</ms:title>
        </ms:isDocumentedBy>
        <ms:isDocumentedBy>
            <ms:title xml:lang="en">Language resources production_
↪models: the case of INTERA multilingual corpus and terminology</ms:title>
        </ms:isDocumentedBy>
        <ms:isDocumentedBy>
            <ms:title xml:lang="en">D5.2 - Report on the_
↪multilingual resources production</ms:title>
            <ms:DocumentIdentifier ms:DocumentIdentifierScheme=
↪"http://purl.org/spar/datacite/url">http://www.elda.org/article176.html</
↪ms:DocumentIdentifier>
        </ms:isDocumentedBy>
        <ms:relation>
            <ms:relationType xml:lang="en">isExtractedfrom</
↪ms:relationType>
            <ms:relatedLR>
                <ms:resourceName xml:lang="en">INTERA corpus</
↪ms:resourceName>
            </ms:relatedLR>
        </ms:relation>
        <ms:LRS subclass>
            <ms:LexicalConceptualResource>
                <ms:lrType>LexicalConceptualResource</
↪ms:lrType>
            <ms:lcrSubclass>http://w3id.org/meta-share/
↪meta-share/wordlist</ms:lcrSubclass>

```

(continues on next page)

(continued from previous page)

```

<ms:encodingLevel>http://w3id.org/meta-share/
↪meta-share/morphology</ms:encodingLevel>
      <ms:LexicalConceptualResourceMediaPart>
        <ms:LexicalConceptualResourceTextPart>
          <ms:lcrMediaType>
↪LexicalConceptualResourceTextPart</ms:lcrMediaType>
          <ms:mediaType>http://w3id.org/
↪meta-share/meta-share/text</ms:mediaType>
          <ms:lingualityType>http://
↪w3id.org/meta-share/meta-share/bilingual</ms:lingualityType>
          <ms:multilingualityType>http://
↪w3id.org/meta-share/meta-share/parallel</ms:multilingualityType>
          <ms:language>
            <ms:languageTag>bg</
↪ms:languageTag>
            <ms:languageId>bg</
↪ms:languageId>
          </ms:language>
          <ms:language>
            <ms:languageTag>en</
↪ms:languageTag>
            <ms:languageId>en</
↪ms:languageId>
          </ms:language>
          <ms:metalanguage>
            <ms:languageTag>und</
↪ms:languageTag>
            <ms:languageId>und</
↪ms:languageId>
          </ms:metalanguage>
          <ms:modalityType>http://w3id.
↪org/meta-share/meta-share/writtenLanguage</ms:modalityType>
        </ms:LexicalConceptualResourceTextPart>
      </ms:LexicalConceptualResourceMediaPart>
      <ms:DatasetDistribution>
        <ms:DatasetDistributionForm>http://
↪w3id.org/meta-share/meta-share/downloadable</ms:DatasetDistributionForm>
        <ms:accessLocation>http://metashare.
↪ilsp.gr:8080/repository/download/
↪cdba66329e8111e581e1842b2b6a04d770c91fa84de04f259240aa450aaa9081</ms:accessLocation>
        <ms:distributionTextFeature>
          <ms:size>
            <ms:amount>7581</
↪ms:amount>
            <ms:sizeUnit>http://
↪w3id.org/meta-share/meta-share/word3</ms:sizeUnit>
          </ms:size>
          <ms:dataFormat>http://w3id.
↪org/meta-share/omtd-share/Text</ms:dataFormat>
        </ms:distributionTextFeature>
        <ms:licenceTerms>
          <ms:licenceTermsName xml:lang=
↪"en">CC-BY-4.0</ms:licenceTermsName>
          <ms:licenceTermsURL>https://
↪spdx.org/licenses/CC-BY-4.0.html</ms:licenceTermsURL>
          <ms:conditionOfUse>http://
↪w3id.org/meta-share/meta-share/attribution</ms:conditionOfUse>

```

(continues on next page)

(continued from previous page)

```

</ms:licenceTerms>
  <ms:attributionText xml:lang="en">The
↪INTERA Corpus - the Bulgarian-English terms from the BG-EN pair of the ILSP/RC
↪Athena licensed under CC-BY as accessed via META-SHARE</ms:attributionText>
</ms:DatasetDistribution>
<ms:personalDataIncluded>>false</
↪ms:personalDataIncluded>
  <ms:sensitiveDataIncluded>>false</
↪ms:sensitiveDataIncluded>
    </ms:LexicalConceptualResource>
  </ms:LRSubclass>
</ms:LanguageResource>
</ms:DescribedEntity>
</ms:MetadataRecord>

```

Example 2: Computational lexicon

MCL - Multifunctional Computational Lexicon of Contemporary Portuguese

Published at: <https://live.european-language-grid.eu/catalogue/#/resource/service/lcr/918>

```

<?xml version="1.0" encoding="UTF-8"?>
<ms:MetadataRecord xmlns:ms="http://w3id.org/meta-share/meta-share/" xmlns:xsi="http://
↪www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://w3id.org/meta-share/
↪meta-share/ ../Schema/ELG-SHARE.xsd">
  <ms:DescribedEntity>
    <ms:LanguageResource>
      <ms:entityType>LanguageResource</ms:entityType>
      <ms:resourceName xml:lang="en">MCL - Multifunctional
↪Computational Lexicon of Contemporary Portuguese</ms:resourceName>
      <ms:description xml:lang="en">MCL is a 26,443 lemma Frequency
↪Lexicon with 140,315 tokens, with the minimum lemma frequency of 6, extracted from
↪CORLEX, a contemporary Portuguese corpus (16,210,438 words). CORLEX is a subcorpus
↪of the Reference Corpus of Contemporary Portuguese and contains written and spoken
↪texts of several types, being genre diversity a characteristic of this corpus.
↪CORLEX contains mainly journalistic texts (56% of the written corpus and 53% of the
↪whole corpus). In order to extract the lexicon, all the different lexical forms
↪occurring in the corpus were indexed and subsequently tagged morphosyntactically
↪and lemmatised by PALAVROSO. Each lemma in MCL is followed by morphosyntactic and
↪quantitative information. The same information is given regarding each lemma token
↪(inflected forms and some compounds). The lexicon indexations are listed in
↪alphabetical order or decreasing frequency order.</ms:description>
      <ms:LRIdentifier ms:LRIdentifierScheme="http://w3id.org/meta-
↪share/meta-share/islrn">489-956-642-755-8</ms:LRIdentifier>
      <ms:LRIdentifier ms:LRIdentifierScheme="http://w3id.org/meta-
↪share/meta-share/other">ELRA-L0096</ms:LRIdentifier>
      <ms:version>1.0</ms:version>
      <ms:versionDate>2016-01-20</ms:versionDate>
      <ms:additionalInfo>
        <ms:landingPage>http://catalog.elra.info/product_info.
↪php?products_id=1254</ms:landingPage>
      </ms:additionalInfo>
      <ms:additionalInfo>
        <ms:email>contact@somedomain.com</ms:email>
      </ms:additionalInfo>

```

(continues on next page)

(continued from previous page)

```

<ms:keyword xml:lang="en">lexicalconceptualresource</
↪ms:keyword>
    <ms:LRSubclass>
        <ms:LexicalConceptualResource>
            <ms:lrType>LexicalConceptualResource</
↪ms:lrType>
                <ms:lcrSubclass>http://w3id.org/meta-share/
↪meta-share/lexicon</ms:lcrSubclass>
                <ms:encodingLevel>http://w3id.org/meta-share/
↪meta-share/morphology</ms:encodingLevel>
                <ms:encodingLevel>http://w3id.org/meta-share/
↪meta-share/syntax</ms:encodingLevel>
                <ms:LexicalConceptualResourceMediaPart>
                    <ms:LexicalConceptualResourceTextPart>
                        <ms:lcrMediaType>
↪LexicalConceptualResourceTextPart</ms:lcrMediaType>
                            <ms:mediaType>http://w3id.org/
↪meta-share/meta-share/text</ms:mediaType>
                                <ms:lingualityType>http://
↪w3id.org/meta-share/meta-share/monolingual</ms:lingualityType>
                                    <ms:language>
                                        <ms:languageTag>pt</
↪ms:languageTag>
                                            <ms:languageId>pt</
↪ms:languageId>
                                                </ms:language>
                                                <ms:metalanguage>
                                                    <ms:languageTag>und</
↪ms:languageTag>
                                                        <ms:languageId>und</
↪ms:languageId>
                                                            </ms:metalanguage>
                                                                </
↪ms:LexicalConceptualResourceTextPart>
                                                                    </ms:LexicalConceptualResourceMediaPart>
                                                                    <ms:DatasetDistribution>
                                                                        <ms:DatasetDistributionForm>http://
↪w3id.org/meta-share/meta-share/downloadable</ms:DatasetDistributionForm>
                                                                            <ms:distributionTextFeature>
                                                                                <ms:size>
                                                                                    <ms:amount>26443</
↪ms:amount>
                                                                                        <ms:sizeUnit>http://
↪w3id.org/meta-share/meta-share/entry</ms:sizeUnit>
                                                                                            </ms:size>
                                                                                            <ms:dataFormat>http://w3id.
↪org/meta-share/omtd-share/Pdf</ms:dataFormat>
                                                                                                <ms:dataFormat>http://w3id.
↪org/meta-share/omtd-share/Text</ms:dataFormat>
                                                                                                    </ms:distributionTextFeature>
                                                                                                    <ms:licenceTerms>
                                                                                                        <ms:licenceTermsName xml:lang=
↪"en">ELRA-VAR-ACADEMIC-MEMBER-COMMERCIALUSE-1.0</ms:licenceTermsName>
                                                                                                            <ms:licenceTermsURL>http://
↪www.elra.info/licenses/ELRA-VAR-ACADEMIC-MEMBER-COMMERCIALUSE-1.0.html</
↪ms:licenceTermsURL>
                                                                                                                <ms:conditionOfUse>http://
↪w3id.org/meta-share/meta-share/attribution</ms:conditionOfUse>

```

(continues on next page)

(continued from previous page)

```

</ms:licenceTerms>
<ms:availabilityStartDate>2016-01-20</
↪ms:availabilityStartDate>

<ms:distributionRightsHolder>
  <ms:Organization>
    <ms:actorType>
↪Organization</ms:actorType>
    <ms:organizationName
↪xml:lang="en">ELRA</ms:organizationName>
    <ms:website>http://
↪www.elra.info/en/</ms:website>
  </ms:Organization>
</ms:distributionRightsHolder>
</ms:DatasetDistribution>
<ms:DatasetDistribution>
  <ms:DatasetDistributionForm>http://
↪w3id.org/meta-share/meta-share/downloadable</ms:DatasetDistributionForm>
  <ms:distributionTextFeature>
    <ms:size>
      <ms:amount>26443</
↪ms:amount>
      <ms:sizeUnit>http://
↪w3id.org/meta-share/meta-share/entry</ms:sizeUnit>
    </ms:size>
    <ms:dataFormat>http://w3id.
↪org/meta-share/omtd-share/Pdf</ms:dataFormat>
    <ms:dataFormat>http://w3id.
↪org/meta-share/omtd-share/Text</ms:dataFormat>
  </ms:distributionTextFeature>
  <ms:licenceTerms>
    <ms:licenceTermsName xml:lang=
↪"en">ELRA-END-USER-ACADEMIC-MEMBER-NONCOMMERCIALUSE-1.0</ms:licenceTermsName>
    <ms:licenceTermsURL>http://
↪www.elra.info/licenses/ELRA-END-USER-ACADEMIC-MEMBER-NONCOMMERCIALUSE-1.0.html</
↪ms:licenceTermsURL>
    <ms:conditionOfUse>http://
↪w3id.org/meta-share/meta-share/nonCommercialUse</ms:conditionOfUse>
    <ms:conditionOfUse>http://
↪w3id.org/meta-share/meta-share/attribution</ms:conditionOfUse>
  </ms:licenceTerms>
  <ms:availabilityStartDate>2016-01-20</
↪ms:availabilityStartDate>

  <ms:distributionRightsHolder>
    <ms:Organization>
      <ms:actorType>
↪Organization</ms:actorType>
      <ms:organizationName
↪xml:lang="en">ELRA</ms:organizationName>
      <ms:website>http://
↪www.elra.info/en/</ms:website>
    </ms:Organization>
  </ms:distributionRightsHolder>
</ms:DatasetDistribution>
<ms:DatasetDistribution>
  <ms:DatasetDistributionForm>http://
↪w3id.org/meta-share/meta-share/downloadable</ms:DatasetDistributionForm>
  <ms:distributionTextFeature>

```

(continues on next page)

(continued from previous page)

```

<ms:size>
  <ms:amount>26443</
↪ms:amount>
  <ms:sizeUnit>http://
↪w3id.org/meta-share/meta-share/entry</ms:sizeUnit>
  </ms:size>
  <ms:dataFormat>http://w3id.
↪org/meta-share/omtd-share/Pdf</ms:dataFormat>
  <ms:dataFormat>http://w3id.
↪org/meta-share/omtd-share/Text</ms:dataFormat>
  </ms:distributionTextFeature>
  <ms:licenceTerms>
    <ms:licenceTermsName xml:lang=
↪"en">ELRA-VAR-COMMERCIAL-MEMBER-COMMERCIALUSE-1.0</ms:licenceTermsName>
    <ms:licenceTermsURL>http://
↪www.elra.info/licenses/ELRA-VAR-COMMERCIAL-MEMBER-COMMERCIALUSE-1.0.html</
↪ms:licenceTermsURL>
    <ms:conditionOfUse>http://
↪w3id.org/meta-share/meta-share/attribution</ms:conditionOfUse>
  </ms:licenceTerms>
  <ms:availabilityStartDate>2016-01-20</
↪ms:availabilityStartDate>
  <ms:distributionRightsHolder>
    <ms:Organization>
      <ms:actorType>
↪Organization</ms:actorType>
      <ms:organizationName_
↪xml:lang="en">ELRA</ms:organizationName>
      <ms:website>http://
↪www.elra.info/en/</ms:website>
    </ms:Organization>
  </ms:distributionRightsHolder>
</ms:DatasetDistribution>
<ms:DatasetDistribution>
  <ms:DatasetDistributionForm>http://
↪w3id.org/meta-share/meta-share/downloadable</ms:DatasetDistributionForm>
  <ms:distributionTextFeature>
    <ms:size>
      <ms:amount>26443</
↪ms:amount>
      <ms:sizeUnit>http://
↪w3id.org/meta-share/meta-share/entry</ms:sizeUnit>
    </ms:size>
    <ms:dataFormat>http://w3id.
↪org/meta-share/omtd-share/Pdf</ms:dataFormat>
    <ms:dataFormat>http://w3id.
↪org/meta-share/omtd-share/Text</ms:dataFormat>
    </ms:distributionTextFeature>
    <ms:licenceTerms>
      <ms:licenceTermsName xml:lang=
↪"en">ELRA-END-USER-COMMERCIAL-MEMBER-NONCOMMERCIALUSE-1.0</ms:licenceTermsName>
      <ms:licenceTermsURL>http://
↪www.elra.info/licenses/ELRA-END-USER-COMMERCIAL-MEMBER-NONCOMMERCIALUSE-1.0.html</
↪ms:licenceTermsURL>
      <ms:conditionOfUse>http://
↪w3id.org/meta-share/meta-share/academicUser</ms:conditionOfUse>
      <ms:conditionOfUse>http://
↪w3id.org/meta-share/meta-share/attribution</ms:conditionOfUse>

```

(continues on next page)

(continued from previous page)

```

</ms:licenceTerms>
<ms:availabilityStartDate>2016-01-20</
↪ms:availabilityStartDate>

<ms:distributionRightsHolder>
  <ms:Organization>
    <ms:actorType>
↪Organization</ms:actorType>
    <ms:organizationName
↪xml:lang="en">ELRA</ms:organizationName>
    <ms:website>http://
↪www.elra.info/en/</ms:website>
  </ms:Organization>
</ms:distributionRightsHolder>
</ms:DatasetDistribution>
<ms:DatasetDistribution>
  <ms:DatasetDistributionForm>http://
↪w3id.org/meta-share/meta-share/downloadable</ms:DatasetDistributionForm>
  <ms:distributionTextFeature>
    <ms:size>
      <ms:amount>26443</
↪ms:amount>
      <ms:sizeUnit>http://
↪w3id.org/meta-share/meta-share/entry</ms:sizeUnit>
    </ms:size>
    <ms:dataFormat>http://w3id.
↪org/meta-share/omtd-share/Pdf</ms:dataFormat>
    <ms:dataFormat>http://w3id.
↪org/meta-share/omtd-share/Text</ms:dataFormat>
  </ms:distributionTextFeature>
  <ms:licenceTerms>
    <ms:licenceTermsName xml:lang=
↪"en">ELRA-VAR-ACADEMIC-NOMEMBER-COMMERCIALUSE-1.0</ms:licenceTermsName>
    <ms:licenceTermsURL>http://
↪www.elra.info/licenses/ELRA-VAR-ACADEMIC-NOMEMBER-COMMERCIALUSE-1.0.html</
↪ms:licenceTermsURL>
    <ms:conditionOfUse>http://
↪w3id.org/meta-share/meta-share/attribution</ms:conditionOfUse>
  </ms:licenceTerms>
  <ms:availabilityStartDate>2016-01-20</
↪ms:availabilityStartDate>

  <ms:distributionRightsHolder>
    <ms:Organization>
      <ms:actorType>
↪Organization</ms:actorType>
      <ms:organizationName
↪xml:lang="en">ELRA</ms:organizationName>
      <ms:website>http://
↪www.elra.info/en/</ms:website>
    </ms:Organization>
  </ms:distributionRightsHolder>
</ms:DatasetDistribution>
<ms:DatasetDistribution>
  <ms:DatasetDistributionForm>http://
↪w3id.org/meta-share/meta-share/downloadable</ms:DatasetDistributionForm>
  <ms:distributionTextFeature>
    <ms:size>
      <ms:amount>26443</
↪ms:amount>

```

(continues on next page)

(continued from previous page)

```

<ms:sizeUnit>http://
↪w3id.org/meta-share/meta-share/entry</ms:sizeUnit>
</ms:size>
<ms:dataFormat>http://w3id.
↪org/meta-share/omtd-share/Pdf</ms:dataFormat>
<ms:dataFormat>http://w3id.
↪org/meta-share/omtd-share/Text</ms:dataFormat>
</ms:distributionTextFeature>
<ms:licenceTerms>
  <ms:licenceTermsName xml:lang=
↪"en">ELRA-END-USER-ACADEMIC-NOMEMBER-NONCOMMERCIALUSE-1.0</ms:licenceTermsName>
  <ms:licenceTermsURL>http://
↪www.elra.info/licenses/ELRA-END-USER-ACADEMIC-NOMEMBER-NONCOMMERCIALUSE-1.0.html</
↪ms:licenceTermsURL>
  <ms:conditionOfUse>http://
↪w3id.org/meta-share/meta-share/attribution</ms:conditionOfUse>
  <ms:conditionOfUse>http://
↪w3id.org/meta-share/meta-share/nonCommercialUse</ms:conditionOfUse>
</ms:licenceTerms>
<ms:availabilityStartDate>2016-01-20</
↪ms:availabilityStartDate>
  <ms:distributionRightsHolder>
    <ms:Organization>
      <ms:actorType>
↪Organization</ms:actorType>
      <ms:organizationName_
↪xml:lang="en">ELRA</ms:organizationName>
      <ms:website>http://
↪www.elra.info/en/</ms:website>
    </ms:Organization>
  </ms:distributionRightsHolder>
</ms:DatasetDistribution>
<ms:DatasetDistribution>
  <ms:DatasetDistributionForm>http://
↪w3id.org/meta-share/meta-share/downloadable</ms:DatasetDistributionForm>
  <ms:distributionTextFeature>
    <ms:size>
      <ms:amount>26443</
↪ms:amount>
      <ms:sizeUnit>http://
↪w3id.org/meta-share/meta-share/entry</ms:sizeUnit>
    </ms:size>
    <ms:dataFormat>http://w3id.
↪org/meta-share/omtd-share/Pdf</ms:dataFormat>
    <ms:dataFormat>http://w3id.
↪org/meta-share/omtd-share/Text</ms:dataFormat>
  </ms:distributionTextFeature>
  <ms:licenceTerms>
    <ms:licenceTermsName xml:lang=
↪"en">ELRA-VAR-COMMERCIAL-NOMEMBER-COMMERCIALUSE-1.0</ms:licenceTermsName>
    <ms:licenceTermsURL>http://
↪www.elra.info/licenses/ELRA-VAR-COMMERCIAL-NOMEMBER-COMMERCIALUSE-1.0.html</
↪ms:licenceTermsURL>
    <ms:conditionOfUse>http://
↪w3id.org/meta-share/meta-share/attribution</ms:conditionOfUse>
  </ms:licenceTerms>
  <ms:availabilityStartDate>2016-01-20</
↪ms:availabilityStartDate>

```

(continues on next page)

(continued from previous page)

```

<ms:distributionRightsHolder>
  <ms:Organization>
    <ms:actorType>
↪Organization</ms:actorType>
    <ms:organizationName_
↪xml:lang="en">ELRA</ms:organizationName>
    <ms:website>http://
↪www.elra.info/en/</ms:website>
  </ms:Organization>
</ms:distributionRightsHolder>
</ms:DatasetDistribution>
<ms:DatasetDistribution>
  <ms:DatasetDistributionForm>http://
↪w3id.org/meta-share/meta-share/downloadable</ms:DatasetDistributionForm>
  <ms:distributionTextFeature>
    <ms:size>
      <ms:amount>26443</
↪ms:amount>
      <ms:sizeUnit>http://
↪w3id.org/meta-share/meta-share/entry</ms:sizeUnit>
    </ms:size>
    <ms:dataFormat>http://w3id.
↪org/meta-share/omtd-share/Pdf</ms:dataFormat>
    <ms:dataFormat>http://w3id.
↪org/meta-share/omtd-share/Text</ms:dataFormat>
  </ms:distributionTextFeature>
  <ms:licenceTerms>
    <ms:licenceTermsName xml:lang=
↪"en">ELRA-END-USER-COMMERCIAL-NOMEMBER-NONCOMMERCIALUSE-1.0</ms:licenceTermsName>
    <ms:licenceTermsURL>http://
↪www.elra.info/licenses/ELRA-END-USER-COMMERCIAL-NOMEMBER-NONCOMMERCIALUSE-1.0.html</
↪ms:licenceTermsURL>
    <ms:conditionOfUse>http://
↪w3id.org/meta-share/meta-share/attribution</ms:conditionOfUse>
    <ms:conditionOfUse>http://
↪w3id.org/meta-share/meta-share/nonCommercialUse</ms:conditionOfUse>
  </ms:licenceTerms>
  <ms:availabilityStartDate>2016-01-20</
↪ms:availabilityStartDate>
  <ms:distributionRightsHolder>
    <ms:Organization>
      <ms:actorType>
↪Organization</ms:actorType>
      <ms:organizationName_
↪xml:lang="en">ELRA</ms:organizationName>
      <ms:website>http://
↪www.elra.info/en/</ms:website>
    </ms:Organization>
  </ms:distributionRightsHolder>
</ms:DatasetDistribution>
<ms:personalDataIncluded>>false</
↪ms:personalDataIncluded>
  <ms:sensitiveDataIncluded>>false</
↪ms:sensitiveDataIncluded>
</ms:LexicalConceptualResource>
</ms:LRSubclass>
</ms:LanguageResource>

```

(continues on next page)

(continued from previous page)

```
</ms:DescribedEntity>
</ms:MetadataRecord>
```

1.17.4 3. Register the lexical/conceptual resource at ELG

The current release of ELG offers two options for registering a catalogue item:

- the ELG **interactive editor** (see *Use the interactive editor*)
- the **upload of a metadata file** that conforms to the *ELG schema* in XML format (see *Create and upload metadata files*).

To upload the content files for the lexical/conceptual resource, you can follow the procedure described here.

1.17.5 4. Manage and submit for publication

Through the “My items” page you can access your metadata record (see *Manage your items*) and edit it until you are satisfied. You can then *submit it for publication*, in line with the *publication lifecycle* defined for ELG metadata records.

Before it is published, your submission undergoes a validation process, which is described in detail at *CHAPTER 4: VALIDATING ITEMS*.

Once approved, it will appear on the ELG catalogue and you will receive a notification email.

1.18 Contribute an organization

This page describes how to contribute information on an *organization* to the European Language Grid.

1.18.1 0. Before you start

- Please make sure that the metadata record you want to contribute complies with our *terms of use*.
- Please make sure you have *registered* and been assigned the *provider role*.

1.18.2 2. Describe the organization

Metadata overview

The organization must be described according to the *ELG schema* and comply at least with the minimal version. The metadata elements that you need to provide for the organization comprise a set of elements organized (for presentation purposes) into the following groups:

- *Minimal elements for all entities*
- *Minimal elements for organizations*

Examples

Example 1: University

Charles University, Prague

Published at: <https://live.european-language-grid.eu/catalogue/#/resource/organizations/385>

```

<?xml version="1.0" encoding="UTF-8"?>
<ms:MetadataRecord xsi:schemaLocation="http://w3id.org/meta-share/meta-share/ ../
↳ Schema/ELG-SHARE.xsd" xmlns:ms="http://w3id.org/meta-share/meta-share/" xmlns:xsi=
↳ "http://www.w3.org/2001/XMLSchema-instance">
  <ms:DescribedEntity>
    <ms:Organization>
      <ms:entityType>Organization</ms:entityType>
      <ms:OrganizationIdentifier ms:OrganizationIdentifierScheme="http://w3id.org/
↳ meta-share/meta-share/elg">automatically assigned by ELG - please don't change</
↳ ms:OrganizationIdentifier>
        <ms:organizationName xml:lang="en">Charles University</ms:organizationName>
        <ms:organizationShortName xml:lang="en">CUNI</ms:organizationShortName>
        <ms:organizationAlternativeName xml:lang="en">UNIVERZITA KARLOVA</
↳ ms:organizationAlternativeName>
          <ms:organizationRole>http://w3id.org/meta-share/meta-share/LTSupplier</
↳ ms:organizationRole>
            <ms:organizationRole>http://w3id.org/meta-share/meta-share/
↳ researchOrganization</ms:organizationRole>
            <ms:organizationRole>http://w3id.org/meta-share/meta-share/
↳ languageServiceProvider</ms:organizationRole>
            <ms:organizationLegalStatus>http://w3id.org/meta-share/meta-share/
↳ academicInstitution</ms:organizationLegalStatus>
            <ms:organizationBio xml:lang="en">Charles University was founded in 1348,
↳ making it one of the oldest universities in the world. Yet it is also renowned as a
↳ modern, dynamic, cosmopolitan and prestigious institution of higher education. It
↳ is the largest and most renowned Czech university, and is also the best-rated Czech
↳ university according to international rankings. There are currently 17 faculties at
↳ the University, plus 3 institutes, 6 other centres of teaching, research,
↳ development and other creative activities, a centre providing information services,
↳ 5 facilities serving the whole University, and the Rectorate - which is the
↳ executive management body for the whole University.</ms:organizationBio>
            <ms:logo>https://cuni.cz/UKEN-1-version1-afoto.jpg</ms:logo>
            <ms:LTArea>
              <ms:LTClassRecommended>http://w3id.org/meta-share/omtd-share/
↳ LanguageTechnology</ms:LTClassRecommended>
              </ms:LTArea>
              <ms:LTArea>
                <ms:LTClassRecommended>http://w3id.org/meta-share/omtd-share/
↳ MachineTranslation</ms:LTClassRecommended>
                </ms:LTArea>
                <ms:LTArea>
                  <ms:LTClassRecommended>http://w3id.org/meta-share/omtd-share/
↳ SpeechRecognition</ms:LTClassRecommended>
                  </ms:LTArea>
                  <ms:LTArea>
                    <ms:LTClassRecommended>http://w3id.org/meta-share/omtd-share/
↳ Annotation</ms:LTClassRecommended>
                    </ms:LTArea>
                    <ms:LTArea>
                      <ms:LTClassRecommended>http://w3id.org/meta-share/omtd-share/
↳ LexiconCreation</ms:LTClassRecommended>

```

(continues on next page)

(continued from previous page)

```

</ms:LTArea>
<ms:LTArea>
  <ms:LTCClassOther>language resources creation</ms:LTCClassOther>
</ms:LTArea>
<ms:LTArea>
  <ms:LTCClassOther>Dialog systems</ms:LTCClassOther>
</ms:LTArea>
<ms:keyword xml:lang="en">Computational Linguistics</ms:keyword>
<ms:keyword xml:lang="en">Natural Language Processing</ms:keyword>
<ms:keyword xml:lang="en">Language Resources</ms:keyword>
<ms:keyword xml:lang="en">Research infrastructures</ms:keyword>
<ms:keyword xml:lang="en">Language Resources</ms:keyword>
<ms:keyword xml:lang="en">Digital Humanities</ms:keyword>
<ms:website>https://www.cuni.cz</ms:website>
<ms:headOfficeAddress>
  <ms:address xml:lang="en">OVOCNY TRH 560/5</ms:address>
  <ms:zipCode>116 36</ms:zipCode>
  <ms:city xml:lang="en">PRAHA 1</ms:city>
  <ms:country>CZ</ms:country>
</ms:headOfficeAddress>

</ms:Organization>
</ms:DescribedEntity>
</ms:MetadataRecord>

```

Example 2: University division

Institute of Formal and Applied Linguistics

```

<?xml version="1.0" encoding="UTF-8"?>
<ms:MetadataRecord xsi:schemaLocation="http://w3id.org/meta-share/meta-share/ ../
→Schema/ELG-SHARE.xsd" xmlns:ms="http://w3id.org/meta-share/meta-share/" xmlns:xsi=
→"http://www.w3.org/2001/XMLSchema-instance">
  <ms:DescribedEntity>
    <ms:Organization>
      <ms:entityType>Organization</ms:entityType>
      <ms:organizationName xml:lang="en">Institute of Formal and
→Applied Linguistics</ms:organizationName>
      <ms:organizationShortName xml:lang="en">UFAL</
→ms:organizationShortName>
      <ms:organizationRole>http://w3id.org/meta-share/meta-share/
→LTSupplier</ms:organizationRole>
      <ms:organizationRole>http://w3id.org/meta-share/meta-share/
→researchOrganization</ms:organizationRole>
      <ms:organizationRole>http://w3id.org/meta-share/meta-share/
→languageServiceProvider</ms:organizationRole>
      <ms:organizationLegalStatus>http://w3id.org/meta-share/meta-
→share/academicInstitution</ms:organizationLegalStatus>
      <ms:organizationBio xml:lang="en">'Institute of Formal and
→Applied Linguistics (ÚFAL) at the Computer Science School, Faculty of Mathematics
→and Physics, Charles University, Czech Republic. The institute was established in
→1990 as a continuation of the research and teaching activities carried out by the
→former Laboratory of Algebraic Linguistics since the early 60s at the Faculty of
→Philosophy and later at the Faculty of Mathematics and Physics, Charles University.
→The Institute is a primarily research department working on many topics in the area
→of Computational Linguistics, and on many research projects both nationally and
→internationally. However, the Institute of Formal and Applied Linguistics is also a
→regular department in the sense that it carries a comprehensive teaching program.

```

1.18. Contribute an organization

Free (Mgr., or MSc.) as well as for a doctorate (Ph.D.) in Computational Linguistics. Both programs are taught in Czech and English. The Institute is also a member of the double-degree "Master's LCT programme" of the EU. Students also can take advantage of the Erasmus program for typically semester-long stays at partner Universities abroad. ' </ms:organizationBio>

(continued from previous page)

```

        <ms:LTArea>
            <ms:LTCClassRecommended>http://w3id.org/meta-share/
↪ omdt-share/LanguageTechnology</ms:LTCClassRecommended>
        </ms:LTArea>
        <ms:LTArea>
            <ms:LTCClassRecommended>http://w3id.org/meta-share/
↪ omdt-share/MachineTranslation</ms:LTCClassRecommended>
        </ms:LTArea>
        <ms:LTArea>
            <ms:LTCClassRecommended>http://w3id.org/meta-share/
↪ omdt-share/SpeechRecognition</ms:LTCClassRecommended>
        </ms:LTArea>
        <ms:LTArea>
            <ms:LTCClassRecommended>http://w3id.org/meta-share/
↪ omdt-share/Annotation</ms:LTCClassRecommended>
        </ms:LTArea>
        <ms:LTArea>
            <ms:LTCClassRecommended>http://w3id.org/meta-share/
↪ omdt-share/LexiconCreation</ms:LTCClassRecommended>
        </ms:LTArea>
        <ms:LTArea>
            <ms:LTCClassOther>language resources creation</
↪ ms:LTCClassOther>
        </ms:LTArea>
        <ms:LTArea>
            <ms:LTCClassOther>Dialog systems</ms:LTCClassOther>
        </ms:LTArea>
        <ms:keyword xml:lang="en">Computational Linguistics</
↪ ms:keyword>
        <ms:keyword xml:lang="en">Natural Language Processing</
↪ ms:keyword>
        <ms:keyword xml:lang="en">Language Resources</ms:keyword>
        <ms:keyword xml:lang="en">Research infrastructures</
↪ ms:keyword>
        <ms:keyword xml:lang="en">Language Resources</ms:keyword>
        <ms:keyword xml:lang="en">Digital Humanities</ms:keyword>
        <ms:website>https://ufal.mff.cuni.cz/</ms:website>
        <ms:headOfficeAddress>
            <ms:address xml:lang="en">OVOCNY TRH 560/5</
↪ ms:address>
            <ms:zipCode>116 36</ms:zipCode>
            <ms:city xml:lang="en">PRAHA 1</ms:city>
            <ms:country>CZ</ms:country>
        </ms:headOfficeAddress>
        <ms:divisionCategory>http://w3id.org/meta-share/meta-share/
↪ institute</ms:divisionCategory>
        <ms:isDivisionOf>
            <ms:actorType></ms:actorType>
            <ms:organizationName xml:lang="en">Charles University
↪ </ms:organizationName>
            <ms:website>https://www.cuni.cz</ms:website>
        </ms:isDivisionOf>
        </ms:Organization>
        </ms:DescribedEntity>
    </ms:MetadataRecord>

```

Example 3: SME

Evaluation and Language Resources Distribution Agency (ELDA)

Published at: <https://live.european-language-grid.eu/catalogue/#/resource/organizations/646>

```
<?xml version="1.0" encoding="UTF-8"?>
<ms:MetadataRecord xsi:schemaLocation="http://w3id.org/meta-share/meta-share/ ../
Schema/ELG-SHARE.xsd" xmlns:ms="http://w3id.org/meta-share/meta-share/" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance">
  <ms:DescribedEntity>
    <ms:Organization>
      <ms:entityType>Organization</ms:entityType>
      <ms:OrganizationIdentifier ms:OrganizationIdentifierScheme=
        "http://w3id.org/meta-share/meta-share/elg">automatically assigned by ELG - please
        don't change</ms:OrganizationIdentifier>
      <ms:organizationName xml:lang="en">Evaluation and Language
        Resources Distribution Agency</ms:organizationName>
      <ms:organizationShortName xml:lang="en">ELDA</
ms:organizationShortName>
      <ms:organizationAlternativeName xml:lang="en">EVALUATIONS AND
        LANGUAGE RESOURCES DISTRIBUTION AGENCY</ms:organizationAlternativeName>
      <ms:organizationRole>http://w3id.org/meta-share/meta-share/
        LTSupplier</ms:organizationRole>
      <ms:organizationRole>http://w3id.org/meta-share/meta-share/
        dataEvaluator</ms:organizationRole>
      <ms:organizationLegalStatus>http://w3id.org/meta-share/meta-
        share/sme</ms:organizationLegalStatus>
      <ms:organizationBio xml:lang="en">The Evaluations and
        Language Resources Distribution Agency (ELDA), was created in 1995 as the
        organizational infrastructure with the mission of providing a central clearing
        house for Language Resources (LR) of the European Language Resources Association
        (ELRA). ELDA was set up to identify, classify, collect, validate and distribute
        the language resources that are needed by the Human Language Technology (HLT)
        community. Anticipating the evolutions in the HLT field, ELDA broadened its
        activities to cover multimedia/multimodal resources as well as evaluation
        activities, distributing the language resources needed for evaluation purposes,
        and conducting/coordinating evaluation campaigns. ELDA has played a significant
        role within the major Multimedia and Multimodal production projects that resulted
        in one of the most impressive catalogues of available data sets, embracing all
        aspects of Language Technologies. ELDA was also involved in evaluation initiatives,
        in several FPs' projects involving HLT infrastructures, as well as in national
        programmes. In addition to work on data production, processing and annotation,
        validation and quality control, several of these projects also involved work on
        legal framework management for the produced resources. Moreover, ELDA has
        contributed to the development of open platforms and has joined forces with other
        European key players by bringing its assets (LR catalogue, evaluation services and
        benchmarking) to constitute Europe's backbone for Language Resources sharing and
        distribution. ELDA is also the initiator of the Language Resource and the
        Evaluation Conference (LREC), since 1998. With over 1200 participants, LREC is the
        major event on Language Resources (LRs) and Evaluation for Human Language
        Technologies (HLT).</ms:organizationBio>
      <ms:logo>https://www.european-language-grid.eu/wp-content/uploads/
        2019/03/logo__consortium-elda.svg</ms:logo>
      <ms:LTArea>
        <ms:LTClassRecommended>http://w3id.org/meta-share/
          omd-share/LanguageTechnology</ms:LTClassRecommended>
      </ms:LTArea>
```

(continues on next page)

(continued from previous page)

```

<ms:LTArea>
  <ms:LTCClassRecommended>http://w3id.org/meta-share/
  omt-d-share/Evaluation</ms:LTCClassRecommended>
</ms:LTArea>
<ms:LTArea>
  <ms:LTCClassOther>Language Resource collection,
  processing, production</ms:LTCClassOther>
</ms:LTArea>
<ms:LTArea>
  <ms:LTCClassOther>legal clearing</ms:LTCClassOther>
</ms:LTArea>
<ms:LTArea>
  <ms:LTCClassOther>HLT evaluation and dissemination</
ms:LTCClassOther>
</ms:LTArea>
<ms:keyword xml:lang="en">Language Resources and Evaluation</
ms:keyword>
<ms:keyword xml:lang="en">Legal support</ms:keyword>
<ms:keyword xml:lang="en">Data management</ms:keyword>
<ms:website>http://www.elra.info/en/</ms:website>
<ms:headOfficeAddress>
  <ms:address xml:lang="en">9 RUE DES CORDELIERES</
ms:address>
  <ms:zipCode>75 013</ms:zipCode>
  <ms:city xml:lang="en">Paris</ms:city>
  <ms:country>FR</ms:country>
</ms:headOfficeAddress>
</ms:Organization>
</ms:DescribedEntity>
</ms:MetadataRecord>

```

1.18.3 3. Register the organization at ELG

The current release of ELG offers two options for registering a catalogue item:

- the **ELG interactive editor** (see *Use the interactive editor*)
- the **upload of a metadata file** that conforms to the *ELG schema* in XML format (see *Create and upload metadata files*).

1.18.4 4. Manage and submit for publication

Through the “My items” page you can access your metadata record (see *Manage your items*) and edit it until you are satisfied. You can then *submit it for publication*, in line with the *publication lifecycle* defined for ELG metadata records.

At this stage, the metadata record can no longer be edited and is only visible to you and to us, the ELG platform administrators.

Before it is published, your submission undergoes a validation process, which is described in detail at *Validate a “metadata-only record”*.

Once approved, it will appear on the ELG catalogue and you will receive a notification email.

1.19 Contribute a project

This page describes how to contribute information for a *project* to the European Language Grid.

1.19.1 0. Before you start

- Please make sure that the information you want to contribute complies with our *terms of use*.
- Please make sure you have *registered* and been assigned the *provider role*.

1.19.2 1. Describe the project

Metadata overview

The project must be described according to the *ELG schema* and comply at least with the minimal version. The metadata elements that you need to provide for the project comprise a set of elements organized (for presentation purposes) into the following groups:

- *Minimal elements for all entities*
- *Minimal elements for projects*

Examples

Example 1

Bergamot – Browser-based Multilingual Translation

Published at: <https://live.european-language-grid.eu/catalogue/#/resource/projects/392>

```
<?xml version="1.0" encoding="UTF-8"?>
<ms:MetadataRecord xsi:schemaLocation="http://w3id.org/meta-share/meta-share/ ../
Schema/ELG-SHARE.xsd" xmlns:ms="http://w3id.org/meta-share/meta-share/" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance">
  <ms:DescribedEntity>
    <ms:Project>
      <ms:entityType>Project</ms:entityType>
      <ms:ProjectIdentifier ms:ProjectIdentifierScheme="http://w3id.
org/meta-share/meta-share/cordis">219608</ms:ProjectIdentifier>
      <ms:projectName xml:lang="en">Browser-based Multilingual
Translation</ms:projectName>
      <ms:projectShortName xml:lang="en">Bergamot</
ms:projectShortName>
      <ms:fundingType>http://w3id.org/meta-share/meta-share/euFunds
</ms:fundingType>
      <ms:funder>
        <ms:Organization>
          <ms:actorType>Organization</ms:actorType>
          <ms:organizationName xml:lang="en">European
Commission</ms:organizationName>
          <ms:website>https://ec.europa.eu/info/index_en
</ms:website>
        </ms:Organization>
      </ms:funder>
    </ms:Project>
  </ms:DescribedEntity>
</ms:MetadataRecord>
```

(continues on next page)

(continued from previous page)

```

<ms:fundingCountry>EU</ms:fundingCountry>
<ms:projectStartDate>2019-01-01</ms:projectStartDate>
<ms:projectEndDate>2021-12-31</ms:projectEndDate>
<ms:website>https://browser.mt/</ms:website>
<ms:logo>https://ufal.mff.cuni.cz/sites/default/files/styles/
↳drupal_projects_logo_style/public/bergamot_logo.png</ms:logo>
<ms:LTArea>
  <ms:LTClassRecommended>http://w3id.org/meta-share/
↳omtd-share/MachineTranslation</ms:LTClassRecommended>
</ms:LTArea>
<ms:LTArea>
  <ms:LTClassOther>Browser-based Machine Translation</
↳ms:LTClassOther>
</ms:LTArea>
<ms:domain>
  <ms:categoryLabel xml:lang="en">http://w3id.org/meta-
↳share/omtd-share/NewsMediaJournalismAndPublishing</ms:categoryLabel>
</ms:domain>
<ms:domain>
  <ms:categoryLabel xml:lang="en">General</
↳ms:categoryLabel>
</ms:domain>
<ms:keyword xml:lang="en">Machine translation</ms:keyword>
<ms:keyword xml:lang="en">translation integration</ms:keyword>
<ms:grantNumber>825303</ms:grantNumber>
<ms:projectSummary xml:lang="en">'The Bergamot project will
↳add and improve client-side machine translation in a web browser. Unlike current
↳cloud-based options, running directly on users' machines empowers citizens to
↳preserve their privacy and increases the uptake of language technologies in Europe
↳in various sectors that require confidentiality. Free software integrated with an
↳open-source web browser, such as Mozilla Firefox, will enable bottom-up adoption by
↳non-experts, resulting in cost savings for private and public sector users who
↳would otherwise procure translation or operate monolingually. To understand and
↳support non-expert users, our user experience work package researches their needs
↳and creates the user interface. Rather than simply translating text, this
↳interface will expose improved quality estimates, addressing the rising public
↳debate on algorithmic trust. Building on quality estimation research, we will
↳enable users to confidently generate text in a language they do not speak, enabling
↳cross-lingual online form filling. To improve quality overall, dynamic domain
↳adaptation research addresses the peculiar writing style of a website or user by
↳adapting translation on the fly using local information too private to upload to
↳the cloud. These applications require adaptation and inference to run on desktop
↳hardware with compact model downloads, which we address with neural network
↳efficiency research. Our combined research on user experience, domain adaptation,
↳quality estimation, outbound translation, and efficiency support a broad browser-
↳based innovation plan.'</ms:projectSummary>
<ms:cost>
  <ms:amount>2999096.25</ms:amount>
  <ms:currency>http://w3id.org/meta-share/meta-share/
↳euro</ms:currency>
</ms:cost>
<ms:ecMaxContribution>
  <ms:amount>2999096.25</ms:amount>
  <ms:currency>http://w3id.org/meta-share/meta-share/
↳euro</ms:currency>
</ms:ecMaxContribution>
<ms:fundingSchemeCategory>RIA</ms:fundingSchemeCategory>

```

(continues on next page)

(continued from previous page)

```

        <ms:status>SIGNED</ms:status>
        <ms:relatedCall>H2020-ICT-2018-2</ms:relatedCall>
        <ms:relatedProgramme>H2020</ms:relatedProgramme>
        <ms:relatedSubprogramme>ICT-29-2018</ms:relatedSubprogramme>
        <ms:coordinator>
            <ms:actorType>Organization</ms:actorType>
            <ms:organizationName xml:lang="en">THE UNIVERSITY OF
↪EDINBURGH</ms:organizationName>
            <ms:website>https://www.ed.ac.uk/</ms:website>
        </ms:coordinator>
        <ms:participatingOrganization>
            <ms:actorType>Organization</ms:actorType>
            <ms:organizationName xml:lang="en">TARTU ULIKOOL</
↪ms:organizationName>
        </ms:participatingOrganization>
        <ms:participatingOrganization>
            <ms:actorType>Organization</ms:actorType>
            <ms:organizationName xml:lang="en">MZ DENMARK APS</
↪ms:organizationName>
        </ms:participatingOrganization>
        <ms:participatingOrganization>
            <ms:actorType>Organization</ms:actorType>
            <ms:organizationName xml:lang="en">THE UNIVERSITY OF
↪SHEFFIELD</ms:organizationName>
        </ms:participatingOrganization>
        <ms:participatingOrganization>
            <ms:actorType>Organization</ms:actorType>
            <ms:organizationName xml:lang="en">UNIVERZITA KARLOVA
↪</ms:organizationName>
            <ms:website>https://www.cuni.cz/</ms:website>
        </ms:participatingOrganization>
    </ms:Project>
</ms:DescribedEntity>
</ms:MetadataRecord>

```

Example 2

European Language Grid

Published at: <https://live.european-language-grid.eu/catalogue/#/resource/projects/395>

```

<?xml version="1.0" encoding="UTF-8"?>
<ms:MetadataRecord xsi:schemaLocation="http://w3id.org/meta-share/meta-share/ ../
↪Schema/ELG-SHARE.xsd" xmlns:ms="http://w3id.org/meta-share/meta-share/" xmlns:xsi=
↪"http://www.w3.org/2001/XMLSchema-instance">
    <ms:DescribedEntity>
        <ms:Project>
            <ms:entityType>Project</ms:entityType>
            <ms:ProjectIdentifier ms:ProjectIdentifierScheme="http://w3id.
↪org/meta-share/meta-share/cordis">219378</ms:ProjectIdentifier>
            <ms:projectName xml:lang="en">European Language Grid</
↪ms:projectName>
            <ms:projectShortName xml:lang="en">ELG</ms:projectShortName>
            <ms:fundingType>http://w3id.org/meta-share/meta-share/euFunds
↪</ms:fundingType>

```

(continues on next page)

(continued from previous page)

```

    <ms:funder>
      <ms:Organization>
        <ms:actorType>Organization</ms:actorType>
        <ms:organizationName xml:lang="en">European_
↪Commission</ms:organizationName>
        <ms:website>https://ec.europa.eu/info/index_en
↪</ms:website>
      </ms:Organization>
    </ms:funder>
    <ms:fundingCountry>EU</ms:fundingCountry>
    <ms:projectStartDate>2019-01-01</ms:projectStartDate>
    <ms:projectEndDate>2021-12-31</ms:projectEndDate>
    <ms:website>https://www.european-language-grid.eu/</
↪ms:website>
    <ms:logo>https://www.european-language-grid.eu/wp-content/
↪themes/elg_theme/fab/image/logo/rgb_elg__logo--colour.svg</ms:logo>
    <ms:LTArea>
      <ms:LTClassRecommended>http://w3id.org/meta-share/
↪omtd-share/LanguageTechnology</ms:LTClassRecommended>
    </ms:LTArea>
    <ms:keyword xml:lang="en">Language technology services</
↪ms:keyword>
    <ms:keyword xml:lang="en">Multilingualism</ms:keyword>
    <ms:keyword xml:lang="en">Less-resourced languages</
↪ms:keyword>
    <ms:grantNumber>825627</ms:grantNumber>
    <ms:projectSummary xml:lang="en">With 24 official EU and many_
↪more additional languages, multilingualism in Europe and an inclusive Digital_
↪Single Market can only be enabled through Language Technologies (LTs). European LT_
↪business is dominated by thousands of SMEs and a few large players. Many are world-
↪class, with technologies that outperform the global players. However, European LT_
↪business is also fragmented by nation states, languages, verticals and sectors._
↪Likewise, while much of European LT research is world-class, with results_
↪transferred into industry and commercial products, its full impact is held back by_
↪fragmentation. The key issue and challenge is the fragmentation of the European LT_
↪landscape. The European Language Grid (ELG) project will address this fragmentation_
↪by establishing the ELG as the primary platform for LT in Europe. The ELG will be a_
↪scalable cloud platform, providing, in an easy-to-integrate way, access to hundreds_
↪of commercial and non-commercial Language Technologies for all European languages,_
↪including running tools and services as well as data sets and resources. It will_
↪enable the commercial and non-commercial European LT community to deposit and_
↪upload their technologies and data sets into the ELG, to deploy them through the_
↪grid, and to connect with other resources. The ELG will boost the Multilingual_
↪Digital Single Market towards a thriving European LT community, creating new jobs_
↪and opportunities. Through open calls, up to 20 pilot projects will be financially_
↪supported to demonstrate the usefulness of the ELG. The proposal is rooted in the_
↪experience of a consortium with partners involved in all relevant initiatives._
↪Based on these, 30 national competence centres and the European LT Board will be_
↪set up for European coordination. The ELG will foster language technologies for_
↪Europe built in Europe, tailored to our languages and cultures and to our societal_
↪and economical demands, benefitting the European citizen, society, innovation and_
↪industry.</ms:projectSummary>
    <ms:cost>
      <ms:amount>7460206.25</ms:amount>
      <ms:currency>http://w3id.org/meta-share/meta-share/
↪euro</ms:currency>
    </ms:cost>

```

(continues on next page)

(continued from previous page)

```

        <ms:ecMaxContribution>
          <ms:amount>6999631.25</ms:amount>
          <ms:currency>http://w3id.org/meta-share/meta-share/
↪euro</ms:currency>
        </ms:ecMaxContribution>
        <ms:fundingSchemeCategory>IA</ms:fundingSchemeCategory>
        <ms:status>SIGNED</ms:status>
        <ms:relatedCall>H2020-ICT-2018-2</ms:relatedCall>
        <ms:relatedProgramme>H2020</ms:relatedProgramme>
        <ms:relatedSubprogramme>ICT-29-2018</ms:relatedSubprogramme>
        <ms:coordinator>
          <ms:actorType>Organization</ms:actorType>
          <ms:organizationName xml:lang="en">DEUTSCHES
↪FORSCHUNGSZENTRUM FUR KUNSTLICHE INTELLIGENZ GMBH</ms:organizationName>
          <ms:website>https://www.dfki.de/</ms:website>
        </ms:coordinator>
        <ms:participatingOrganization>
          <ms:actorType>Organization</ms:actorType>
          <ms:organizationName xml:lang="en">SAIL LABS
↪TECHNOLOGY GMBH</ms:organizationName>
          <ms:website>https://www.sail-labs.com/</ms:website>
        </ms:participatingOrganization>
        <ms:participatingOrganization>
          <ms:actorType>Organization</ms:actorType>
          <ms:organizationName xml:lang="en">THE UNIVERSITY OF
↪SHEFFIELD</ms:organizationName>
          <ms:website>https://www.dfki.de/</ms:website>
        </ms:participatingOrganization>
        <ms:participatingOrganization>
          <ms:actorType>Organization</ms:actorType>
          <ms:organizationName xml:lang="en">ATHINA-EREVNITIKO
↪KENTRO KAINOTOMIAS STIS TECHNOLOGIES TIS PLIROFORIAS, TON EPIKOINONION KAI TIS
↪GNOSIS</ms:organizationName>
          <ms:website>https://www.athena-innovation.gr/</ms:website>
↪ms:website>
        </ms:participatingOrganization>
        <ms:participatingOrganization>
          <ms:actorType>Organization</ms:actorType>
          <ms:organizationName xml:lang="en">EVALUATIONS AND
↪LANGUAGE RESOURCES DISTRIBUTION AGENCY</ms:organizationName>
          <ms:website>http://www.elda.org/</ms:website>
        </ms:participatingOrganization>
        <ms:participatingOrganization>
          <ms:actorType>Organization</ms:actorType>
          <ms:organizationName xml:lang="en">TILDE SIA</ms:organizationName>
↪ms:organizationName>
          <ms:website>https://www.tilde.eu/</ms:website>
        </ms:participatingOrganization>
        <ms:participatingOrganization>
          <ms:actorType>Organization</ms:actorType>
          <ms:organizationName xml:lang="en">UNIVERZITA KARLOVA
↪</ms:organizationName>
          <ms:website>https://www.cuni.cz/</ms:website>
        </ms:participatingOrganization>
        <ms:participatingOrganization>
          <ms:actorType>Organization</ms:actorType>
          <ms:organizationName xml:lang="en">THE UNIVERSITY OF
↪EDINBURGH</ms:organizationName>

```

(continues on next page)

(continued from previous page)

```

        <ms:website>https://www.ed.ac.uk/</ms:website>
    </ms:participatingOrganization>
    <ms:participatingOrganization>
        <ms:actorType>Organization</ms:actorType>
        <ms:organizationName xml:lang="en">EXPERT SYSTEM_
↪IBERIA SL</ms:organizationName>
        <ms:website>http://www.expertsystem.com/</ms:website>
    </ms:participatingOrganization>
</ms:Project>
</ms:DescribedEntity>
</ms:MetadataRecord>

```

1.19.3 2. Register the project at ELG

The current release of ELG offers two options for registering a catalogue item:

- the ELG **interactive editor** (see *Use the interactive editor*)
- the **upload of a metadata file** that conforms to the *ELG schema* in XML format (see *Create and upload metadata files*).

1.19.4 3. Manage and submit for publication

Through the “My items” page you can access your metadata record (see *Manage your items*) and edit it until you are satisfied. You can then *submit it for publication*, in line with the *publication lifecycle* defined for ELG metadata records.

At this stage, the metadata record can no longer be edited and is only visible to you and to us, the ELG platform administrators.

Before it is published, your submission undergoes a validation process, which is described in detail at *Validate a “metadata-only record”*.

Once approved, it will appear on the ELG catalogue and you will receive a notification email.

1.20 Claim an item

A subset of the published items in the ELG catalogue have been created and imported in the catalogue by the ELG team. These are organizations (companies, research and academic organizations) active in the Language Technology area at large.

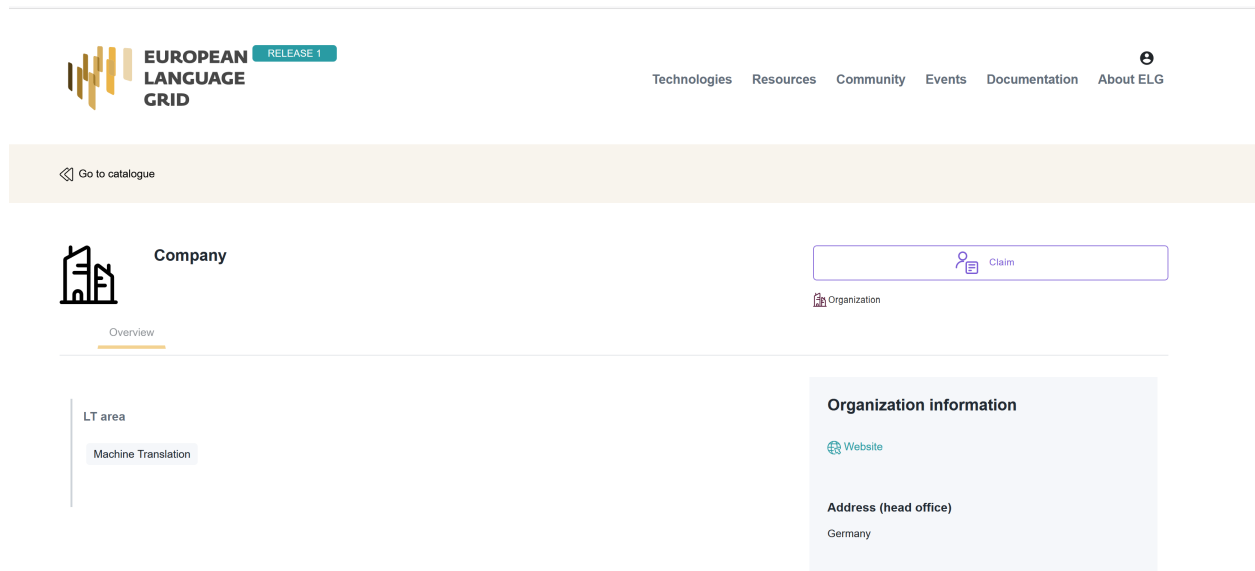
We have created these pages with minimal information and we welcome those interested to enrich these pages and add their LT services and products to the LT catalogue.

You can **claim** your organization’s page by clicking on the “claim” button that appears on the top right corner of the view page, as shown below:

Please, keep in mind that to **claim** an item, you must be logged in; if you don’t have an account, you can follow the instructions to *register*. After signing in, go to the organization page to continue with the claim procedure.

When we receive your claim, we will check that this is a valid claim; for this reason, we recommend the use of a professional email address for the registration.

When we have processed your claim, you will receive an email. If your claim has been approved, please sign in and go to “My items” page to edit your record (see *Manage your items*).



1.21 Contribute via an external repository

The European Language Grid supports metadata harvesting based on the protocol developed by the Open Archives Initiative [OAI-PMH] for metadata records that conform to the [ELG schema](#).

Currently, the following repositories are harvested:

- [ELRC-SHARE](#)
- [LINDAT/CLARIAH](#)

If you wish to contribute resources through harvesting, please contact us at contact@european-language-grid.eu.

1.22 Create catalogue items

This section describes how you can use the ELG platform **to create** new items which will be published at the ELG catalogue.

All these actions are available through the provider's grid. To access it, you must [sign in](#) with an account that has the “provider” role. If you don't have such an account, see how you can [register](#) and become a [provider](#).

The current release of ELG offers two options for registering a catalogue item:

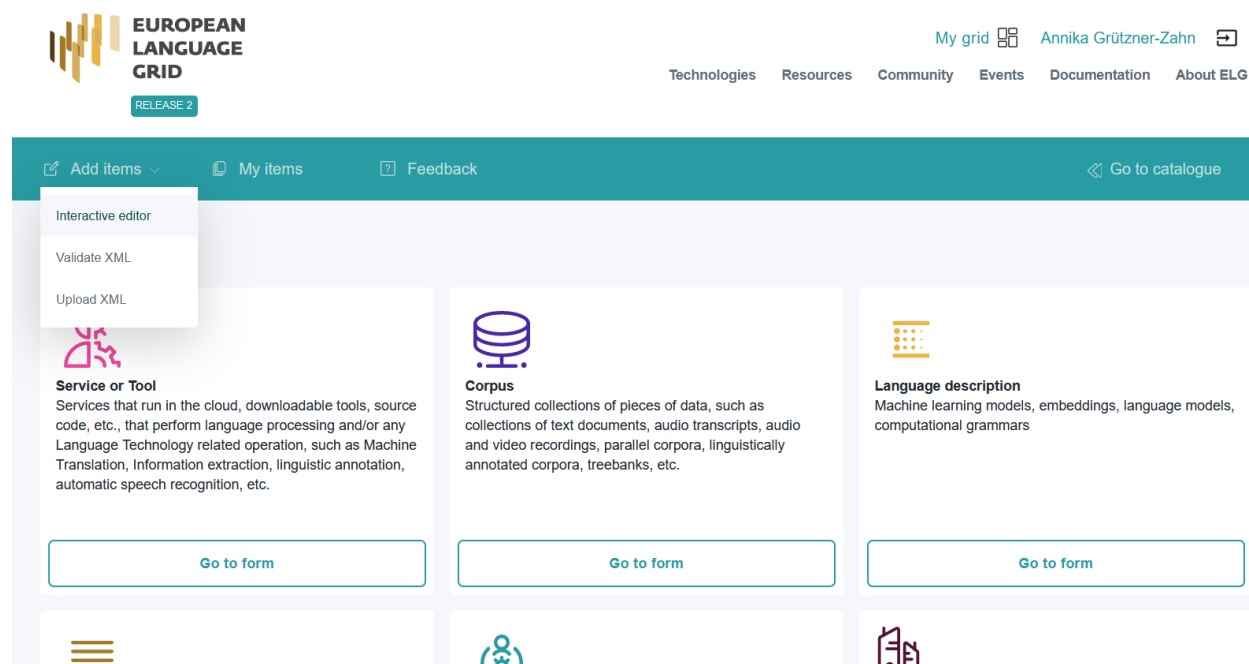
- the **ELG interactive editor** (see [Use the interactive editor](#))
- the **upload of a metadata file** that conforms to the [ELG schema](#) (see [Create and upload metadata files](#)).

Tip: You can use the [Copy records](#) functionality to create similar metadata records to the ones you have already created.

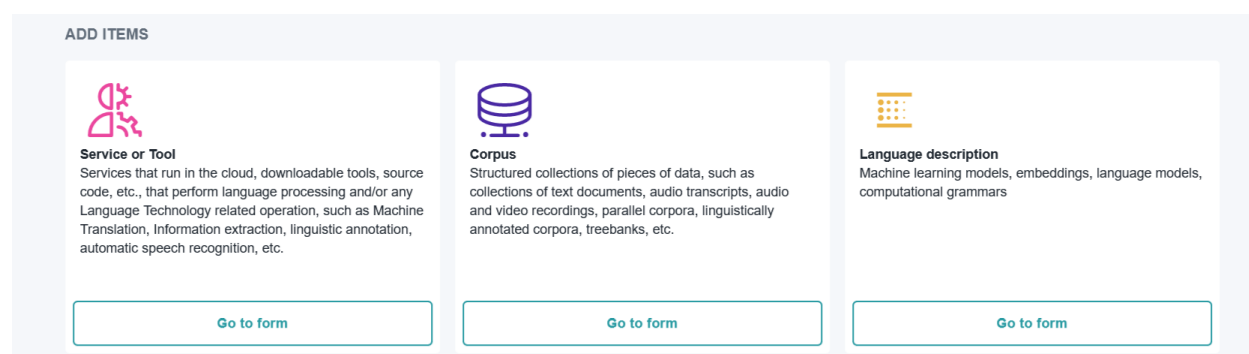
1.22.1 Use the interactive editor

Create item

To access the editor form, sign in to ELG, click on **My grid** and select the “Add items” option. From the dropdown list choose the “Interactive editor” as shown below:




On the item type selection page, select the form for the type of item you wish to create, e.g. “Corpus” in the screenshot below.



Follow the instructions that appear in order to create a new item. The editor form includes all the elements of the minimal version of the ELG schema (mandatory and recommended). These are organized into Sections and Tabs. You can fill in the elements in any order you wish.

Note: Resources marked as “work in progress” are ELG compatible services or resources that will be hosted at ELG and are in the process of being created by the providers. For such cases, some of the information is not yet known (e.g. size of the resource, licence). If you tick the “work in progress” box, when the item is published on the ELG catalogue, the “Download” tab will not be visible. When the resource is ready, please contact us and we will return the record to the editing status, so that you can proceed with the final submission.


**EUROPEAN
LANGUAGE
GRID**
RELEASE 2

[My grid](#)
[Annika Grütner-Zahn](#)

[Technologies](#)
[Resources](#)
[Community](#)
[Events](#)
[Documentation](#)
[About ELG](#)

[Add items](#)
[My items](#)
[Feedback](#)
[Administration](#)
[Go to catalogue](#)

Info

- At any point you are able to save your progress as draft and continue editing at a later time
- Once you submit your record you will not be able to save it as draft any more, but you will still be able to make changes and submit them.
- Visit all tabs in order to fill in as much information as possible for better visibility of your record.
- If the metadata record is for a resource that you plan to deliver later, please check the "work in progress" box.

LANGUAGE RESOURCE/TECHNOLOGY
CORPUS
PART
DISTRIBUTION
☐ Work in progress
[Save draft](#)
[Save](#)

IDENTITY

LRT name *

corpus test

The official name or title of the language resource/technology

language

English

select language

LRT identifier

A string used to uniquely identify the language resource/technology

[Fill in](#)

LRT short name

English

An abbreviation, acronym, etc. used for the language resource/technology

select language

Description *

English

General information on the language resource/technology

select language

CATEGORIES

CONTACT

DOCUMENTATION

RELATED LRTS

To save the record, you can click on “Save draft” or “Submit”, depending on whether you have filled in all the required elements (see [Publication lifecycle](#)). In both cases, you can come back later and continue editing it.

When you are satisfied with the description, you can proceed to submit the record for *publication*. For this step, you **MUST** have filled in all the required elements; to find out which ones these are, see the section relevant to the type of item at *Minimal version*.

Manage your item through view page

Each time you click on “Submit”, you can view the metadata record as shown on the ELG catalogue:

The screenshot shows the ELG catalogue interface. At the top left is the ELG logo with 'RELEASE 2' below it. To the right is a navigation bar with links: Technologies, Resources, Community, Events, Documentation, and About ELG. Further right is a user profile section for 'Annika Grützner-Zahn' with a 'My grid' icon. A 'Go to catalogue' link is in the top right corner. Below the navigation bar is a 'STATUS' section with a text box explaining the workflow: 'You can continue editing your metadata record while the status is draft or syntactically valid; when you are satisfied with it, you can submit it for publication; the ELG team will check for conformance with technical requirements and publish it or, if required, contact you for further information'. Below this is a progress bar with four stages: 'draft' (selected), 'syntactically valid', 'submitted', and 'published'. The main content area shows the record details for 'service tool 3' (Version: 1.0.0 (automatically assigned)). There are two tabs: 'Overview' (active) and 'Download/Run'. Below the tabs, it says 'provides a service'. On the right side, there is a 'ToolService' icon and a list of actions: 'Edit Metadata' (with a pencil icon), 'Copy record' (with a plus icon), 'Submit for publication' (with a checkmark icon), and 'Delete Metadata' (with a trash icon). Each action is in a button with a corresponding icon.

From this page, you can select to go back to editing mode, submit it for publication or delete the metadata, depending on the status of the record (see [Manage your items](#) for more information).

1.22.2 Create and upload metadata files

Create a metadata file

If you wish to create metadata files compliant with the *ELG schema*, you will find at the ELG GitLab [templates](#) (that you can use to create new metadata records) and [examples](#). For this release, the metadata files **MUST** be in XML format.

Note: The metadata files that you upload to ELG **MUST** be “syntactically valid”; i.e, they **MUST** contain all required elements of the *Minimal version* of the ELG schema.

In contrast, the metadata records created via the editor can also be saved as “draft” (only a “name” is required for this action); **only when all the required elements have been filled in**, can they be submitted and considered “syntactically valid”.

Validate metadata file(s) against the ELG XSD

You can validate the metadata file(s) you create against the ELG schema XSD, using the ELG validator, which is accessible in two modes:

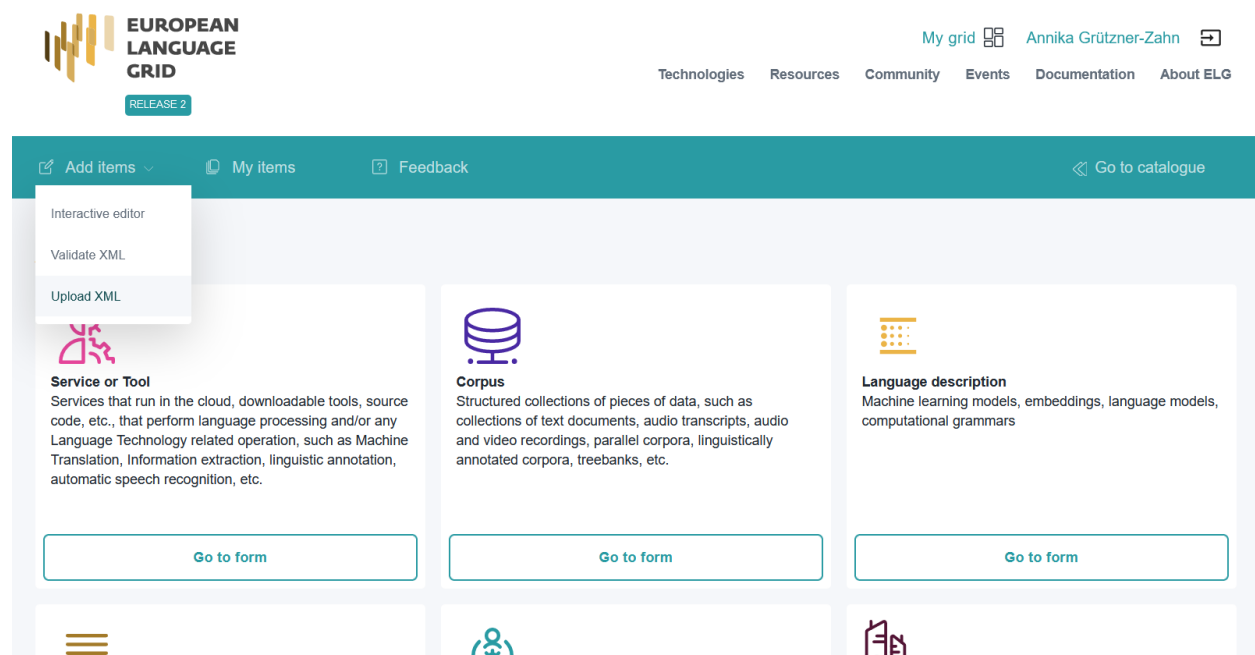
- publicly available at <https://live.european-language-grid.eu/catalogue/#/validate-xml>, for interested users without signing in at ELG;
- through the grid, by clicking on **My grid** and selecting the “Add items” option. From the dropdown list, select “Validate XML” as below:

FIXME add screenshot

The ELG validator can be used for validating single XML files or zipped archives.

Upload the metadata file(s)

Click on **My grid** and select the “Add items” option. From the dropdown list, select “Upload XML” as below:






Upload a single file

Now upload the file you created.






Note: Resources marked as “work in progress” are ELG compatible services or resources that will be hosted at ELG and are in the process of being created by the providers. For such cases, some of the information is not yet known (e.g. size of the resource, licence). If you tick the “work in progress” box, when the item is published on the ELG catalogue, the “Download” tab will not be visible. When the resource is ready, please contact us and we will return the record to the editing status, so that you can proceed with the final submission.

If there are any errors in your XML file, these will be shown to you. Fix them and try the upload again.

**EUROPEAN
LANGUAGE
GRID**
RELEASE 2

My grid  Annika Grützner-Zahn 

TechnologiesResourcesCommunityEventsDocumentationAbout ELG

Add items My items Feedback Administration Go to catalogue 

VALIDATE XML FILES**UPLOAD SINGLE ITEM**UPLOAD MULTIPLE ITEMS

You can upload one **xml** file each time.

It is highly recommended that you **validate** your XML file against the ELG schema before you proceed.


☐ Work in progress



If the metadata record is for a resource that you plan to deliver later, please check the "work in progress" box.

☐ ELG-compatible service





If the metadata record is for a service to be integrated in ELG (https://european-language-grid.readthedocs.io/en/release1.1.1/all/3_Contributing/Service.html), please check the box "ELG-compatible service".

Drag & Drop your file or [Browse](#)

**EUROPEAN
LANGUAGE
GRID**
RELEASE 2

My grid  Annika Grützner-Zahn 

TechnologiesResourcesCommunityEventsDocumentationAbout ELG

Add items My items Feedback Go to catalogue 

VALIDATE XML FILES**UPLOAD SINGLE ITEM**UPLOAD MULTIPLE ITEMS

You can upload one **xml** file each time.

It is highly recommended that you **validate** your XML file against the ELG schema before you proceed.

☐ Work in progress

If the metadata record is for a resource that you plan to deliver later, please check the "work in progress" box.

☒ ELG-compatible service

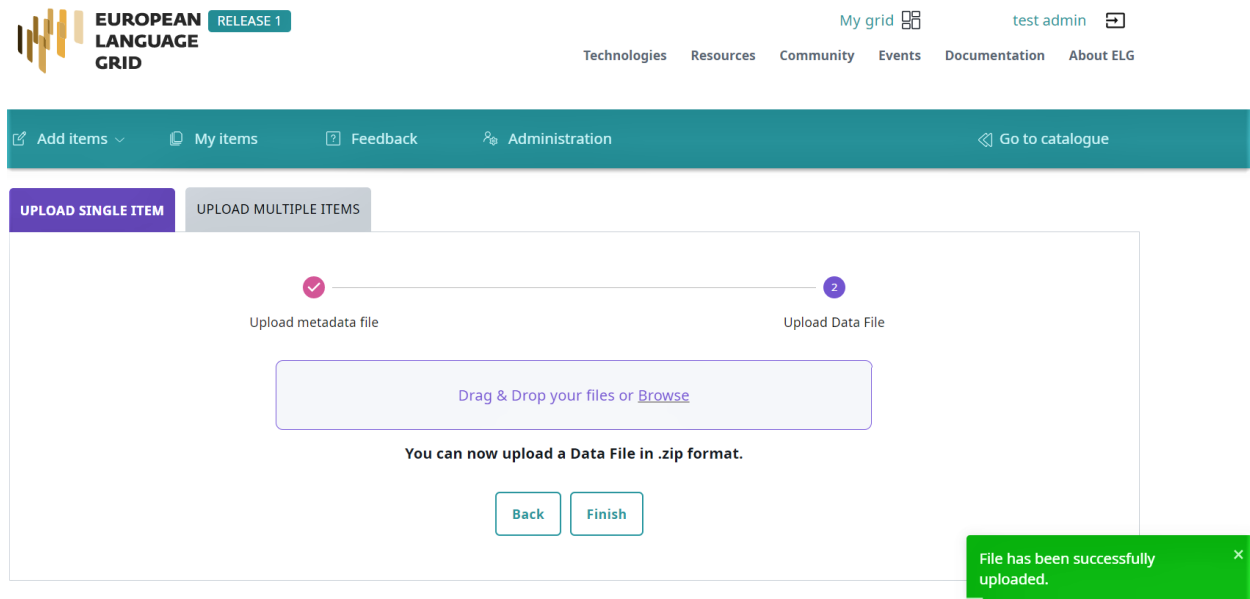
If the metadata record is for a service to be integrated in ELG (https://european-language-grid.readthedocs.io/en/release1.1.1/all/3_Contributing/Service.html), please check the box "ELG-compatible service".

Error during upload
Tap to retry

Error

```
{
  "ms:DescribedEntity": {
    "ms:organizationName": [
      "This field may not be null."
    ]
  }
}
```

Eventually, a success message will be shown to you and the metadata will be imported into the database. You will then be prompted to upload the content files for the record you submitted, if you want to.



When you have finished, you can submit more metadata files, or proceed to the “My items” page to view, edit and submit the record(s) for publication (see [Manage your items](#)).

Upload batch files

By following the same process, you can upload multiple metadata files zipped together in a file (with extension .zip).

You will receive an email when the upload has finished with a report on the files that were successfully uploaded and the errors, if any, of failed records. You can then go to the “My items” page to view, edit and submit for publication the uploaded metadata records (see [Manage your items](#)).


1.23 Manage your items



This section guides you to **manage** your items, i.e. perform the following actions:

- edit or update a metadata record,
- submit it for publication,
- export your metadata record(s),
- delete a *metadata record*,
- upload content files to a metadata record, and
- replace the *content files* you have uploaded.






All these actions are available through the provider’s grid (**My grid**). If you are not a provider and wish to become one, see the instructions provided [here](#). There are currently two access modes to manage your items:

- a. the **My items** page, from where you can select one or multiple items, and
- b. the **view page** of each metadata record.


**EUROPEAN
LANGUAGE
GRID**RELEASE 1


My grid  test admin 

TechnologiesResourcesCommunityEventsDocumentationAbout ELG

 Add items  My items  Feedback  Administration  Go to catalogue

UPLOAD SINGLE ITEM **UPLOAD MULTIPLE ITEMS**

 Upload metadata file


 Upload Data File



All steps completed

New upload





Please go to my items to edit the metadata record and submit it for publication

Go to my items

**EUROPEAN
LANGUAGE
GRID**RELEASE 2

My grid  Annika Grützner-Zahn 

TechnologiesResourcesCommunityEventsDocumentationAbout ELG

 Add items  My items  Feedback  Go to catalogue

VALIDATE XML FILES **UPLOAD SINGLE ITEM** **UPLOAD MULTIPLE ITEMS**


You can upload one **zip** file each time (with multiple XML files).

☐ Work in progress


If the metadata record is for a resource that you plan to deliver later, please check the "work in progress" box.

☐ ELG-compatible service

If the metadata records are for services to be integrated in ELG (https://european-language-grid.readthedocs.io/en/release1.1.1/all/3_Contributing/Service.html), please check the box "ELG-compatible service".

 Organization test upload.zip

885 bytes



Note: Please, keep in mind, especially when you wish to perform an action on multiple items, that actions are in direct relation to the items status. When you select multiple items with different statuses, only actions that are valid for all of them will be displayed.

The following table shows the actions which can be performed on an item in the various phases of the publication lifecycle.

Status	edit	submit for publication	delete	upload content	replace content
draft	y	n	y	y	y
syntactically valid	y	y	y	y	y
submitted for publication	n	n/a	n	n	n
published	n	n/a	n	n	n

1.23.1 a. My items

The “My items” page looks like this:

The screenshot displays the 'My items' page of the European Language Grid. At the top, there is a navigation bar with the logo and 'RELEASE 2' badge. The main header includes links for 'My grid', 'Annika Grütznér-Zahn', and various categories like 'Technologies', 'Resources', 'Community', 'Events', 'Documentation', and 'About ELG'. Below this is a teal navigation bar with 'Add items', 'My items', 'Feedback', and 'Go to catalogue'.

The 'MY ITEMS' section features a search bar and a 'Search' button. On the left, there are filter panels for 'Items' (Corpus, Organization, Tool/Service) and 'Status' (syntactically valid, draft). The main area shows 4 search results in a table:

Resource name	Actions	Status
service tool 2 1.0.0 (automatically assigned) <input type="checkbox"/> Created: 08 April 2021 Updated: 12 April 2021 Tool/Service	Actions	syntactically valid
Organization 2 <input type="checkbox"/> Created: 08 April 2021 Updated: 08 April 2021 Organization	Actions	syntactically valid
Organization <input type="checkbox"/> Created: 08 April 2021 Updated: 08 April 2021 Organization	Actions	syntactically valid
test corpus 2 1.0.0 (automatically assigned) <input type="checkbox"/> Created: 08 April 2021 Updated: 08 April 2021 Corpus	Actions	draft

Here, you can perform actions on a selected item. To do this, simply click on the icon on the right of its name and you

will be presented with a choice of options.

The screenshot displays the 'My items' interface of the European Language Grid. On the left, a sidebar lists item types: Corpus (1), Organization (1), and Tool/Service (1). Below this, a status filter shows 'syntactically valid' (3) and 'draft' (1). The main area lists four items: 'service tool 2' (1.0.0, automatically assigned), 'Organization 2', 'Organization', and 'test corpus 2' (1.0.0, automatically assigned). Each item has a checkbox and a 'syntactically valid' or 'draft' status button. An 'Actions' dropdown menu is open for the 'test corpus 2' item, showing options: 'Edit Metadata', 'Copy record', 'Submit for publication', and 'Delete Metadata'. The 'Edit Metadata' option is highlighted.

You can also select multiple items by clicking on the box next to their name. Depending on their status, you will be presented with a dropdown list from the “Action” box on top, as below. Whichever action you choose will be applied to all the selected items.

1.23.2 b. View item


From the “My items” page you can be directed to the view page for an item. To do so, click on the actions and select “Edit”. On this page, you can again see the available actions from the action box.



Regardless of the mode you choose, the actions which you can perform on one or multiple items are described next.

1.24 Edit

You can change the metadata (i.e. **edit** or **update** them) of a catalogue item you have either created or been assigned to curate, if the status of the item allows it.

All updates are made via the *Use the interactive editor*. To view the items for which you are responsible, you can visit the “My items” page through **My grid** and *select* the one you wish to edit.


**EUROPEAN
LANGUAGE
GRID**
RELEASE 2

[My grid](#)  [Annika Grützner-Zahn](#) 

[Technologies](#) [Resources](#) [Community](#) [Events](#) [Documentation](#) [About ELG](#)

[Add items](#) [My items](#) [Feedback](#) [Go to catalogue](#)

[Clear all filters](#)

Items
+ Corpus (1)
+ Organization (1)
+ Tool/Service (1)
Status
+ syntactically valid (3)
+ draft (1)

Action

Submit for publication

Delete

4 search results

Actions

Status

service tool 2

1.0.0 (automatically assigned)

☒

Created: 08 April 2021

Updated: 12 April 2021

Tool/Service

Actions

syntactically valid

Organization 2


Created: 08 April 2021



Updated: 08 April 2021

Organization

Actions

syntactically valid


**EUROPEAN
LANGUAGE
GRID**
RELEASE 2

[My grid](#)  [Annika Grützner-Zahn](#) 

[Technologies](#) [Resources](#) [Community](#) [Events](#) [Documentation](#) [About ELG](#)

[Go to catalogue](#)

STATUS


You can continue editing your metadata record while the status is draft or syntactically valid; when you are satisfied with it, you can submit it for publication; the ELG team will check for conformance with technical requirements and publish it or, if required, contact you for further information

draft

syntactically valid

submitted

published



service tool 3
Version: 1.0.0 (automatically assigned)

[Overview](#) [Download/Run](#)

provides a service

Edit Metadata

Copy record

Submit for publication

Delete Metadata

ToolService

1.24. Edit

85

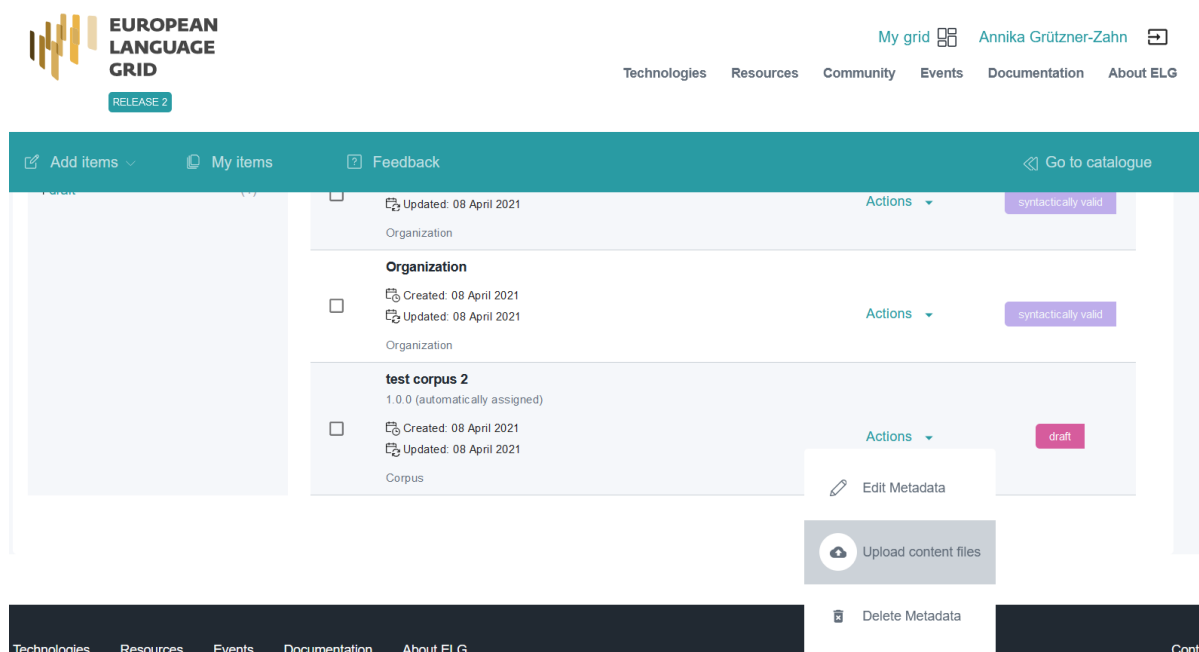
1.25 Upload and Replace content files

Note: For this release, upload of content files is supported only for **Corpora**, **Lexical/Conceptual resources** and **Language descriptions**. Please, keep in mind, that the content files must be compressed in a **zip** format.

You can upload content files:

- immediately after uploading an *XML metadata file*, or
- by selecting the “upload dataset” action via the *myItems* page once the metadata record has been created.

From this page, you can select an item and upload content files.



If you need to replace the content files you have uploaded, you can do it following the same procedure.

Alternatively, you can use the the view item page.

Note: You cannot delete the content files you have uploaded but by replacing them, they are no longer accessible. If you wish to have them removed, please contact us at contact@european-language-grid.eu.

1.26 Submit for publication

Once you are satisfied with the description of an item, you can submit it for publication, in line with the *publication lifecycle* defined for ELG metadata records.

You can do this through the **My items** page for a single or multiple items, or the landing page of the metadata record.

The screenshot shows the European Language Grid (ELG) interface. At the top, the logo and 'RELEASE 2' are visible. The navigation bar includes 'Technologies', 'Resources', 'Community', 'Events', 'Documentation', and 'About ELG'. A user profile 'Annika Grützner-Zahn' is logged in. Below the navigation bar, there are links for 'Add items', 'My items', 'Feedback', and 'Go to catalogue'. A modal dialog is open, titled 'Please select a .zip file in order to upload a dataset for the test corpus 2 record'. The dialog contains a button 'Drag & Drop your files or Browse' and two buttons at the bottom: 'cancel' and 'upload dataset'.

The screenshot shows the European Language Grid (ELG) interface with 'RELEASE 1' and 'Test Provider' logged in. The navigation bar includes 'Technologies', 'Resources', 'Community', 'Events', 'Documentation', and 'About ELG'. Below the navigation bar, there are links for 'Add items', 'My items', 'Feedback', 'Administration', and 'Go to catalogue'. A sidebar on the left shows filters for 'Items' and 'Status'. The main content area displays a table of resources with columns 'Resource name', 'Actions', and 'status'.

Resource name	Actions	status
downloadable software TEST Created: 15 February 2021 Updated: 15 February 2021 Tool/Service		syntactically valid
testCorpus Created: 15 February 2021 Updated: 15 February 2021 Corpus		draft
ogr123 Created: 14 February 2021 Updated: 14 February 2021 Organization		draft
test project Created: 10 February 2021 Updated: 10 February 2021 Project		draft
test corpus upload Created: 09 February 2021 Updated: 09 February 2021 Corpus		syntactically valid
test corpus Created: 31 January 2021 Updated: 09 February 2021 Corpus		

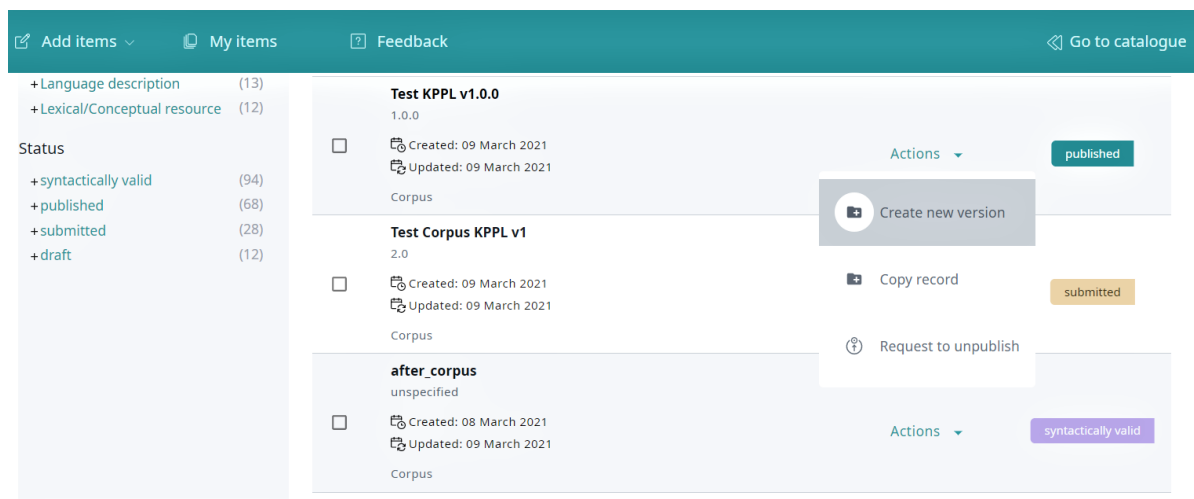
A modal dialog is open over the 'test corpus' row, showing options: 'Edit', 'Submit', and 'replace dataset'.

1.27 Create new versions

When you create a new version for an LRT already registered in the ELG catalogue, you can submit it through the “Create new version” action. This will create a copy of the existing record with all the metadata that you have used for the old version, allow you to edit the information required for the new version and automatically create a relation with the previous version.

Note: This action is available only for published metadata records.

From the **My items** page select an item and then from the list of actions choose **Create new version**. Fill in the version number in the field that appears and, optionally, a version date. For organizations and projects, you will be prompted to add a new name.



When the new record is created, you will be redirected to its view page; you can then proceed to edit it and submit for publication, following the publication lifecycle for all LRTs. If you have uploaded content files for the previous version, these will not be copied to the new version, so that you can upload the new files.

When you return to the **My items** page, you will see both versions of the metadata record.

Note: Versions of an LRT appear all on the catalogue, with the newest version first.

On the view page of each version, links to previous/new versions are clearly marked.

1.28 Copy records

To describe similar LRTs, organizations or projects, you can use the **Copy Record** feature. This allows you to use an existing record as the matrix for generating them.

Note: This action is not available for draft records.

From the **My items** page select **Copy Record** from the action button next to the resource name.


Fill in the name and (optionally) version fields.

[Go to catalogue](#)

STATUS

You can continue editing your metadata record while the status is draft or syntactically valid; when you are satisfied with it, you can submit it for publication; the ELG team will check for conformance with technical requirements and publish it or, if required, contact you for further information

draft syntactically valid submitted published

 **Test KPPL v1.1.0**
Version: 1.1.0

Overview Download Related LRTs

Created for testing

Keyword	Domain	Corpus subclass
corpus	FINANCE	raw corpus

Corpus

Edit Metadata

Upload content files

Copy record

Submit for publication

Delete Metadata

Previous

Test KPPL

Additional information

[Add items](#) [My items](#) [Feedback](#) [Go to catalogue](#)

MY ITEMS

Search for services, tools, datasets, organizations...

Search

202 search results

Clear all filters

Items

- +Corpus (73)
- +Tool/Service (41)
- +Organization (38)
- +Project (25)
- +Language description (13)
- +Lexical/Conceptual resource (12)

Status

- +syntactically valid (94)
- +published (68)
- +submitted (28)
- +draft (12)

Resource name	Actions	Status
Test KPPL v1.1.0 1.1.0 Created: 09 March 2021 Updated: 09 March 2021 Corpus	<input type="checkbox"/> Actions	published
Test KPPL v1.0.0 1.0.0 Created: 09 March 2021 Updated: 09 March 2021 Corpus	<input type="checkbox"/> Actions	published
Test Corpus KPPL v1 2.0 Created: 09 March 2021 Updated: 09 March 2021	<input type="checkbox"/> Actions	submitted

Search

Language resources & technologies

Service functions

Languages

Licences

Conditions of use for data

Related entities

81 results for Test KPPL v1.1.0

Test KPPL v1.1.0
1.1.0

0 views

Created for testing
Keyword: corpus
Languages: English · Bulgarian
Licences: Public Domain · Creative Commons - Attribution 4.0 International (CC BY 4.0) ...

Test Project v1.1.0

0 views

Project created for testing purposes.
Keywords: Machine translation · translation integration

Test KPPL v1.0.0
1.0.0

8 views

Created for testing
Keyword: corpus
Languages: English · Bulgarian
Licence: Public Domain

test provider

My grid

[Go to catalogue](#)

Test KPPL v1.1.0
Version: 1.1.0

Corpus

Actions

OverviewDownloadRelated LRTs

Created for testing

Keyword
corpus

Domain
FINANCE

Corpus subclass
raw corpus

Corpus part

TEXT

Linguality type
bilingual

Multilinguality type
parallel

Previous versions

[Test KPPL v1.0.0 \(1.0.0\)](#)

Test KPPL v1.0.0 (1.1.0)


Additional information

[Landing page](#)

Contact



Annie Rusinova

Email








**EUROPEAN
LANGUAGE
GRID**

RELEASE 2

My grid  Annika Grütznher-Zahn 

Technologies Resources Community Events Documentation About ELG

Add items  My items  Feedback  Go to catalogue 

Clear all filters 






Items


- + Corpus (1)
- + Organization (1)
- + Tool/Service (1)

Status

- + syntactically valid (3)
- + draft (1)



4 search results

Resource name	Actions	Status
service tool 2 1.0.0 (automatically assigned) <input type="checkbox"/> Created: 08 April 2021 <input type="checkbox"/> Updated: 12 April 2021 Tool/Service	Actions  Edit Metadata  Copy record  Submit for publication  Delete Metadata 	syntactically valid syntactically valid syntactically valid
Organization 2 <input type="checkbox"/> Created: 08 April 2021 <input type="checkbox"/> Updated: 08 April 2021 Organization		
Organization <input type="checkbox"/> Created: 08 April 2021 <input type="checkbox"/> Updated: 08 April 2021 Organization		







**EUROPEAN
LANGUAGE
GRID**

RELEASE 2

My grid  Annika Grütznher-Zahn 

Technologies Resources Community Events Documentation About ELG

Add items  My items  Feedback  Go to catalogue 

Please fill in the following fields in order to create a copy of service tool 2

resource name *

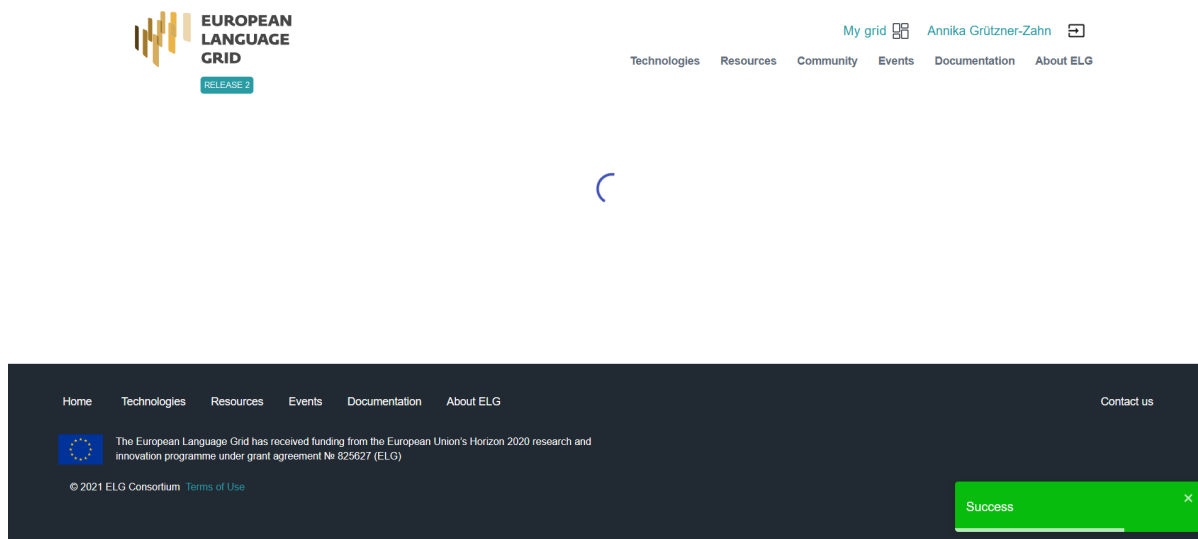
The official name or title of the language resource/technology

Version

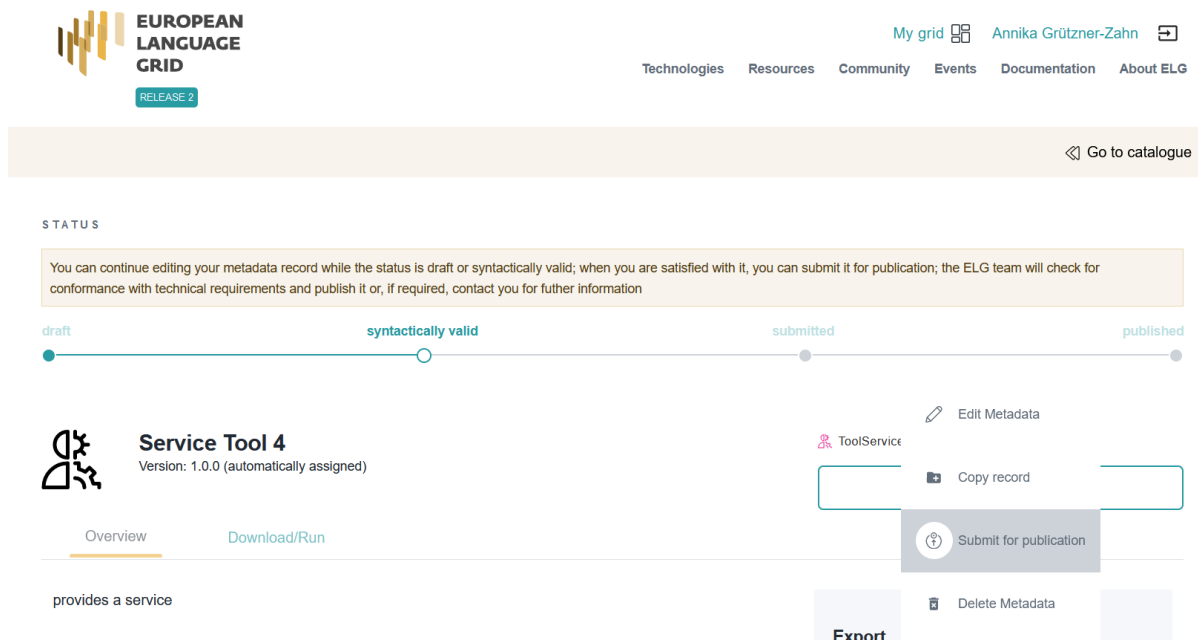
The new version of the record that will be created. Recommended format: major_version.minor_version.patch (see semantic versioning guidelines at <http://semver.org>)

cancel Create copy

You will be notified if the copy has been successfully created.



When the new record is created, you will be redirected to its view page; the status is set to **syntactically valid**, so you can proceed to edit it and submit for publication, following the publication lifecycle for all LRTs.



Note: Please, keep in mind, that if the copy serves as a new version, you have to provide this information in the metadata, in the **replaces** relation field.

When you return to the **My items** page, you will see both versions of the metadata record.

Add items
My items
Feedback
Administration
Go to catalogue

Social network account

The social network account of the project (multiple values allowed)

Fill in

Replaces

The project that has been replaced by the project described here

Project name

Test project v1.1.0

×

Missing **Test project v1.1.0**? Add "Test project v1.1.0"

Test Project v1.1.0

<https://browser.mt/>

My grid
test provider

Technologies
Resources
Community
Events
Documentation
About ELG

Feedback		Go to catalogue	
➔	new test tool 1.1.2 <input type="checkbox"/> Created: 08 March 2021 <input type="checkbox"/> Updated: 08 March 2021 Tool/Service	Actions ▾	syntactically valid
	new test tool 1.0.0 (automatically assigned) <input type="checkbox"/> Created: 07 March 2021 <input type="checkbox"/> Updated: 07 March 2021 Tool/Service	Actions ▾	submitted
➔	testToolR2 1 <input type="checkbox"/> Created: 07 March 2021 <input type="checkbox"/> Updated: 07 March 2021 Tool/Service	Actions ▾	published
	test tool <input type="checkbox"/> Created: 07 March 2021 <input type="checkbox"/> Updated: 07 March 2021 Tool/Service	Actions ▾	draft

1.29 Export metadata records

You can export the metadata of all the items you have created in XML format, except for those marked as “draft”.

You can do this through the landing page of the metadata record, as shown on the following figure.

The screenshot shows the European Language Grid (ELG) interface. At the top, there's a navigation bar with the ELG logo, a 'RELEASE 2' badge, and links for 'Technologies', 'Resources', 'Community', 'Events', 'Documentation', and 'About ELG'. User links for 'My grid' and 'test provider' are also present. A 'Go to catalogue' button is in the top right. Below the navigation bar, a 'STATUS' section contains a message: 'You can continue editing your metadata record while the status is draft or syntactically valid; when you are satisfied with it, you can submit it for publication; the ELG team will check for conformance with technical requirements and publish it or, if required, contact you for further information'. A progress bar shows four stages: 'draft' (selected), 'syntactically valid', 'submitted', and 'published'. Below this, the 'new relation' record is displayed with a 'RelNew' icon and version '1.0.4 (13/08/1967)'. There are tabs for 'Overview' (active), 'Download', and 'Related LRTs'. A 'Corpus' section with an 'Actions' dropdown is on the right. The main content area shows a placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Nulla facilisi nullam vehicula ipsum a arcu cursus vitae congue. Lobortis scelerisque fermentum dui faucibus in ornare quam viverra. Tincidunt nunc pulvinar sapien et ligula ullamcorper malesuada proin libero. Ullamcorper velit sed ullamcorper morbi tincidunt ornare. Diam sollicitudin tempor id eu nisl nunc mi. Risus in hendrerit gravida rutrum quisque. Fusce ut placerat orci nulla pellentesque ... Read More'. On the right, an 'Export' section shows 'XML' as the selected format.

1.30 Delete catalogue items

If you are not satisfied with the description of an item, you can delete its metadata record.

You can do this through the **My items** page for a single or multiple items, or the landing page of the metadata record.

1.31 Overview

This chapter is for **validators**, i.e. users assigned to validate items submitted for publication. At present, only members of the ELG consortium can assume the validator role.

In the following sections, you will be provided with information on the validation procedure. Depending on the type and source of the contributed item, a validation process is foreseen as follows:

- for ELG compatible services, validation is performed at the metadata and technical level by the same individual (see *Validate an ELG compatible LT service (at technical/metadata level)*) and at the legal level (see *Validate an ELG compatible LT service or an LRT hosted in ELG at legal level*); all validators are assigned manually by an administrator of the ELG platform
- for resources hosted at ELG, validation is performed at the metadata and technical level by the same individual (see *Validate an LRT hosted in ELG (at technical/metadata level)*) and at the legal level (see *Validate an ELG compatible LT service or an LRT hosted in ELG at legal level*); all validators are automatically assigned by the system (based on work overload) but can also be manually re-assigned by the administrator

- for items with metadata only (e.g., projects, organizations, corpora accessible through other repositories), validation is performed at the metadata level only (see *Validate a “metadata-only record”*); the validator is automatically assigned by the system (based on work overload) but can also be manually re-assigned by the administrator
- for harvested metadata records, no validation is performed.

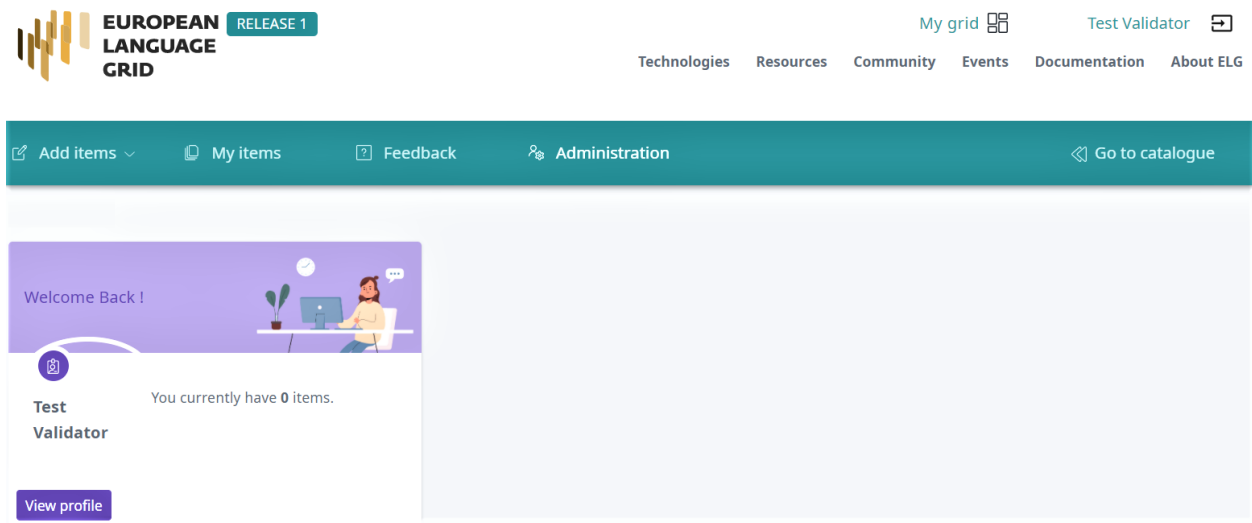
When approved by **all validators**, the item is published at the ELG catalogue.

The following table shows the validation operations foreseen for each item type / source of metadata.

Type of records	Validation type		
	Metadata	Technical	Legal
Harvested metadata	N/A	N/A	N/A
Metadata records uploaded by ELG admin	Yes	N/A	N/A
“Metadata only” records	Yes	N/A	N/A
ELG compatible services	Yes	Yes	Yes
LRTs uploaded (hosted) in ELG	Yes	Yes	Yes

1.32 Access the validation forms

As soon as you are assigned an item to validate, you will receive an email notification. To access the validation form, sign in to ELG, click on **My grid** and select the “Administration” option.



You will be re-directed to the backend administration pages. There, in the management section, you will see one or more of the following validations groups depending on the role/s you have assumed:

- **Legal** validations
- **Technical** validations
- **Metadata** validations

ELG Backend Administration

Site administration

MANAGEMENT

Legal validations

Metadata validations

Technical validations

PROCESSING

LT Service Registration

Recent actions

My actions

None available

1.33 Validate an ELG compatible LT service (at technical/metadata level)

You can access the validation forms as described [here](#).
When you are at the backend administration pages click on `Technical validations` to view the list of metadata records that have been assigned to you.

ELG Backend Administration

WELCOME, TEST. [BACK TO CATALOGUE](#)

Home > Management > Technical validations

Select technical validation to change

q

Search

Action: Go 0 of 7 selected

<input type="checkbox"/> METADATA RECORD	HAS DATA	METADATA VALID	TECHNICALLY VALID	LEGALLY VALID	STATUS
<input type="checkbox"/> [Corpus] testCorpus123	True	?	?	?	ingested
<input type="checkbox"/> [Corpus] corpus35 upload	True	✓	✓	✓	published
<input type="checkbox"/> [Corpus] test corpus	True	✓	✓	✓	published
<input type="checkbox"/> [Corpus] Upload to test s3_2	True	✓	✓	✓	published
<input type="checkbox"/> [LexicalConceptualResource] new test for upload	True	?	?	?	ingested
<input type="checkbox"/> [Corpus] new corps	False	?	✓	✓	ingested
<input type="checkbox"/> [LexicalConceptualResource] test for upload lexicon evening	True	?	?	?	ingested

7 technical validations

FILTER

By legally valid

All

Yes

No

Unknown

By metadata valid

All

Yes

No

Unknown

By technically valid

All

Yes

No

Unknown

By status

All

draft

internal

ingested

published

unpublished

Click on one of the metadata records to access the validation form:

ELG Backend Administration
WELCOME, TEST. [BACK TO CATALOGUE](#)

[Home](#) > [Management](#) > [Technical validations](#) > [ToolService] test service

Change technical validation
 VIEW ON SITE >
LT SERVICE REGISTRATION >
HISTORY

Technically valid: Unknown ▾

Metadata valid: Unknown ▾

Slack channel:

Jira ticket:

review comments (for the provider):

validator notes (internal):

Review comments: -

1.33.1 Technical validation and service registration

On the validation form, the values of `Technically valid` and `Metadata valid` fields are set to `Unknown`; do not change them yet.

Fill in the `Slack channel` and `Jira ticket` fields and click on “Save” at the right bottom corner of the page.

Before filling-in the rest of the validation form, you must first:

- **Deploy the LT service into the ELG kubernetes cluster:** For this you have to create the required yaml file, as in the example below, in the respective GitLab repository and branch.

```
image : "registry.gitlab.com/quarator-platform/dfki/srv-ler:1.0.569798760"
limits_memory: 2048Mi
scalability: "dynamic"
minScale: "1"
```

To create the yaml file, you will need the `docker_download_location`, `service_adapter_download_location` and `execution_location` metadata element values, that you will find on the form. When the yaml file is committed/pushed, the automatic CI/CD deployment pipeline of ELG will be notified and install the LT service to the respective kubernetes cluster; for instance, the master branch is used for the production ELG cluster, while the develop branch is used for the development cluster. You will also need access to the cluster so that you can check which containers/pods are running, inspect the logs and statuses of the containers, etc.; for this, you will receive the required information and credentials from the ELG technical team.

- **Register/integrate the running LT service at the ELG catalogue:** Click on the “LT Service Registration” button at the right top corner of the validation form.

The form will open at a new tab of your browser:

On this form:

- select the `Tool type` value (IE, MT, ASR, etc.), depending on the type of service you validate.

Home › Processing › LT Service Registration › test service

Change LT Service Registration

VIEW ON SITE ›
VALIDATION FORM ›
HISTORY

Metadata record: [ToolService] test service
The metadata record that describes this LT Service

Tool type:

Elg execution location: Currently: <http://noexecution.location.gr>
change:
ELG location of execution of this LT Service

OpenAPI spec url:
URL to the Open API json file

Elg gui url:
URL GUI of the service

Yaml file:

Status:

Accessor id:
A unique accessor id for REST service calls

Execution location:

Docker download location:

Service adapter download location:

Save and continue editing

SAVE

The metadata of the service will help you decide which is the appropriate value; you can view the metadata record by clicking on the “VIEW ON SITE” button on the top right corner of the validation form.

- fill in the `Elg execution location`; the value of this field depends on how the LT service is deployed into the cluster; e.g. knative vs. non-knative (kubernetes) service, which namespace was used, etc. As described [here](#), this field follows this template: `http://{k8s service name for the registered LT tool}.{k8s namespace for the registered LT tool}.svc.cluster.local{the path where the REST service is running at}`. The `{the path where the REST service is running at}` part can be filled based on the `executionLocation` element value of the metadata record. For this reason, the `Elg execution location` field is pre-filled with the `executionLocation` value, so that you can change only the part that is required.
- set the `Elg gui url`; use
 - `/dev/gui-ie/index-mt.html` for MT services,
 - `/dev/gui-udpipe/` for dependency parsers,
 - `/dev/gui-ie/` for IE tools,
 - `/dev/gui-tts/` for TTS services, and
 - `/dev/gui-ie/index-asr.html` for ASRs.

To specify the text direction of the results in the try out UIs you should use the appropriate parameters; e.g. for ASRs use `/dev/gui-ie/index-asr.html?dir=ltr` or `/dev/gui-ie/index-asr.html?dir=rtl` for left-to-right and right-to-left directions, respectively. Similarly, `/dev/gui-ie/?dir=ltr` or `/dev/gui-ie/?dir=rtl` for IE services. For MT services use `srcdir` and `targetdir` parameters (e.g. `index-mt.html?srcdir=ltr&targetdir=rtl`) to specify the text direction of input/output. Depending on the value that you have set, the appropriate try out UI

will be displayed in the respective tab of the landing page of the LT service. If there is no available/appropriate try out UI for the specific service or for any other reason you can disable/hide the try out UI tab by setting `none` to the `Elg_gui_url` field.

- add the `yaml` file location in the respective field; it is the URL of the file that you have created in the previous step. This will help you easily locate in the future which `yaml` file specifies how the service is deployed into the cluster.
- set the `Accessor_id` of the service. This id is unique and it is used for calling the service via the ELG public LT REST API. For a brief overview of the LT REST API, see [this section](#).
- set the value of `status` to `completed` and click on “Save” at the bottom right corner of the page in order to activate the service
- **est the deployed service:** You have to use the Try out tab of the service page (“VIEW ON SITE” will lead you there) or with any other available clients/tools. You must ensure that the service follows the ELG specifications and works as expected. For any issues that arise, you can use the specified slack channel to communicate with the provider of the service.

Now that the service is tested and integrated into ELG, you can return to the technical/metadata validation form and set `Technically valid` to `Yes`.

Then you have to check the metadata of the service.

1.33.2 Metadata validation

As already mentioned you can see the record by clicking the “VIEW ON SITE” button. You are asked to check whether the values of the following elements are included in the metadata record and whether their values match the description of the service:

- `function`: important for findability purposes
- `input & output language(s)`: for MT services, the output language(s) must be included; for services of other types, the output language is not recommended (redundant information)
- `input & output data type(s)`: important for findability and interoperability purposes
- `output annotation type(s)`: if the tool is of the IE type, it’s recommended that this element has values for the types of information annotated/extracted
- `resource creator(s)` and `publication date`: although not mandatory, they are useful for citation purposes;
- `domain(s)`: recommended for findability purposes; if possible, recommend the use of an existing value.
- `documentation`: user and installation manuals for services are recommended; publications describing the use of the resource are also welcome
- `distribution(s)`: if a resource is available in multiple forms (e.g. as a functional service, but also as source code or downloadable form), it’s recommended to describe them as different distributions
- `software distribution form`: check the values at https://european-language-grid.readthedocs.io/en/release1.1.1/Documentation/ELG-SHARE_xsd.html#SoftwareDistributionForm; depending on the form, a different element (access, download, execution or docker download location) is recommended.

If you are satisfied, set the values of `Metadata valid` to `Yes` and click on “Save”.

If not, add your comments and recommendations at the `Review comments` field and set the value of `Metadata valid` to `No` and click on “Save”, as below:

The provider will be notified by email (containing the review comments) in order to update the record. Once finished, the provider will re-submit the record for publication and you will be notified to perform the validation again.

ELG Backend Administration
WELCOME, TEST. BACK TO CATALOGUE

Home » Management » Technical validations » [ToolService] test service

Change technical validation
VIEW ON SITE
LT SERVICE REGISTRATION
HISTORY

Technically valid: No

Metadata valid: No

Slack channel:

Jira ticket:

review comments (for the provider):

The "function" element doesn't fit the description you have provided.
The docker image is password protected; please provide the credentials.

validator notes (internal):

Review comments: -

Validator notes: -

Please, keep in mind that an item is published only when it has been approved at all validation levels (technical, metadata and legal).

Note: This section is under development and continuously updated with new information. The forms included in the current release are implemented with Django. New forms will be provided later.

1.34 Validate an LRT hosted in ELG (at technical/metadata level)

You can access the validation forms as described [here](#).

When you are at the backend administration pages click on `Technical validations` to view the list of metadata records that have been assigned to you.

Click on one of the metadata records to access the validation form:

On this form, you can click on “DOWNLOAD” to download the uploaded files and “VIEW ON SITE” to view the metadata record.

Please, check that the LRT is as expected, i.e.

- no malicious files are contained
- the data format is as set in the metadata record

You are asked to check whether the values of the following elements are included in the metadata record and whether their values match the description and contents of the dataset:

- language(s), linguality & multilinguality type: important for findability purposes
- resource creator(s) and publication date: although not mandatory, they are useful for citation purposes;
- domain(s): recommended for findability purposes; if possible, recommend the use of an existing value

ELG Backend Administration

WELCOME, **TEST** [BACK TO CATALOGUE](#)[Home](#) > [Management](#) > [Technical validations](#)

Select technical validation to change

 Action: 0 of 7 selected

<input type="checkbox"/> METADATA RECORD	HAS DATA	METADATA VALID	TECHNICALLY VALID	LEGALLY VALID	STATUS
<input type="checkbox"/> [Corpus] testCorpus123	True	?	?	?	ingested
<input type="checkbox"/> [Corpus] corpus35 upload	True	✓	✓	✓	published
<input type="checkbox"/> [Corpus] test corpus	True	✓	✓	✓	published
<input type="checkbox"/> [Corpus] Upload to test s3_2	True	✓	✓	✓	published
<input type="checkbox"/> [LexicalConceptualResource] new test for upload	True	?	?	?	ingested
<input type="checkbox"/> [Corpus] new corps	False	?	✓	✓	ingested
<input type="checkbox"/> [LexicalConceptualResource] test for upload lexicon evening	True	?	?	?	ingested

7 technical validations

FILTER

By legally valid

[All](#)
[Yes](#)
[No](#)
[Unknown](#)

By metadata valid

[All](#)
[Yes](#)
[No](#)
[Unknown](#)

By technically valid

[All](#)
[Yes](#)
[No](#)
[Unknown](#)

By status

[All](#)
[draft](#)
[internal](#)
[ingested](#)
[published](#)
[unpublished](#)

ELG Backend Administration

WELCOME, **TEST** [BACK TO CATALOGUE](#)[Home](#) > [Management](#) > [Technical validations](#) > [Corpus] testCorpus123

Change technical validation

[VIEW ON SITE](#) [DOWNLOAD DATA](#) [HISTORY](#)Technically valid: Metadata valid: Slack channel: Jira ticket:

review comments (for the provider):

validator notes (internal):

Review comments:

- data format(s): the values “unspecified” or “other” must be avoided; if needed, you can use a broader term from the ontology
- media type(s): check that they correspond to the contents; please use “text” for transcribed speech corpora; “audio” is to be used only for data resources in audio formats
- corpus and lexical/conceptual resource subclass: important for findability purposes
- encoding level(s) (for Lexical/conceptual resources): “unspecified” or “other” must be avoided; if needed, a broader term can be used
- content type(s): recommended for findability purposes
- documentation: user and installation manuals for tools are recommended; publications describing the use of the resource are also welcome
- distribution(s): if a resource is available in multiple formats, it’s recommended to describe them as different distributions
- size: a meaningful size unit depending on the resource type can be recommended (e.g. translation units for TMX files)
- dataset distribution form: check the values at https://european-language-grid.readthedocs.io/en/release1.1.2/Documentation/ELG-SHARE_xsd.html#DatasetDistributionForm; depending on the form, a different element (access, download or distribution location) is recommended.

If you are satisfied, set the values of `Technically valid` and `Metadata valid` to `Yes` and click on “Save”.

If not, add your comments and recommendations at the `Review comments` field and set the value of `Technically valid` and/or `Metadata valid` (depending on the source of the issue) to `No` and click on “Save”, as below:

ELG Backend Administration WELCOME, TEST. [BACK TO CATALOGUE](#)

Home › Management › Technical validations › [Corpus] testCorpus123

Change technical validation VIEW ON SITE DOWNLOAD DATA HISTORY

Technically valid:

Metadata valid:

Slack channel:

Jira ticket:

review comments (for the provider):

The corpus you have uploaded is an English - French pair but you have provided a description for a German-French one. Please, either correct the description or replace the content files.

validator notes (internal):

Review comments: -

The provider will be notified by email (containing the review comments) in order to update the record. Once finished, the provider will re-submit the record for publication and you will be notified to perform the validation again.

Please, keep in mind that an item is published only when it has been approved at all validation levels (technical, metadata and legal).

Note: This section is under development and continuously updated with new information. The forms included in the current release are implemented with Django. New forms will be provided later.

1.35 Validate an ELG compatible LT service or an LRT hosted in ELG at legal level

You can access the validation forms as described [here](#).

When you are at the backend administration pages click on `Legal validations` to view the list of metadata records that have been assigned to you.

ELG Backend Administration
WELCOME, TEST BACK TO CATALOGUE

Home > Management > Legal validations

Select legal validation to change

Action: ----- 0 of 5 selected

<input type="checkbox"/> METADATA RECORD	HAS DATA	METADATA VALID	TECHNICALLY VALID	LEGALLY VALID	STATUS
<input type="checkbox"/> [Corpus] testCorpus123	True				ingested
<input type="checkbox"/> [Corpus] corpus35 upload	True				published
<input type="checkbox"/> [Corpus] test corpus	True				published
<input type="checkbox"/> [Corpus] Upload to test s3_2	True				published
<input type="checkbox"/> [Corpus] new corps	False				ingested

5 legal validations

FILTER

By legally valid
All
Yes
No
Unknown

By metadata valid
All
Yes
No
Unknown

By technically valid
All
Yes
No
Unknown

By status
All
draft
internal
ingested
published
unpublished

Click on one of the metadata records to access the validation form. On this form, you can click on “DOWNLOAD” to download the uploaded files and “VIEW ON SITE” to view the metadata record.

Please, check the value(s) of the `Licence` element on the metadata record to identify potential legal issues, such as:

- if the LRT comes from a pilot project, whether it fulfils the legal obligations, i.e. that it has been assigned an open licence, or, at least, a licence that allows free use for academic and research purposes
- in the case of standard licences, please make sure that the licence name and URL is the one from the ELG list of licences
- for non standard licences, a unique human readable name is recommended; in any case, the licence URL must link to a page that contains the licensing terms for the item
- for newly published items, recommend the use of a standard open licence and appropriate to the data type (e.g. Creative Commons for data resources, open source code licences for software)

ELG Backend Administration
WELCOME, TEST. BACK TO CATALOGUE

Home › Management › Legal validations › [Corpus] testCorpus123

Change legal validation

VIEW ON SITE
DOWNLOAD DATA
HISTORY

Legally valid:

Unknown

review comments (for the provider):

validator notes (internal):

Review comments:
-

Validator notes:
-

- in case of LRTs with multiple licensing terms (e.g. for commercial and non-commercial use, cf. <https://live.european-language-grid.eu/catalogue/#/resource/service/corpus/2540>), that the licences are consistently used
- in case of LRTs that combine different resource types (e.g. a lexicon or corpus available via an i/f, or a tool that incorporates models or grammars), it's possible that the LRT is available with different licences for each resource type (e.g. one licence for the data and another one for the software). In this case, the licences must appear on the same distribution.

If you are satisfied, set the value of `Legally valid` to `Yes` and click on “Save”.

If not, add your comments and recommendations at the `Review comments` field and set the value of `Legally valid` to `No` and click on “Save”, as below:

The provider will be notified by email (containing the review comments) in order to update the record. Once finished, the provider will re-submit the record for publication and you will be notified to perform the validation again.

Please, keep in mind that an item is published only when it has been approved at all validation levels (technical, metadata and legal).

Note: This section is under development and continuously updated with new information. The forms included in the current release are implemented with Django. New forms will be provided later.

1.36 Validate a “metadata-only record”

You can access the validation forms as described [here](#).

When you are at the backend administration pages click on `Metadata validations` to view the list of metadata records that have been assigned to you.

Click on one of the metadata records to access the validation form:

On this form, you can click on “VIEW ON SITE” to view the metadata record.

ELG Backend Administration

WELCOME, TEST. BACK TO CATALOGUE

Home › Management › Legal validations › [Corpus] testCorpus123

Change legal validation

[VIEW ON SITE](#) [DOWNLOAD DATA](#) [HISTORY](#)Legally valid: No

review comments (for the provider):

The chosen licence is for software, therefore inappropriate for corpora. Please, correct the mistake.

validator notes (internal):

Review comments: -

Validator notes: -

Slack channel: -

Jira ticket: -

ELG Backend Administration

WELCOME, TEST. BACK TO CATALOGUE

Home › Management › Metadata validations

Select metadata validation to change

 SearchAction: ----- Go 0 of 3 selected

<input type="checkbox"/> METADATA RECORD	HAS DATA	METADATA VALID	TECHNICALLY VALID	LEGALLY VALID	STATUS
<input type="checkbox"/> [Organization] Voxit AB	False	✓	✓	✓	published
<input type="checkbox"/> [Project] European Language Resource Coordination LOT2	False	✓	✓	✓	published
<input type="checkbox"/> [Corpus] new corps	False	?	✓	✓	ingested

3 metadata validations

FILTER

By legally valid

[All](#)
[Yes](#)
[No](#)
[Unknown](#)

By metadata valid

[All](#)
[Yes](#)
[No](#)
[Unknown](#)

By technically valid

[All](#)
[Yes](#)
[No](#)
[Unknown](#)

By status

[All](#)
[draft](#)
[internal](#)
[ingested](#)
[published](#)
[unpublished](#)

ELG Backend Administration
WELCOME, TEST. BACK TO CATALOGUE

Home » Management » Metadata validations » [Corpus] new corps

Change metadata validation
 VIEW ON SITE
HISTORY

Metadata valid: Unknown

review comments (for the provider):

validator notes (internal):

Review comments: -

You are asked to check whether the values of the following elements are included in the metadata record and whether their values match the description and contents of the resource:

- the value for language(s) corresponds to that in the description;
- linguality & multilinguality type: important for findability purposes
- if it's a tool/service, the function value(s) are as in the description;
- the “description” provides helpful information
- resource creator(s) and publication date: although not mandatory, they are useful for citation purposes;
- domain(s): recommended for findability purposes; if possible, recommend the use of an existing value
- corpus and lexical/conceptual resource subclass: important for findability purposes
- media type(s): check that they correspond to the contents; please use “text” for transcribed speech corpora; “audio” is to be used only for data resources in audio formats
- encoding level(s) (for Lexical/conceptual resources): “unspecified” or “other” must be avoided; if needed, a broader term can be used
- content type(s): recommended for findability purposes
- distribution(s): if a resource is available in multiple formats, it's recommended to describe them as different distributions
- size: a meaningful size unit depending on the resource type can be recommended (e.g. translation unit(s) for TMX files)
- dataset distribution form: check the values at https://european-language-grid.readthedocs.io/en/release1.1.1/Documentation/ELG-SHARE_xsd.html#DatasetDistributionForm; depending on the form, a different element (access, download or distribution location) is recommended.
- licence name and URL: if it's one of the standard licences, please make sure that the licence name is the one from the ELG list of licences; in any case, the licence URL must link to a page that contains the licensing terms for the item
- hyperlinks: check for broken links.

If you are satisfied, set the value of `Metadata valid` to `Yes` and click on “Save”.

If not, add your comments and recommendations at the `Review comments` field and set the value of `Metadata valid` to `No` and click on “Save”, as below:

Home > Management > Metadata validations > [Corpus] new corps

Change metadata validation VIEW ON SITE > HISTORY

Metadata valid: No ▾

review comments (for the provider):

Please add information on the domain(s) covered; if possible, add existing values according to specific schemata.

validator notes (internal):

Review comments: -

The provider will be notified by email (containing the review comments) in order to update the record. Once finished, the provider will re-submit the record for publication and you will be notified to perform the validation again.

When you have approved the metadata record, it will be automatically published on the ELG catalogue.

Note: This section is under development and continuously updated with new information. In addition, the forms included in the current release are implemented with Django. New forms will be provided later.

1.37 ELG operations

This chapter is for users who serve ELG in **administrative** roles, i.e., the ELG technical team. It is a growing collection of pages documenting internal processes in the interest of transparency and knowledge transfer.

1.38 Catalogue administration

Note: This section is under development. It is continuously updated with new information.

This section is for administrators of the ELG platform only.

1.38.1 Assigning validators for ELG compatible services

When a metadata record for an ELG compatible service has been submitted for publication, you will receive a notification via email and you must proceed to assign the technical/metadata validator and the legal validator for it. Go to the `REGISTRY` section of the administrator pages and click on `Metadata records`. Select the record of the service that you want to assign (as shown below)

ELG Backend Administration

Home » Registry » Metadata Records

Select Metadata Record to change

Search

Action: 1 of 100 selected

RESOURCE TITLE	ID	TYPE	STATUS	METADATA CURATOR	FUNCTIONAL SERVICE	REVIEW STATUS	ACTIVE	ACTION
European Language Grid	1036	Project	INT	test-provider		-		
aNewService	1035	ToolService	ING	galanisd		APPROVED		Edit Service Registration
new project	1034	Project	INT	test-provider		-		
test service_1	1033	ToolService	ING	mltos		Pending		Edit Service Registration

FILTER

By Status

All Published Unpublished Ingested Internal Draft

By Type

All Project

and use the drop down menu to select the validators to which the record will be assigned.

ELG Backend Administration

Home » Registry » Metadata Records

Select Metadata Record to change

Search

Action: 1 of 100 selected

RESOURCE TITLE	ID	TYPE	STATUS	METADATA CURATOR	FUNCTIONAL SERVICE	REVIEW STATUS	ACTIVE	ACTION
European Language Grid	1036	Project	INT	test-provider		-		
aNewService	1035	ToolService	ING	galanisd		APPROVED		Edit Service Registration
new project	1034	Project	INT	test-provider		-		
test service_1	1033	ToolService	ING	mltos		Pending		Edit Service

FILTER

By Status

All Published Unpublished Ingested Internal Draft

By Type

All Project

1.38.2 Approving or rejecting “claimed” metadata records

When one or more metadata records have been “claimed”, you will receive a notification informing you of the metadata record(s) and the user that has made the request. Next, proceed to the `REGISTRY` section of the administrator pages and click on `Metadata records`. Select the record that has been claimed, select the action “Accept claim of selected” or “Reject claim of selected” from the drop-down list “Action” and click “Go”.

If the user is not yet a provider, you will see a warning message. In this case, proceed to keycloak in order to assign him/her the provider role, and then come back to the catalogue to continue the approval/rejection of the claim.

If you select to reject the claim, you will be prompted to add a reason for this which will be communicated to the user that has made the claim.

1.39 Feedback and Helpdesk

We welcome your feedback (questions, suggestions, problems) regarding the ELG platform.

You can

- use our [contact form](#) to send inquiries of any type to the ELG consortium or
- participate in the discussion and check previous feedback at our dedicated GitLab repository: <https://gitlab.com/european-language-grid/platform/elg-platform>.

ELG Backend Administration
WELCOME, PENNY. BACK TO CATALOGUE

Home » Registry » Metadata Records

Select Metadata Record to change

Action: 1 of 100 selected

- Internalize selected
- Ingest selected
- Publish selected
- Unpublish selected
- Mark selected as deleted
- Register Selected Services with ELG
- Accept claim of selected
- Reject claim of selected
- Assign Curator
- Assign Legal Validator
- Assign Metadata Validator
- Assign Technical Validator

ID	TYPE	STATUS	METADATA CURATOR	CLAIMED	FUNCTIONAL SERVICE	REVIEW STATUS	ACTION
4654	Organization	PUB	tuba.arslan@sestek.com			-	-
4653	ToolService	ING	martin@kamusi.org			PENDING	Edit Service Registration
4652	Project	PUB	i.roberts@sheffield.ac.uk			-	-
4651	Organization	PUB	elg-system			-	-
4650	Organization	PUB	elg-system			-	-
4649	Organization	PUB	elg-system			-	-
4648	Organization	PUB	elg-system			-	-
4647	Organization	PUB	elg-system			-	-
4646	Organization	PUB	elg-system			-	-
4645	Organization	PUB	elg-system			-	-
4644	Organization	PUB	elg-system			-	-

FILTER

By Status

- All
- Published
- Unpublished
- Ingested
- Internal
- Draft

By Type

- All
- Project
- Organization
- Licence
- Person
- Group
- Document
- Tool/Service
- Corpus
- Lexical Conceptual Resource
- Language Description
- Repository

By Functional Service

To access the GitLab repository, you will need to have a GitLab account; if you don't have one already, it's free and easy to register at https://gitlab.com/users/sign_up.

Once logged in, you can join the ongoing conversation at our list of issues: <https://gitlab.com/european-language-grid/platform/elg-platform/-/issues>, or create a new issue at: <https://gitlab.com/european-language-grid/platform/elg-platform/-/issues/new>.

We recommend the use of labels for easier tracking:

- clarification: for asking questions on the use of the platform or metadata descriptions
- error: for bugs or technical issues you've run into while using the platform
- feature: suggestions for improvements of existing features or adding new features.

You can also use our [contact form](#) to send inquiries of any type to the ELG consortium.

1.40 Python SDK

With the ELG Python SDK, you can use LT services and search the catalogue inside your Python projects.

Note: Find details of all the classes in *Main Classes*.

1.40.1 Installation

Via pip / PyPI

```
pip install elg
```

Register on the ELG

Please visit the [ELG website](#) to create a user account if you haven't got one already.

1.40.2 Getting started

1.40.3 Functionalities

Use LTs

Initialize LT service

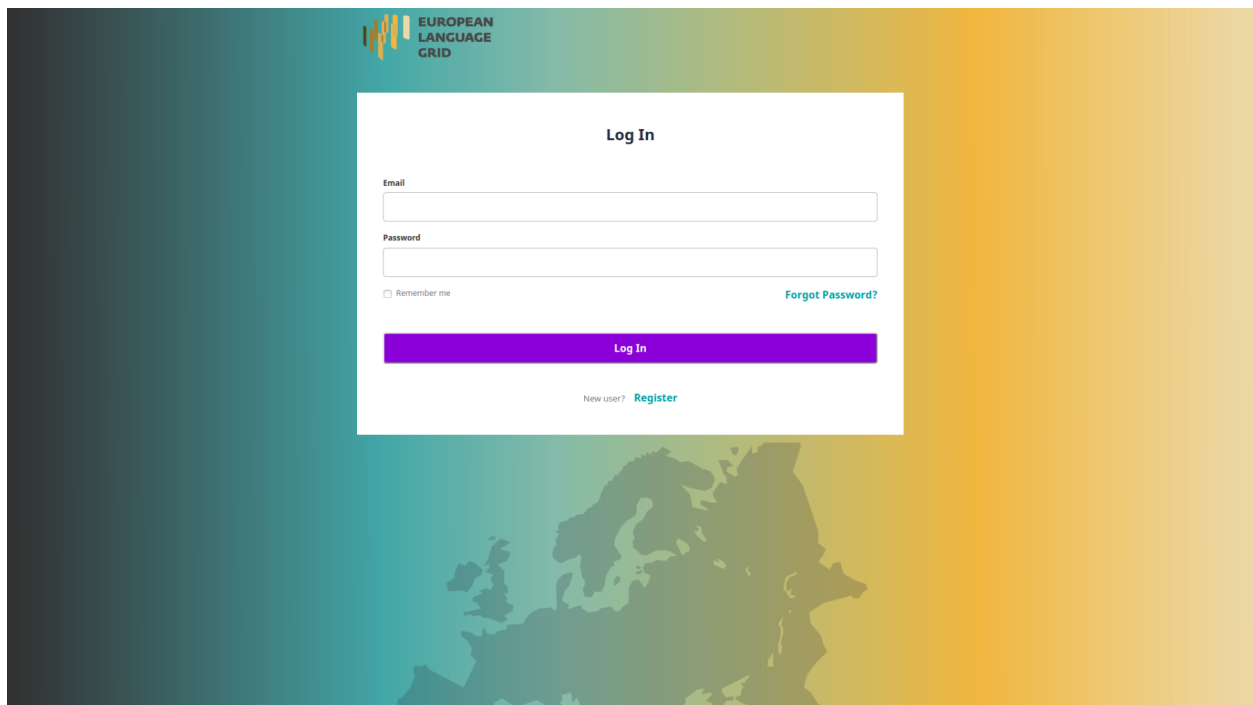
LT Services can be initialized as `Service` objects using the `service_id` parameter and an integer corresponding to the LT service you want to use.

In the following code example, 474 corresponds to the [Cogito Discover Named Entity Recognizer](#) (notice the ID at the end of the URL). For more LT services, search the catalogue via the `Catalog` functionality (see section below) or visit the [list of LT services on the ELG](#).

```
from elg import Service

# Init LT service using its ID
lt = Service.from_id(474)
```

This requires you to login to your ELG account via the URL that is printed on your terminal.



After successful login, your tokens are saved as `~/.cache/elg/tokens.json`, so you do not need to log in again for subsequent calls.

Run LT service

You can either pass an input file or a string / raw text.

```
# Pass an input file that should be processed by the LT service
result = lt("path/to/file")

# You can also directly pass raw text to the LT service in most cases
result = lt("Did Nikola Tesla live in Berlin?")
```

Use corpus

```
from elg import Corpus

corpus = Corpus.from_id(913)
corpus.download()
```

Use the catalogue

```
from elg import Catalog

catalog = Catalog()

# Search and get the result as a list of Entity
results = catalog.search(
    resource = "Tool/Service", # "Corpus", "Lexical/Conceptual resource" or "Language_
↪description"
    function = "Machine Translation", # function should be pass only if resource is_
↪set to "Tool/Service"
    languages = ["en", "fr"], # string or list if multiple languages
    limit = 100,
)

# search interactively
catalog.interactive_search(
    search = "keyword1 keyword2 ...",
    resource = "Tool/Service", # "Corpus", "Lexical/Conceptual resource" or "Language_
↪description"
    function = "Machine Translation", # function should be pass only if resource is_
↪set to "Tool/Service"
    languages = ["en", "fr"], # string or list if multiple languages
)
```

Create a LT service object from the results

```
service = Service.from_entity(results[0])
result = service("Did Nikola Tesla live in Berlin?")
```

Get info of an entity

```
from elg import Entity

entity = Entity.from_id(476)
print(entity)
```

Benchmark

You can also run a benchmark that evaluates multiple services receiving the same input:

```
from elg import Benchmark

ben = Benchmark.from_ids([610, 624])

result = ben(["Bush is the president of the USA and lives in Washington", "My name is _  
↳Dupond and I live in France"], number_of_runs=4)

df = result.compare()
print("General comparison:\n", df)
df = result.compare_results()
print("Comparison of the results:\n", df)
df = result.compare_response_times()
print("Comparison of the response time:\n", df)
```

You can investigate the results by saving the output from the bench call to a variable or by accessing bench.services.

CLI

```
elg run --help
elg download --help
elg info --help
elg search --help
```

1.41 Integrate a service into ELG using the python SDK

Example with the integration of a NER model from HuggingFace: <https://huggingface.co/elastic/distilbert-base-cased-finetuned-conll03-english>

1.41.1 0. Set up the environment

Before starting, we will set up a new environment. Let's create a new folder:

```
mkdir distilbert-ner-en
cd distilbert-ner-en
```

And now a new environment:

```
conda create -n python-services-distilbert-ner-en python=3.7
conda activate python-services-distilbert-ner-en
```

We will also install the packages needed:

```
pip install torch transformers
```

1.41.2 1. Have the model running locally

The first step is to have the model we want to integrate into ELG running locally. In our case, we will firstly download the model using `git lfs`:

```
git lfs install
git clone https://huggingface.co/elastic/distilbert-base-cased-finetuned-conll03-
→english
```

And then create a simple python script that run the model (we can call it `use.py`):

```
from transformers import pipeline

class DistilbertNEREn:

    nlp = pipeline("ner", "distilbert-base-cased-finetuned-conll03-english")

    def run(self, input_str):
        return self.nlp(input_str)
```

We can have a try using the Python Interpreter:

```
>>> from use import DistilbertNEREn
>>> model = DistilbertNEREn()
>>> model.run("Albert is from Germany.")
[{'word': 'Albert', 'score': 0.9995226263999939, 'entity': 'B-PER', 'index': 1}, {
→ 'word': 'Germany', 'score': 0.9997316598892212, 'entity': 'B-LOC', 'index': 4}]
```

The model is working well but is not yet ELG compatible.

1.41.3 2. Create an ELG compatible service

To create the ELG compatible service, we will use the ELG Python SDK installable through PIP:

```
pip install elg
```

Then we need to make our model inherits from the `FlaskService` class of `elg` and uses `ELG Request` and `Response` object. Let's create a new python file called `elg_service.py`:

```
from transformers import pipeline

from elg import FlaskService
from elg.model import AnnotationsResponse

class DistilbertNEREn(FlaskService):

    nlp = pipeline("ner", "distilbert-base-cased-finetuned-conll03-english")

    def convert_outputs(self, outputs, content):
        annotations = {}
        offset = 0
        for output in outputs:
            word = output["word"]
            score = output["score"]
            entity = output["entity"]
            start = content.find(word) + offset
            end = start + len(word)
            content = content[end - offset :]
            offset = end
            if entity not in annotations.keys():
                annotations[entity] = [
                    {
                        "start": start,
                        "end": end,
                        "features": {
                            "word": word,
                            "score": score,
                        },
                    },
                ]
            else:
                annotations[entity].append(
                    {
                        "start": start,
                        "end": end,
                        "features": {
                            "word": word,
                            "score": score,
                        },
                    },
                )
        return AnnotationsResponse(annotations=annotations)

    def process_text(self, content):
        outputs = self.nlp(content.content)
        return self.convert_outputs(outputs, content.content)

flask_service = DistilbertNEREn("distilbert-ner-en")
```

(continues on next page)

(continued from previous page)

```
app = flask_service.app
```

We also need to initialize the service at the end of the file and create the app variable.

We can now test our service from the Python Interpreter to make sure that everything is working:

```
>>> from elg_service import DistilbertNEREn
>>> from elg.model import TextRequest
>>> service = DistilbertNEREn("distilbert-ner-en")
>>> request = TextRequest(content="Albert is from Germany.")
>>> response = service.process_text(request)
>>> print(response)
type='annotations' warnings=None features=None annotations={'B-PER': [
  ↳ [Annotation(start=0, end=6, sourceStart=None, sourceEnd=None, features={'word':
  ↳ 'Albert', 'score': 0.9995226263999939})], 'B-LOC': [Annotation(start=15, end=22,
  ↳ sourceStart=None, sourceEnd=None, features={'word': 'Germany', 'score': 0.
  ↳ 9997316598892212})]}]
>>> print(type(response))
<class 'elg.model.response.AnnotationsResponse.AnnotationsResponse'>
```

Our service is using an ELG Request object as input and an ELG Response object as output. It is therefore ELG compatible.

1.41.4 3. Generate the Docker image

To generate the Docker image, the first step is to create a Dockerfile. To simplify the process, you can use the elg CLI and run:

```
elg docker create --path ./elg_service.py --classname DistilbertNEREn --required_
↳ folders distilbert-base-cased-finetuned-conll03-english --requirements torch --
↳ requirements transformers
```

It will generate the Dockerfile but also a file called `docker_entrypoint.sh` used to start the service inside the container, and a `requirements.txt` file.

We can now build the Docker image (here you need to replace with the name of your docker registry):

```
docker build -t airklizz/distilbert-ner-en:v1 .
```

It will generate a Docker image locally that we can test using the ELG Python SDK as follows:

```
>>> from elg import Service
>>> service = Service.from_docker_image("airklizz/distilbert-ner-en:v1", "http://
↳ localhost:8000/process", 8181)

# at that step, we will be asked to run the docker image locally using the command:
↳ `docker run -p 127.0.0.1:8181:8000 airklizz/distilbert-ner-en:v1`

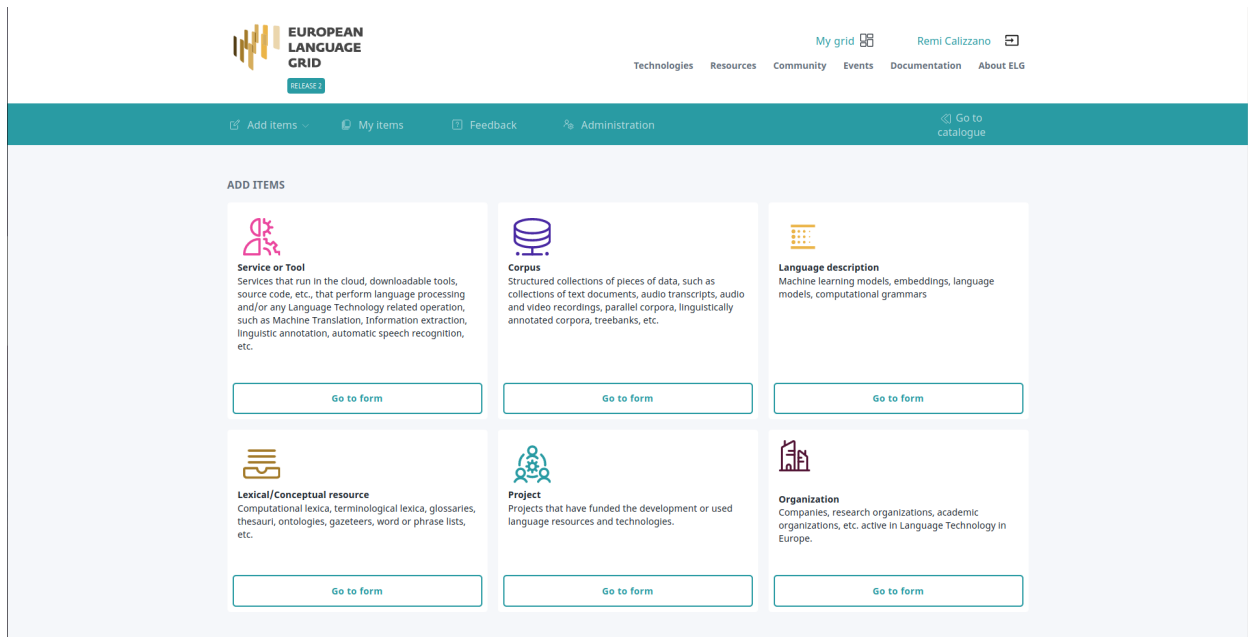
>>> response = service("Albert is from Germany.", sync_mode=True)
>>> print(response)
type='annotations' warnings=None features=None annotations={'B-PER': [
  ↳ [Annotation(start=0, end=6, sourceStart=None, sourceEnd=None, features={'word':
  ↳ 'Albert', 'score': 0.9995226263999939})], 'B-LOC': [Annotation(start=15, end=22,
  ↳ sourceStart=None, sourceEnd=None, features={'word': 'Germany', 'score': 0.
  ↳ 9997316598892212})]}]
↳ 9997316598892212})]}]
```

The image is working and is compatible with ELG so we can push it on Docker Hub (or the Docker registry of your choice):

```
docker push airklizz/distilbert-ner-en:v1
```

1.41.5 4. Create a new service on ELG

You have to go to the ELG, have a provider account and then you can add a new ‘Service or Tool’.



You have to fill all the information needed.

The screenshot shows the 'ADD ITEMS' form in the ELG interface, specifically for creating a new 'TOOL/SERVICE'. The form is divided into several sections. The 'IDENTITY' section includes 'LRT name' (set to 'DistilBERT NER En') and 'Language' (set to 'English'). The 'LRT Identifier' section includes 'LRT short name' (set to 'distilbert-ner-en') and 'Language' (set to 'English'). The 'Description' section includes a text area with the description: 'Model from the HuggingFace Hub: https://huggingface.co/elastic/distilbert-base-cased-finetuned-conll03-english' and 'Language' (set to 'English'). The 'Version' section includes 'Version' (set to 'v1') and 'Version date'. There are also checkboxes for 'Work in progress' and 'ELG-compatible service', and buttons for 'Save draft' and 'Save'.

And then save and create the item. You will be redirect to the landing page of the service.

EUROPEAN LANGUAGE GRID

My grid Remi Calizzano

Technologies Resources Community Events Documentation About ELG

[Go to catalogue](#)

STATUS

You can continue editing your metadata record while the status is draft or syntactically valid; when you are satisfied with it, you can submit it for publication; the ELG team will check for conformance with technical requirements and publish it or, if required, contact you for further information

draft syntactically valid submitted published

DistilBERT NER En
distilbert-ner-en
Version: 0.1 (30/04/2021)
Functional service

ToolService

Actions

Overview Download/Run

Model from the HuggingFace Hub: <https://huggingface.co/elastic/distilbert-base-cased-finetuned-conll03-english> DistilBERT base cased, fine-tuned for NER using the conll03 english dataset. Note that this model is sensitive to capital letters — "english" is different than "English".

Keyword
named entity recognition
huggingface DistilBERT

Intended application
Named Entity Recognition

Export
XML

Resource provider
 Remi Calizzano
[Email](#)

When the landing page looks good to you, you can submit the service for publication through the ‘Actions’ button.

At that step, the ELG technical team will deploy the Docker image of your service into the ELG Kubernetes cluster. Once it done, the service is tested and published.

The service is now publicly available on ELG.

EUROPEAN LANGUAGE GRID

My grid Remi Calizzano

Technologies Resources Community Events Documentation About ELG

[Go to catalogue](#)

DistilBERT NER En
distilbert-ner-en
Version: 0.1 (30/04/2021)
Functional service

ToolService

Actions

Overview Download/Run Try out Code samples

Model from the HuggingFace Hub: <https://huggingface.co/elastic/distilbert-base-cased-finetuned-conll03-english> DistilBERT base cased, fine-tuned for NER using the conll03 english dataset. Note that this model is sensitive to capital letters — "english" is different than "English".

Keyword
named entity recognition
huggingface DistilBERT

Intended application
Named Entity Recognition

Export
XML

Resource provider
 Remi Calizzano
[Email](#)

Additional information
[Landing page](#)

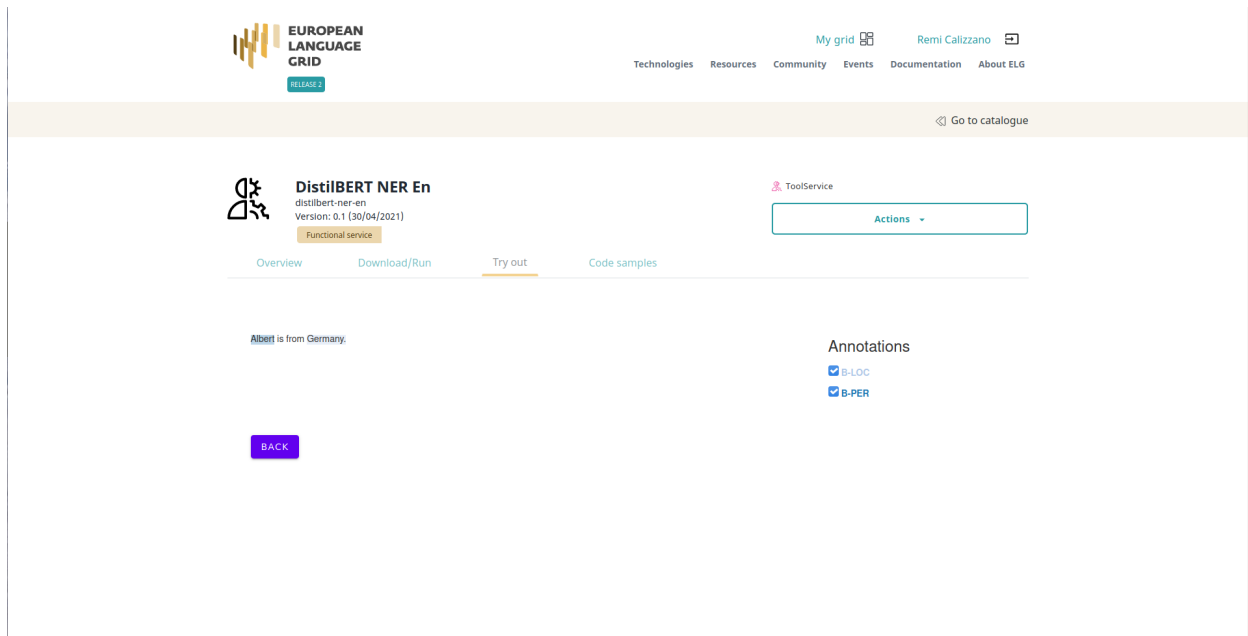
Contact
 Elastic
[Website](#)

Input content resource
Processing resource type
user input text
Marilia Tunes

Function
Function
Named Entity Recognition
Language dependent
false

Output resource
Processing resource type
output text
Data format
CoNLL-2003

And can be run through the UI:



or through the Python SDK (you need to find the id of the service, here: 6366):

```
from elg import Service

service = Service.from_id(6366)
response = service('Albert is from Germany.')
print(response)
# type='annotations' warnings=None features=None annotations={'B-PER': [
  ↳ [Annotation(start=0, end=6, sourceStart=None, sourceEnd=None, features={'word':
  ↳ 'Albert', 'score': 0.9995226263999939})], 'B-LOC': [Annotation(start=15, end=22,
  ↳ sourceStart=None, sourceEnd=None, features={'word': 'Germany', 'score': 0.
  ↳ 9997316598892212})] ] }
```

1.42 Main Classes

Catalog

Authentication

Entity

Corpus

Service

Benchmark

Pipeline

FlaskService

1.43 Catalog

class `elg.catalog.Catalog` (*domain: str = 'live'*)

Class to use the ELG search API. Browse the ELG catalogue using Python.

Examples:

```
from elg import Catalog

# First you have to init a catalog object.
catalog = Catalog()

# Then you can use the search method to search for resources. This method returns
↳ a list of Entity which can be displayed individually.
# For example, we can search for a Machine Translation service for English and
↳ French.
results = catalog.search(
    resource = "Tool/Service", # "Corpus", "Lexical/Conceptual resource" or
↳ "Language description"
    function = "Machine Translation", # function should be pass only if resource
↳ is set to "Tool/Service"
    languages = ["en", "fr"], # string or list if multiple languages
    limit = 100,
)
print(f"Machine Translation service for English and French:\n{results[0]}")

# Another example can be a German NER corpora.
results = catalog.search(
    resource = "Corpus", # "Corpus", "Lexical/Conceptual resource" or "Language
↳ description"
    languages = ["German"], # string or list if multiple languages
    search="ner",
    limit = 100,
)
print(f"German corpus for NER:\n{results[0]}")

# You can init a service from an Entity.

# We can use the catalog to search a Named Entity Recognizer for French and init
↳ a Service with the returned Entity.
results = catalog.search(
    resource = "Tool/Service",
    function = "Named Entity Recognition",
    languages = ["fr"],
    limit = 1,
)
entity = results[0]
print(entity)

from elg import Service

lt = Service.from_entity(entity=entity)
result = lt("Jean Dupond vit à Paris.")
print(f"\n{result}")
```

interactive_search (*entity: str = 'LanguageResource', search: Optional[str] = None, resource: Optional[str] = None, function: Optional[str] = None, languages: Optional[Union[str, list]] = None, license: Optional[str] = None*)

Method to search resources interactively. Warn: not well coded and tested.

search (*entity: str = 'LanguageResource', search: Optional[str] = None, resource: Optional[str] = None, function: Optional[str] = None, languages: Optional[Union[str, list]] = None, license: Optional[str] = None, limit: int = 100*)

Method to send a search request to the API.

Parameters

- **entity** (*str, optional*) – type of the entity to search. Can be ‘LanguageResource’, ‘Organization’, or ‘Project’. Defaults to “LanguageResource”.
- **search** (*str, optional*) – terms to use for the search request. Defaults to None.
- **resource** (*str, optional*) – type of the language resource. Only used when the entity is set to ‘LanguageResource’. Can be ‘Tool/Service’, ‘Lexical/Conceptual resource’, ‘Corpus’, or ‘Language description’. Defaults to None.
- **function** (*str, optional*) – type of the function of the service. Only used when resource set to ‘Tool/Service’. Defaults to None.
- **languages** (*Union[str, list], optional*) – language filter for the search request. Can be a string or a list of string. If it is a list of strings, the results of the request will match will all the languages and not one among all. The full name or the ISO639 code of the language can be used. Defaults to None.
- **license** (*str, optional*) – license filter. Defaults to None.
- **limit** (*int, optional*) – limit number of results. Defaults to 100.

Returns list of the results.

Return type List[elg.Entity]

Examples:

```
results = catalog.search(  
    resource = "Tool/Service",  
    function = "Machine Translation",  
    languages = ["en", "fr"],  
    limit = 100,  
)  
results = catalog.search(  
    resource = "Corpus",  
    languages = ["German"],  
    search="ner",  
    limit = 100,  
)
```

1.44 Authentication

class `elg.authentication.Authentication` (*domain: str*)

Class to authenticate in the ELG using out-of-band authentication.

Parameters

- **base_url** (*str*) – url to authenticate.
- **token_url** (*str*) – url to request tokens.
- **client** (*str*) – name of the Keycloak client.
- **redirect_uri** (*str*) – redirect uri.

create (*scope: str = 'openid'*)

Method to create Keycloak authentication tokens.

Parameters **scope** (*str, optional*) – scope to use when requesting tokens. Can be set to “openid” or “offline_access” to get offline tokens. Defaults to “openid”.

classmethod **create_authentication_url** (*scope: str = 'openid', domain: str = 'live'*) → *str*

Class method to create the Keycloak authentication url.

Parameters

- **scope** (*str, optional*) – scope to use when requesting tokens. Can be set to “openid” or “offline_access” to get offline tokens. Defaults to “openid”.
- **domain** (*str, optional*) – ELG domain you want to use. “live” to use the public ELG, “dev” to use the development ELG and another value to use a local ELG. Defaults to “live”.

Returns Keycloak authentication url.

Return type *str*

classmethod **from_json** (*filename: str*)

Class method to init an Authentication object from a json file.

Parameters **filename** (*str*) – name of the json file.

Returns Authentication object with Keycloak authentication tokens.

Return type `elg.Authentication`

classmethod **from_success_code** (*code: str, domain: str = 'live'*)

Class method to init an Authentication object from a success code.

Parameters

- **code** (*str*) – success code obtained after authentication.
- **domain** (*str, optional*) – ELG domain you want to use. “live” to use the public ELG, “dev” to use the development ELG and another value to use a local ELG. Defaults to “live”.

Returns Authentication object with Keycloak authentication tokens.

Return type `elg.Authentication`

classmethod **init** (*scope: str = 'openid', domain: str = 'live'*)

Class method to init an Authentication object and authenticate to the ELG.

Parameters

- **scope** (*str*, *optional*) – scope to use when requesting tokens. Can be set to “openid” or “offline_access” to get offline tokens. Defaults to “openid”.
- **domain** (*str*, *optional*) – ELG domain you want to use. “live” to use the public ELG, “dev” to use the development ELG and another value to use a local ELG. Defaults to “live”.

Returns Authentication object with Keycloak authentication tokens.

Return type `elg.Authentication`

refresh()

Method to refresh to access_token using the refresh_token.

Raises AuthenticationException – There is no refresh_token to refresh the access_token.

refresh_if_needed()

Method that call the refresh method only if needed, i.e. the access_token is expired.

Raises RefreshTokenExpirationException – The refresh_token is expired.

to_json (*filename: str*)

Save the Keycloak authentication tokens to a json file.

Parameters filename (*str*) – name of the json file.

class `elg.authentication.NeedAuthentication`

Parent class for class which needs authentication. Provide useful methods

`elg.authentication.need_authentication()`

Decorator for methods to refresh to authentication tokens before calling the method

1.45 Entity

class `elg.entity.Entity` (*id: int, resource_name: str, resource_short_name: List[str], resource_type: str, entity_type: str, description: str, keywords: List[str], detail: str, licences: List[str], languages: List[str], country_of_registration: List[str], creation_date: str, last_date_updated: str, functional_service: bool, functions: List[str], intended_applications: List[str], views: int, downloads: int, size: int, service_execution_count: int, status: str, under_construction: bool, domain: str, record: Optional[dict] = None, **kwargs*)

Class to represent every ELG entity

classmethod `from_id` (*id: int, domain: str = 'live', use_cache: bool = True, cache_dir: str = '~/.cache/elg'*)

Class method to init an Entity object from its id.

Parameters

- **id** (*int*) – id of the entity.
- **domain** (*str*, *optional*) – ELG domain you want to use. “live” to use the public ELG, “dev” to use the development ELG and another value to use a local ELG. Defaults to “live”.
- **use_cache** (*bool*, *optional*) – True if you want to use cached files. Defaults to True.

- **cache_dir** (*str*, *optional*) – path to the cache_dir. Set it to None to not store any cached files. Defaults to “~/cache/elg”.

Returns Entity object.

Return type elg.Entity

classmethod from_search_result (*result: dict*, *domain: str* = 'live')

Class method to init an Entity object from a search result.

Parameters

- **result** (*dict*) – result of the search API.
- **domain** (*str*, *optional*) – ELG domain you want to use. “live” to use the public ELG, “dev” to use the development ELG and another value to use a local ELG. Defaults to “live”.

Returns Entity object.

Return type elg.Entity

1.46 Corpus

```
class elg.corpus.Corpus (id: int, resource_name: str, resource_short_name: List[str], resource_type: str, entity_type: str, description: str, keywords: List[str], detail: str, licences: List[str], languages: List[str], country_of_registration: List[str], creation_date: str, last_date_updated: str, functional_service: bool, functions: List[str], intended_applications: List[str], views: int, downloads: int, size: int, service_execution_count: int, status: str, under_construction: bool, record: dict, auth_object: elg.authentication.Authentication, auth_file: str, scope: str, domain: str, use_cache: bool, cache_dir: str)
```

Class to represent a corpus. Download ELG corpora.

Examples:

```
from elg import Corpus

# You can initialize a corpus from its id. You will be asked to authenticate on
↳ the ELG website.
corpus = Corpus.from_id(913)

# You can display the corpus information.
print(corpus)

# You can download the corpus. Note that only corpora hosted on ELG are
↳ downloadable using the python SDK.
corpus.download()

# By default the corpus is downloaded at the current location and the filename is
↳ the name of the ELG corpus.
# You can overwrite this with the folder and filename parameters.
corpus.download(filename="ELG_corpus", folder="/tmp/")

# You can create an corpus from a catalog search result. First you need to search
↳ for a service using the catalog.
```

(continues on next page)

(continued from previous page)

```
# Let's search an English to French Machine Translation service.
from elg import Catalog

catalog = Catalog()
results = catalog.search(
    resource = "Corpus",
    languages = ["German"],
    search="ner",
    limit = 1,
)

corpus = Corpus.from_entity(results[0])
print(corpus)
```

download (*distribution_idx: int = 0, filename: str = None, folder: str = './*)

Method to download the corpus if possible.

Parameters

- **distribution_idx** (*int, optional*) – Index of the distribution of the corpus to download. Defaults to 0.
- **filename** (*str, optional*) – Name of the output file. If None, the name of the corpus will be used. Defaults to None.
- **folder** (*str, optional*) – path to the folder where to save the downloaded file. Defaults to “.”.

classmethod from_entity (*entity: elg.entity.Entity, auth_object: Optional[str] = None, auth_file: Optional[str] = None, scope: Optional[str] = None, use_cache: bool = True, cache_dir='~/cache/elg'*)

Class method to init a Corpus class from an Entity object. You can provide authentication information through the `auth_object` or the `auth_file` attributes. If not authentication information is provided, the Authentication object will be initialized.

Parameters

- **entity** (*elg.Entity*) – Entity object to init as a Corpus.
- **auth_object** (*elg.Authentication, optional*) – `elg.Authentication` object to use. Defaults to None.
- **auth_file** (*str, optional*) – json file that contains the authentication tokens. Defaults to None.
- **scope** (*str, optional*) – scope to use when requesting tokens. Can be set to “openid” or “offline_access” to get offline tokens. Defaults to “openid”.
- **domain** (*str, optional*) – ELG domain you want to use. “live” to use the public ELG, “dev” to use the development ELG and another value to use a local ELG. Defaults to “live”.
- **use_cache** (*bool, optional*) – True if you want to use cached files. Defaults to True.
- **cache_dir** (*str, optional*) – path to the cache_dir. Set it to None to not store any cached files. Defaults to “~/cache/elg”.

Returns Corpus object with authentication information.

Return type `elg.Corporus`


```
classmethod from_id(id: int, auth_object: Optional[elg.authentication.Authentication] = None,
                    auth_file: Optional[str] = None, scope: Optional[str] = None, domain: Optional[str] = None, use_cache: bool = True, cache_dir: str =
                    '~/.cache/elg')
```

Class method to init a Corpus class from its id. You can provide authentication information through the `auth_object` or the `auth_file` attributes. If not authentication information is provided, the Authentication object will be initialized.

Parameters

- **id** (*int*) – id of the corpus.
- **auth_object** (*elg.Authentication*, *optional*) – *elg.Authentication* object to use. Defaults to None.
- **auth_file** (*str*, *optional*) – json file that contains the authentication tokens. Defaults to None.
- **scope** (*str*, *optional*) – scope to use when requesting tokens. Can be set to “openid” or “offline_access” to get offline tokens. Defaults to “openid”.
- **domain** (*str*, *optional*) – ELG domain you want to use. “live” to use the public ELG, “dev” to use the development ELG and another value to use a local ELG. Defaults to “live”.
- **use_cache** (*bool*, *optional*) – True if you want to use cached files. Defaults to True.
- **cache_dir** (*str*, *optional*) – path to the cache_dir. Set it to None to not store any cached files. Defaults to “~/.cache/elg”.

Returns the corpus object initialized.

Return type *elg.Corporus*

```
class elg.corpus.Distribution(corpus_id: int, domain: str, form: str, distribution_location:
                             str, download_location: str, access_location: str, licence:
                             elg.corpus.Licence, cost: str, attribution_text: str)
```

Class to represent a corpus distribution

```
classmethod from_data(corpus_id: int, domain: str, data: dict)
```

Class method to init the distribution object from the metadata information.

Parameters

- **corpus_id** (*int*) – id of the corpus the distribution is from.
- **domain** (*str*) – ELG domain you want to use. “live” to use the public ELG, “dev” to use the development ELG and another value to use a local ELG.
- **data** (*dict*) – metadata information of the distribution.

Returns the distribution object initialized.

Return type *elg.Distribution*

```
is_downloadable() → str
```

Method to get if the distribution is downloadable.

Returns return True is the distribution is downloadable, False if not.

Return type bool

```
class elg.corpus.Licence(name: str, urls: List[str], identifiers: List[dict])
```

Class to represent a licence

1.47 Service

```
class elg.service.Service(id: int, resource_name: str, resource_short_name: List[str], resource_type: str, entity_type: str, description: str, keywords: List[str], detail: str, licences: List[str], languages: List[str], country_of_registration: List[str], creation_date: str, last_date_updated: str, functional_service: bool, functions: List[str], intended_applications: List[str], views: int, downloads: int, size: int, service_execution_count: int, status: str, under_construction: bool, record: dict, auth_object: elg.authentication.Authentication, auth_file: str, scope: str, domain: str, use_cache: bool, cache_dir: str)
```

Class to use ELG service. Run an ELG service directly from python.

Examples:

```
from elg import Service

# You can initialize a service from its id. You will be asked to authenticate on
↳the ELG website.
service = Service.from_id(474)

# You can then directly run the service.
result = service("Nikolas Tesla lives in Berlin.")
print(f"\nResult:\n{result}")

# You can also create a service from a catalog search result.

# First you need to search for a service using the catalog. Let's search an
↳English to French Machine Translation service.
from elg import Catalog

catalog = Catalog()
results = catalog.search(
    resource = "Tool/Service",
    function = "Machine Translation",
    languages = ["en", "fr"],
    limit = 1,
)

# Now you can initialize the service using the first result. You will not be
↳asked to authenticate because your token has been cached.
service = Service.from_entity(results[0])

# And run the service as before.
result = service("ELG is an amazing project.")
print(f"\nResult:\n{result}")

# It is possible to use a file as input when running the service.
with open("/tmp/example.txt", "w") as f:
    f.write("ELG is an amazing project.")

result = service("/tmp/example.txt")
print(f"\nResult:\n{result}")

# You can apply a method to the result to extract the information needed. To do
↳so, you have to pass a
```

(continues on next page)

(continued from previous page)

```
# callable object in the output_func parameter.
service = Service.from_id(5228)
pretty_result = service("Ich habe diesen Film geliebt. Die Schauspieler, das_
↳ Drehbuch: alles von einem Meisterwerk.", output_func=lambda x: x.dict()["texts
↳ "][0]["content"])
print("Translation to Finnish: ", pretty_result)

# You can also set the output_func parameter to "auto" to extract the information_
↳ needed automaticly.
# This is not working for all the services.
service = Service.from_id(5228)
pretty_result = service("Ich habe diesen Film geliebt. Die Schauspieler, das_
↳ Drehbuch: alles von einem Meisterwerk.", output_func="auto")
print("Translation to Finnish: ", pretty_result)
```

__call__ (*request_input: Union[str, List[str], elg.model.base.Request.Request] = None, request_type: str = 'text', sync_mode: bool = False, timeout: int = None, check_file: bool = True, verbose: bool = True, output_func: Union[str, Callable] = <function Service.<lambda>>, **kwargs*) → Union[dict, str]

Method to call a service. You can enter a string input or the path to the file to process. The output is returned in JSON format.

Parameters

- **request_input** (*Union[str, List[str], Request]*) – can be the text to process directly, the name of the file to process, a list of texts, or directly a Request object.
- **request_type** (*str, optional*) – precise the type of the request. Can be “text”, “structuredText”, or “audio”. It is only used if request_input is not a Request object. Defaults to “text”.
- **sync_mode** (*bool, optional*) – True to use the sync_mode. Defaults to False.
- **timeout** (*int, optional*) – number of seconds before timeout. Defaults to None.
- **check_file** (*bool, optional*) – True to check if request_input can be a file or not. Defaults to True.
- **verbose** (*bool, optional*) – False to avoid print messages. Defaults to True.
- **output_func** (*Union[str, Callable], optional*) – function applied to the service response. It can be used to extract only the content from the response. If set to ‘auto’, a generic extractive function will be used. Defaults to lambda response: response.
- **kwargs** – additional keyword arguments used to hide deprecated arguments

Raises

- **ValueError** – if a parameter is not correct.
- **ElgException** – can raise a specific Elg exception if the request to the service did not succeed.

Returns service response in JSON format or as a string if output_func returns a string.

Return type Union[dict, str]

classmethod from_docker_image (*image: str, execution_location: str, port: int = 8000*)

Class method to init a Service class from a Docker image ELG compatible.

Parameters

- **image** (*str*) – name of the Docker image ELG compatible
- **execution_location** (*str*) – url of the ELG compatible endpoint, e.g., *http://localhost:8000/process*.
- **port** (*int*, *optional*) – port to use locally. Defaults to 8000.

Returns Service object with authentication information.

Return type `elg.Service`

```
classmethod from_entity (entity: elg.entity.Entity, auth_object: Optional[str] = None, auth_file: Optional[str] = None, scope: Optional[str] = None, use_cache: bool = True, cache_dir='~/cache/elg')
```

Class method to init a Service class from an Entity object. You can provide authentication information through the `auth_object` or the `auth_file` attributes. If not authentication information is provided, the Authentication object will be initialized.

Parameters

- **entity** (*elg.Entity*) – Entity object to init as a Service.
- **auth_object** (*elg.Authentication*, *optional*) – `elg.Authentication` object to use. Defaults to `None`.
- **auth_file** (*str*, *optional*) – json file that contains the authentication tokens. Defaults to `None`.
- **scope** (*str*, *optional*) – scope to use when requesting tokens. Can be set to “openid” or “offline_access” to get offline tokens. Defaults to “openid”.
- **domain** (*str*, *optional*) – ELG domain you want to use. “live” to use the public ELG, “dev” to use the development ELG and another value to use a local ELG. Defaults to “live”.
- **use_cache** (*bool*, *optional*) – True if you want to use cached files. Defaults to `True`.
- **cache_dir** (*str*, *optional*) – path to the cache_dir. Set it to `None` to not store any cached files. Defaults to “~/cache/elg”.

Returns Service object with authentication information.

Return type `elg.Service`

```
classmethod from_id (id: int, auth_object: Optional[elg.authentication.Authentication] = None, auth_file: Optional[str] = None, scope: Optional[str] = None, domain: Optional[str] = None, use_cache: bool = True, cache_dir: str = '~/cache/elg')
```

Class method to init a Service class from its id. You can provide authentication information through the `auth_object` or the `auth_file` attributes. If not authentication information is provided, the Authentication object will be initialized.

Parameters

- **id** (*int*) – id of the service.
- **auth_object** (*elg.Authentication*, *optional*) – `elg.Authentication` object to use. Defaults to `None`.
- **auth_file** (*str*, *optional*) – json file that contains the authentication tokens. Defaults to `None`.
- **scope** (*str*, *optional*) – scope to use when requesting tokens. Can be set to “openid” or “offline_access” to get offline tokens. Defaults to “openid”.

- **domain** (*str*, *optional*) – ELG domain you want to use. “live” to use the public ELG, “dev” to use the development ELG and another value to use a local ELG. Defaults to “live”.
- **use_cache** (*bool*, *optional*) – True if you want to use cached files. Defaults to True.
- **cache_dir** (*str*, *optional*) – path to the cache_dir. Set it to None to not store any cached files. Defaults to “~/cache/elg”.

Returns Service object with authentication information.

Return type elg.Service

1.48 Benchmark

class elg.benchmark.Benchmark (*services: List[elg.service.Service]*)

Class to execute multiple services in parallel and allows for easy comparison of their outputs using the same input.

Examples:

```
from elg import Benchmark

# A benchmark can be initialized with a list of services ids (from_ids method) or
↳ with a list of entities (from_entities method).
# Here we compare English to German Machine Translation services.
ben = Benchmark.from_ids([610, 624])

# The benchmark can be run on multiple inputs and can be run multiple times to
↳ guarantee the result (the first run is also
# usually longer than the next ones due to the service pods initialization).
result = ben(["Bush is the president of the USA and lives in Washington.", "ELG
↳ is an amazing project."], number_of_runs=2)

# The benchmark call returns a benchmark result object that can be used to
↳ compare the result.

# You can have an overview of the result,
df = result.compare()
print("General comparison:\n", df)

# compare only the results,
df = result.compare_results()
print("Comparison of the results:\n", df)

# or only the response time.
df = result.compare_response_times()
print("Comparison of the response time:\n", df)

# The compare methods return a DataFrame object that can be exported to csv,
↳ excel and many other formats to have a
# better visualization
result.compare().to_csv("/tmp/result.csv")

# We can take another example and compare sentiment analysis services.
```

(continues on next page)

(continued from previous page)

```

ben = Benchmark.from_ids([477, 510])
inputs = [
    "This movie is not good at all.",
    "This movie is not good but it was a good moment at the cinema.",
    "This movie is not so bad.",
    "I liked the movie but it was not must seen.",
    "It was the best movie I have ever seen."
]
result = ben(
    inputs,
    output_funcs=[
        lambda x: x.features["OVERALL"],
        lambda x: x.annotations["SentenceSet"][0].features["score"] * 100
    ]
)

print("Result:\n")
result.compare()

```

__call__ (*request_inputs: Optional[Union[str, List[str], elg.model.base.Request.Request, List[elg.model.base.Request.Request]]] = None, request_type: str = 'text', sync_mode: bool = False, timeout: Optional[int] = None, check_file: bool = True, output_funcs: Union[str, Callable, List[Union[Callable, str]]] = 'auto', number_of_runs: int = 2*)
 Method to run the comparison of the services with the given inputs.

Parameters

- **request_inputs** (*Union[str, List[str], Request, List[Request]], optional*) – list of inputs on which to compare the services. Each input must correspond to the request_input parameter of the Service `__call__` method. Defaults to None.
- **request_type** (*str, optional*) – precise the type of the request. Can be “text”, “structuredText”, or “audio”. It is only used if request_input is not a Request object. Defaults to “text”.
- **sync_mode** (*bool, optional*) – sync_mode parameter to give to the Service `__call__` method. Defaults to False.
- **timeout** (*int, optional*) – timeout parameter to give to the Service `__call__` method. Defaults to None.
- **check_file** (*bool, optional*) – check_file parameter to give to the Service `__call__` method. Defaults to True.
- **output_funcs** (*Union[str, Callable, List[Union[str, Callable]]], optional*) – output_func parameters to give to the Services `__call__` method. Defaults to “auto”.
- **number_of_runs** (*int, optional*) – number of times to run the services on each input. It is recommended to run the services at least 2 times because on the first time the services usually need to be loaded in the ELG cluster, which will increase the response time. The response time of the second pass is, therefore, more precise. Defaults to 2.

Returns

result of the Benchmark call. To obtain the pandas DataFrame containing all the results, run the compare method on the obtained BenchmarkResult object.

Return type *BenchmarkResult*

```
classmethod from_entities (entities: List[elg.entity.Entity], auth_object: Optional[str] =  
None, auth_file: Optional[str] = None, scope: Optional[str] =  
None, use_cache: bool = True, cache_dir: str = '~/.cache/elg')
```

Class method to init a Benchmark using a list of entities which will be convert into services using the *from_entity* class method of the Service class. Refer to this method for further explanation.

```
classmethod from_ids (ids: List[int], auth_object: Optional[elg.authentication.Authentication] =  
None, auth_file: Optional[str] = None, scope: Optional[str] = None, do-  
main: Optional[str] = None, use_cache: bool = True, cache_dir: str =  
'~/.cache/elg')
```

Class method to init a Benchmark using a list of ids which will be convert into services using the *from_id* class method of the Service class. Refer to this method for further explanation.

```
class elg.benchmark.BenchmarkResult (services: List[elg.service.Service], request_inputs:  
List[str])
```

Class the represent the result of a Benchmark call

```
compare (columns: List[str] = ['result', 'response_time'], func: Union[str, list, dict] = 'last', level: str =  
'run', colwidth: int = 0, **agg_kwargs)
```

Method to compare the obtained results. It returns a pandas DataFrame object containing the comparison

Parameters

- **columns** (*List[str], optional*) – columns of the DataFrame to returned. Defaults to ["result", "response_time"].
- **func** (*Union[str, list, dict], optional*) – function to use for the comparison. To see all the possible function, please see <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.core.groupby.DataFrameGroupBy.aggregate.html?>. Defaults to "last".
- **level** (*str, optional*) – level of the comparison. The level value can be: 'service', 'request_input
- **' _**
- **'run'**. Defaults to "run". (or) –
- **colwidth** (*int, optional*) – if set, will change the colwidth parameter of pandas to better vizualize the DataFrame. Defaults to 0.

Raises ValueError – error if the level parameters is not set to a correct value.

Returns pandas DataFrame object containing the comparison

Return type pd.DataFrame

```
compare_response_times (func: Union[str, list, dict] = 'describe', level: str = 'service', colwidth:  
int = 0, **agg_kwargs)
```

Method similar to the compare method with default parameters optimized to compare the response_times

```
compare_results (func: Union[str, list, dict] = 'last', level: str = 'request_input', colwidth: int = 0,  
**agg_kwargs)
```

Method similar to the compare method with default parameters optimized to compare the results

```
set_colwidth (value: Optional[int] = None)
```

Method to easily change the colwidth value of pandas to better vizualize the DataFrame

Parameters value (*int, optional*) – value of the colwidth. Defaults to None.

1.49 Pipeline

class `elg.pipeline.Pipeline` (*services: List[`elg.service.Service`]*)

Class to execute multiple services one after the other. The output of the first service is used as the input of the next one and so on. It is a basic approach but allows users to create complex services.

Examples:

```
from elg import Pipeline

# A pipeline is simply a list of services and therefore can be initialized with a
↳ list of service ids or a list of entities.
german_sentiment_analysis_pipeline = Pipeline.from_ids([607, 510])

# To map the output of a service to the next one, we use the output_funcs
↳ parameter to extract the needed information
# from the service response. In this example, we need to extract only the
↳ translated text to use it as input of the
# sentiment analyser service. The first item of the output_funcs list corresponds
↳ to the callable object that extracts
# the translated text from the first service output, and the second item of the
↳ list corresponds to the function apply
# to the result of the second service. Here it is set the auto which means that
↳ the automatic content extraction will be used.
results = german_sentiment_analysis_pipeline(
    "Ich habe diesen Film geliebt. Die Schauspieler, das Drehbuch: alles von
↳ einem Meisterwerk.",
    output_funcs=[
        lambda x: x.texts[0].content,
        "auto"
    ]
)

# The returned results object contains the result of each service of the pipeline.
print("Result of the translation service: ", results[0])
print("\nResult of the sentiment analyser service: ", results[1])

# With the pipeline class we can create a lot of different tools. For example, let
↳ 's create a pipeline that generate an
# German summary from an English news article. To do that, we will use a
↳ summarization service (478) and an English to
# German Machine Translation service (610).
pipeline = Pipeline.from_ids([478, 610])
TEXT_TO_SUMMARIZE = '''In his new book, Fulfillment, Alec MacGillis writes of an
↳ Amazon distribution center in Sparrows Point, Maryland that sits on land once
↳ occupied by a Bethlehem Steel plant. The story underscores how dramatically the
↳ U.S. economy has transformed in recent years. Instead of making things, many of
↳ our biggest companies now distribute things made elsewhere. We've moved from an
↳ economy of production to one of dispersion. The shift from factory to
↳ fulfillment work is core to the American story right now. For the American
↳ worker, a factory job like one at Bethlehem Steel was dangerous, but it paid
↳ $30 to $40 per hour, and many stuck with it for life. At an Amazon Fulfillment
↳ Center, pay starts at $15 per hour, algorithms monitor your performance, and
↳ many workers leave soon after joining. "There's 100% turnover in the warehouses,
↳ " MacGillis told me this week. "100% \every single year." Some blame the move
↳ to fulfillment work entirely on Amazon, but it didn't happen in a vacuum.
↳ American politicians helped it along by signing trade deals like NAFTA and
↳ enthusiastically welcoming China into the World Trade Organization - and doing
↳ so without sufficient safeguards. American industry then suffered the
↳ consequences, and Amazon reaped the benefits. Listen to Alec MacGillis on
↳ week's Big Technology Podcast on Apple, Spotify, or your app of choice. The U.S.
↳ embrace of globalization flooded the country's markets with inexpensive
↳ products, and plants stateside couldn't keep up. As American factories went
↳ under or moved overseas, there was glaring a need for a company that could get
```

(continues on next page)

(continued from previous page)

```

results = pipeline(
    TEXT_TO_SUMMARIZE,
    output_funcs=[
        lambda x: " ".join([data.features["prefLabel"] for data in x.annotations[
            ↪ "Main sentence"]]),
        lambda x: x.texts[0].content,
    ]
)
print("German summary: ", results[-1])

```

__call__ (*request_input: Optional[Union[str, List[str], elg.model.base.Request.Request]] = None, request_types: Union[str, List[str]] = 'text', sync_mode: bool = False, timeout: Optional[int] = None, check_file: bool = True, verbose: bool = True, output_funcs: Union[str, Callable, List[Union[Callable, str]]] = 'auto')*
 Method to run the pipeline.

Parameters

- **request_input** (*Union[str, List[str], Request], optional*) – input that corresponds to the request_input parameter of the Service **__call__** method. Defaults to None.
- **request_type** (*str, optional*) – precise the type of the request. Can be “text”, “structuredText”, or “audio”. It is only used if request_input is not a Request object. Defaults to “text”.
- **sync_mode** (*bool, optional*) – sync_mode parameter to give to the Service **__call__** method. Defaults to False.
- **timeout** (*int, optional*) – timeout parameter to give to the Service **__call__** method. Defaults to None.
- **check_file** (*bool, optional*) – check_file parameter to give to the Service **__call__** method. Defaults to True.
- **output_funcs** (*Union[str, Callable, List[Union[str, Callable]]], optional*) – output_func parameters to give to the Services **__call__** method. Defaults to “auto”.

Returns service response in JSON format or as a string if output_func returns a string.

Return type Union[Dict, str]

classmethod from_entities (*entities: List[elg.entity.Entity], auth_object: Optional[str] = None, auth_file: Optional[str] = None, scope: Optional[str] = None, use_cache: bool = True, cache_dir: str = '~/.cache/elg'*)

Class method to init a Pipeline using a list of entities which will be convert into services using the *from_entity* class method of the Service class. Refer to this method for further explanation.

classmethod from_ids (*ids: List[int], auth_object: Optional[elg.authentication.Authentication] = None, auth_file: Optional[str] = None, scope: Optional[str] = None, domain: Optional[str] = None, use_cache: bool = True, cache_dir: str = '~/.cache/elg'*)

Class method to init a Pipeline using a list of ids which will be convert into services using the *from_id* class method of the Service class. Refer to this method for further explanation.

1.50 FlaskService

class `elg.FlaskService` (*name: str*)

Class to help the creation of an ELG compatible service from a python model

classmethod `create_docker_files` (*required_files: List = [], required_folders: List = [], path: Optional[str] = None*)

Class method to create the correct Dockerfile.

Parameters

- **required_files** (*List, optional*) – List of files needed for the service. Defaults to [].
- **required_folders** (*List, optional*) – List of folders needed for the service. Defaults to [].
- **path** (*str, optional*) – Path where to generate the file. Defaults to None.

classmethod `create_requirements` (*requirements: List = [], path: Optional[str] = None*)

Class method to create the correct requirements.txt file.

Parameters

- **requirements** (*List, optional*) – List of required pip packages. Defaults to [].
- **path** (*str, optional*) – Path where to generate the file. Defaults to None.

classmethod `docker_build_image` (*tag: str, pull: bool = True, path: Optional[str] = None, **kwargs*)

Class method to do *docker build* ... in python.

classmethod `docker_push_image` (*repository: str, tag: str, username: Optional[str] = None, password: Optional[str] = None, **kwargs*)

Class method to do *docker push* ... in python.

invalid_request_error ()

Generates a valid ELG “failure” response if the request cannot be parsed

process ()

Main request processing logic - accepts a JSON request and returns a JSON response.

process_audio (*request: elg.model.request.AudioRequest.AudioRequest*)

Method to implement if the service takes audio as input.

Parameters **request** (*AudioRequest*) – AudioRequest object.

process_request (*request*)

Method to process the request object. This method only calls the right process method regarding the type of the request.

process_structured_text (*request: elg.model.request.StructuredTextRequest.StructuredTextRequest*)

Method to implement if the service takes structured text as input.

Parameters **request** (*StructuredTextRequest*) – StructuredTextRequest object.

process_text (*request: elg.model.request.TextRequest.TextRequest*)

Method to implement if the service takes text as input.

Parameters **request** (*TextRequest*) – TextRequest object.

run ()

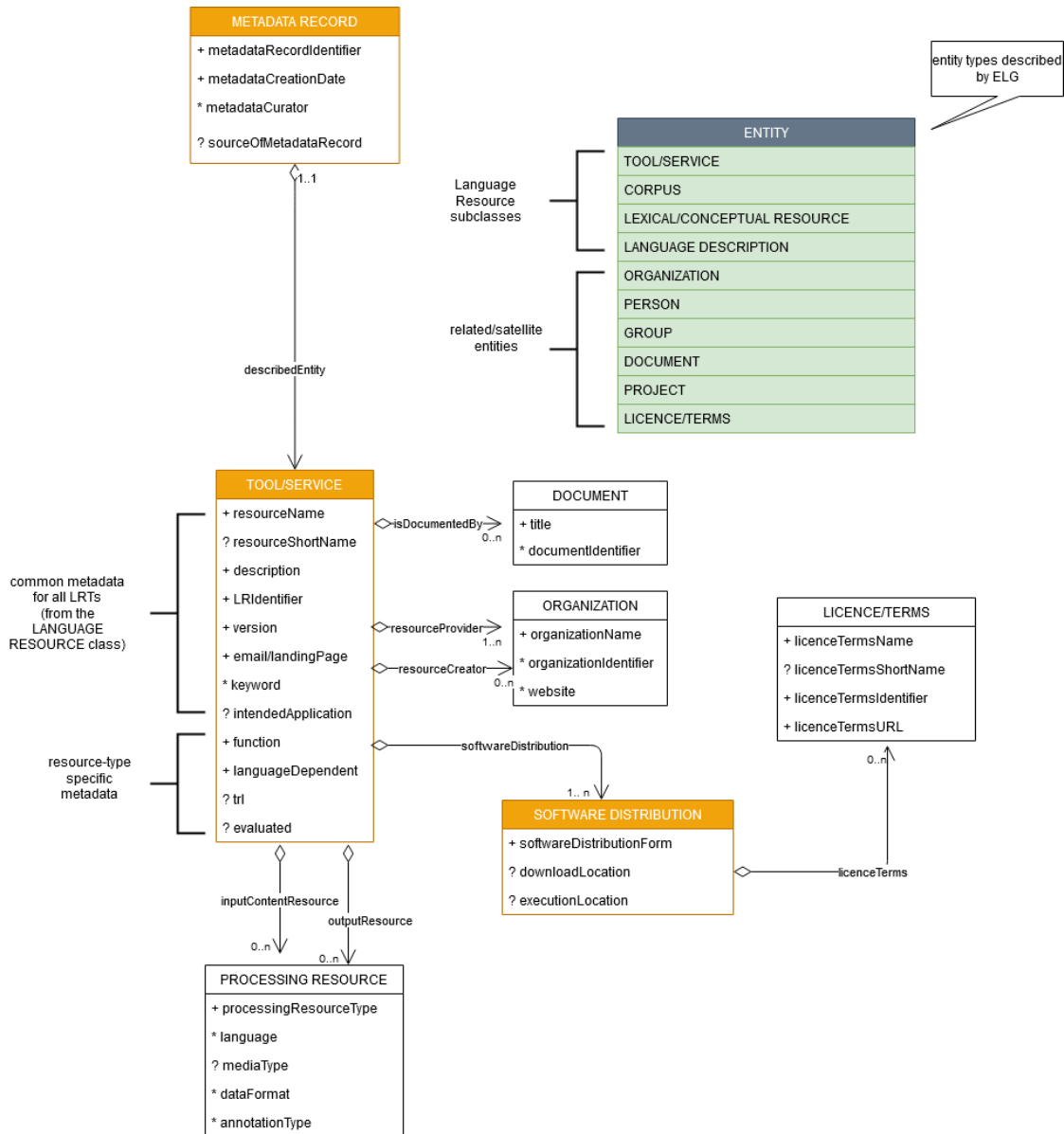
Method to start the flask app.

1.51 Metadata schema

This annex provides an overview of ELG’s metadata schema, ELG-SHARE. We describe the basic concepts, provide links to the full schema documentation, and finally present the “minimal version” of the schema, consisting only of required and recommended elements.

1.51.1 Basic concepts

The following figure shows the main notions upon which the ELG schema builds, using the “tool/service” as the illustrative case.



The main concepts include:

- **MetadataRecord**: It corresponds to the catalogue item, and records information concerning the registration process, such as who created the item and when, whether it was harvested from another catalogue, who is

responsible for its curation (updates), etc.

- **DescribedEntity:** It corresponds to any entity that can be described by a metadata record. It can be a Language Resource, a Person, Organization, etc. (cf. *Types of catalogue items* and the green box in the above image). The `LanguageResource` class is further distinguished into one of four resource types: `ToolService`, `Corpus`, `LexicalConceptualResource` and `LanguageDescription`. A Language Resource can be described through a set of metadata elements common to all types, and a further set that fits to each of these four types.
- **Distribution:** It corresponds to the physical form with which a Language Resource is made available through the catalogue, e.g. as a downloadable file, or a form accessed via an interface, etc.

1.51.2 Full schema documentation

You can find the full schema XSD, documentation as well as templates and examples of metadata records for all resource types in the [ELG SHARE schema Git repository](#).

You can browse the full schema documentation here:

- [Metadata record \(Base item\)](#)
- **Language Resource**
 - [Tool/Service](#)
 - [Corpus](#)
 - [Language description](#)
 - [Lexical/Conceptual resource](#)
- [Project](#)
- [Organization](#)
- [Group](#)
- [Person](#)
- [Licence/terms of use](#)
- [Document](#)

1.52 Minimal version

The minimal version comprises a set of carefully selected metadata elements that are deemed important for various reasons, such as:

- *identification and citation:* resource name(s); identifier(s); a short description of contents; versioning information; a contact point for further information (email or landing page); data of the resource provider(s) and resource creator(s); classification by domain, keywords and intended LT application; language coverage (language and, if needed, dialect); publication date;
- *support:* links to manuals, training material; samples of the resource;
- *usage/access:* distribution form (e.g. as downloadable file, a form that can be accessed via an interface, source code or binary file of software, etc.); licensing conditions; access location.

These metadata elements can be used to describe all resources, irrespective of the resource type. Additional metadata elements, particular to each resource type, are required, such as size and format for data files, dependencies and technical requirements for tools and services, etc.

1.52.1 Outline and explanations for the following sections

The following sections present the minimal schema, grouped as described above, i.e. first for elements common to all LRTs, and then by resource type. Each section includes:

- an overview, with a tabular presentation of the mandatory (M) and recommended (R) elements. More specifically, the table provides information on the element name, the element optionality and the section tab where the user can find each element in the interactive editor. The elements are grouped according to the tab where they are found. The values for optionality are:
 - **Mandatory (M)**: the element must always be filled in the metadata record
 - **Recommended (R)**: the use of the element is not enforced but provides important information
 - **Mandatory if applicable (MA)**: the element must be filled in when specific conditions apply
 - **Recommended if applicable (RA)**: the use of the element is recommended when specific conditions apply
- a detailed presentation for each metadata element with the following information:
- *Path*: the path of the element as in the XSD
- *Data type*:
 - string
 - multilingual string: you can repeat the element for different language versions; to specify the language, you must use the xml attribute `lang` with a value from IETF BCP 47, the [IANA Language Subtag Registry](#); for all metadata elements, a value in English (“en”) is mandatory
 - component: group of elements
 - Controlled Vocabulary (CV): value taken from a controlled vocabulary; a link to the relevant controlled vocabulary is provided
 - date: date in the format `xs:date`
 - URL
- *Optionality*:

For an explanation of the values, see above.

- *Explanation & Instructions*: A short definition of the element, followed by instructions on how it should be used in the specific context.
- *Example*: One or more examples for the element in XML format.

1.52.2 Minimal elements for all entities

This page describes the minimal metadata elements common to all types of entities.

1. Overview

Element name	Optionality	Section	Tab
metadataCreationDate	R		
metadataCurator	R		
compliesWith	R		
metadataCreator	R		
sourceOfMetadataRecord	R	LRT	Identity
		Organization	
		Project	

2. Element presentation

In this section all the aforementioned elements are presented following the order of the elements in the table of the previous section.

MetadataRecord

Path MetadataRecord

Data type component

Optionality Mandatory

Explanation & Instructions

A set of formalized structured information used to describe the contents, structure, function, etc. of an entity, usually according to a specific set of rules (metadata schema)

The `MetadataRecord` element wraps together a set of administrative data, of which the main elements (**automatically assigned by the ELG software**) for metadata records registered by individuals (presented in the previous table) are:

- `metadataCreationDate`: the date when the metadata record was created
- `metadataCurator`: the person that will be assigned the responsibility to update the metadata record when imported in the ELG database; it is usually the same person as the `metadataCreator`
- `compliesWith`: for ELG metadata records, this is by default the ELG-SHARE metadata schema
- `metadataCreator`: the person that has created the metadata record
- `sourceOfMetadataRecord`: used for metadata records that have been imported into ELG from other catalogues, either automatically harvested or through a manual collection procedure; it consists of two mandatory elements, `repositoryName` and `repositoryURL`, and the optional element `repositoryIdentifier`.

All elements apart from the `sourceOfMetadataRecord` are automatically assigned; they are, therefore, not displayed on the interactive editor and they do not have to be added in the metadata file.

The `sourceOfMetadataRecord` is mandatory for harvested records and automatically assigned for them. It is recommended for records registered by individuals and, therefore, displayed in the interactive editor form under the section “Language Resource/Technology”, “Project” or “Organization”.

Example

```

<ms:MetadataRecord>
  <ms:MetadataRecordIdentifier ms:MetadataRecordIdentifierScheme="http://w3id.org/
↪meta-share/meta-share/elg">default id</ms:MetadataRecordIdentifier>
  <ms:metadataCreationDate>2020-02-28</ms:metadataCreationDate>
  <ms:metadataCurator>
    <ms:actorType>Person</ms:actorType>
    <ms:surname xml:lang="en">Smith</ms:surname>
    <ms:givenName xml:lang="en">John</ms:givenName>
  </ms:metadataCurator>
  <ms:compliesWith>http://w3id.org/meta-share/meta-share/ELG-SHARE</ms:compliesWith>
  <ms:metadataCreator>
    <ms:actorType>Person</ms:actorType>
    <ms:surname xml:lang="en">Brown</ms:surname>
    <ms:givenName xml:lang="en">George</ms:givenName>
  </ms:metadataCreator>
  <sourceOfMetadataRecord>
    <repositoryName xml:lang="en">ELRC-SHARE</repositoryName>
    <repositoryURL>https://www.elrc-share.eu/</repositoryName>
  </sourceOfMetadataRecord>
</ms:metadataRecord>

```

1.52.3 Minimal elements for all language resources and technologies

This page describes the minimal metadata elements common to **all language resources and technologies (LRTs)**.

1. Overview

Element name	Optionality	Section	Tab
resourceName	M	LRT	Identity
LRIdentifier	R	LRT	Identity
resourceShortName	R	LRT	Identity
description	M	LRT	Identity
version	M	LRT	Identity
versionDate	R	LRT	Identity
resourceProvider	R	LRT	Identity
resourceCreator	R	LRT	Identity
publicationDate	R	LRT	Identity
fundingProject	R	LRT	Identity
logo	R	LRT	Identity
sourceOfMetadataRecord	R	LRT	Identity
intendedApplication	R	LRT	Categories
domain	R	LRT	Categories
keyword	M	LRT	Categories
additionalInfo	M	LRT	Contact
contact	R	LRT	Contact
isDocumentedBy	R	LRT	Documentation
replaces	R	LRT	Related LRTs
isVersionOf	R	LRT	Related LRTs
isPartOf	R	LRT	Related LRTs
isSimilarTo	R	LRT	Related LRTs
relation	R	LRT	Related LRTs

2. Element presentation

In this section all the aforementioned elements are presented each one separately. The presentation follows the order of the elements in the table of the previous section.

resourceName

Path `MetadataRecord.DescribedEntity.LanguageResource.resourceName`

Data type multilingual string

Optionality Mandatory

Explanation & Instructions

Introduces a human-readable name or title by which the resource is known

This is the “brand name” of your resource; try to use a name that is unique.

Example

```
<ms:resourceName xml:lang="en">GATE: English Named Entity Recognizer</ms:resourceName>
```

LRIdentifier

Path `MetadataRecord.DescribedEntity.LanguageResource.LRIdentifier`

Data type string with attribute

Optionality Recommended when applicable

Explanation & Instructions

A string (e.g., PID, DOI, internal to an organization , etc.) used to uniquely identify a language resource

You must also use the attribute `LRIdentifierScheme` to specify the identifier scheme (e.g., DOI, Handle, ...)

If the resource is already described in another repository/catalogue and has a PID, please add it with the appropriate attribute.

Example

```
<ms:LRIdentifier ms:LRIdentifierScheme="http://w3id.org/meta-share/meta-share/elg">  
→ELG id automatically assigned</ms:LRIdentifier>
```

resourceShortName

Path MetadataRecord.DescribedEntity.LanguageResource.resourceShortName

Data type multilingual string

Optionality Recommended

Explanation & Instructions

Introduces a short form (e.g., abbreviation, acronym , etc.) used to refer to a language resource

Example

```
<ms:resourceShortName xml:lang="en">annie-named-entity-recognizer</ms:resourceShortName>
```

description

Path MetadataRecord.DescribedEntity.LanguageResource.description

Data type multilingual string

Optionality Mandatory

Explanation & Instructions

Introduces a short free-text account that provides information about the resource (e.g., service function, contents of a data resource, technical information , etc.)

Example

```
<ms:description xml:lang="en">Identifies names of persons, locations, organizations, as well as money amounts, time and date expressions in English texts automatically.</ms:description>
```

version

Path MetadataRecord.DescribedEntity.LanguageResource.version

Data type string

Optionality Mandatory

Explanation & Instructions

Associates a language resource with a pattern that indicates its version; the recommended way is to follow the semantic versioning guidelines (<http://semver.org>) and use a numeric pattern of the form major_version.minor_version.patch

If no version is provided, the system will automatically assign the resource a ‘v1.0.0 (automatically assigned)’ value

Example

```
<ms:version>v8.6</ms:version>
```

versionDate

Path MetadataRecord.DescribedEntity.LanguageResource.versionDate

Data type date

Optionality Recommended

Explanation & Instructions

Identifies the date associated with the version of the language resource being described (as a recommendation, of the latest update of the particular version)

Example

```
<ms:versionDate>2020-02-10</ms:versionDate>
```

resourceProvider

Path MetadataRecord.DescribedEntity.LanguageResource.resourceProvider

Data type component

Optionality Recommended

Explanation & Instructions

The person/organization responsible for providing, curating, maintaining and making available (publishing) the resource

The resource provider is very similar to the publisher of scientific articles; it can be an individual or an organization.

For organizations you must add the name of the organization (organizationName) and, if possible, the website.

For persons, you must add the given name and surname and, if possible, an email address or an identifier (such as [ORCID id](#)) to help uniquely identify them.

Example

```
<ms:resourceProvider>
  <ms:Organization>
    <ms:actorType>Organization</ms:actorType>
    <ms:organizationName xml:lang="en">Organization</
↪ms:organizationName>
    <ms:website>https://provider.org/</ms:website>
  </ms:Organization>
</ms:resourceProvider>

<ms:resourceProvider>
  <ms:Person>
    <ms:actorType>Person</ms:actorType>
    <ms:surname xml:lang="en">Smith</ms:surname>
    <ms:givenName xml:lang="en">John</ms:givenName>
  </ms:Person>
</ms:resourceProvider>
```

resourceCreator

Path MetadataRecord.DescribedEntity.LanguageResource.resourceCreator

Data type component

Optionality Recommended

Explanation & Instructions

Links a resource to the person, group or organization that has created the resource

The element is important for citation and acknowledgement purposes.

For organizations, you must add the name of the organization (organizationName) and, if possible, the website.

For persons, you must add the given name and surname and, if possible, an email address or an identifier (such as [ORCID id](#)) to help uniquely identify them.

Example

```
<ms:resourceCreator>
  <ms:Organization>
    <ms:actorType>Organization</ms:actorType>
    <ms:organizationName xml:lang="en">example organization</
    <ms:organizationName>
      <ms:website>https://provider.org/</ms:website>
    </ms:Organization>
  </ms:resourceCreator>

<ms:resourceCreator>
  <ms:Person>
    <ms:actorType>Person</ms:actorType>
    <ms:surname xml:lang="en">Smith</ms:surname>
    <ms:givenName xml:lang="en">John</ms:givenName>
  </ms:Person>
</ms:resourceCreator>
```

publicationDate

Path MetadataRecord.DescribedEntity.LanguageResource.publicationDate

Data type date

Optionality Recommended

Explanation & Instructions

Specifies the date when a language resource has been made available to the public

Publication date is important for citation purposes, just as for scientific articles. If this is the first time your resource is published, please use the same date as for metadataCrationDate. If the resource has been previously published in another repository, please add the date it was first provided there.

Example

```
<ms:publicationDate>2015-12-17</ms:publicationDate>
```

fundingProject

Path MetadataRecord.DescribedEntity.LanguageResource.fundingProject

Data type component

Optionality Recommended when applicable

Explanation & Instructions

Links a language resource to the project that has funded its creation, enrichment, extension , etc.

Funding information is important for acknowledgement purposes.

For projects, you must provide the name of the project (projectName) and, if possible, a website (website) and/or an identifier (ProjectIdentifier). You may also provide a grant number issued by the funding authority (grantNumber), the funder(s) (funder), in the form of organization, person or group, and a value selected from the fundingType controlled vocabulary.

Example

```
<ms:fundingProject>
  <ms:projectName xml:lang="en">European Language Resource Coordination LOT3
↪ </ms:projectName>
  <ms:ProjectIdentifier ms:ProjectIdentifierScheme="http://w3id.org/meta-
↪ share/meta-share/other">SMART 2015/1091 - 30-CE-0816766/00-92</ms:ProjectIdentifier>
  <ms:website>http://www.lr-coordination.eu</ms:website>
  <ms:grantNumber>EU 1234567890</ms:grantNumber>
  <ms:fundingType>http://w3id.org/meta-share/meta-share/serviceContract</
↪ ms:fundingType>
  <ms:fundingType>http://w3id.org/meta-share/meta-share/other</
↪ ms:fundingType>
  <ms:funder>
    <ms:Organization>
      <ms:actorType>Organization</ms:actorType>
      <ms:organizationName xml:lang="en">Ministry of Research_
↪ and Innovation</ms:organizationName>
      <ms:website>http://www.ministry.org</ms:website>
    </ms:Organization>
  </ms:funder>
</ms:fundingProject>
```

logo

Path MetadataRecord.DescribedEntity.LanguageResource.logo

Data type URL

Optionality Recommended

Explanation & Instructions

Links to a URL with an image file containing a symbol or graphic object used to identify the entity

The logo is like a brand name for the resource; it is displayed next to the resource name in the catalogue. In the interactive editor form, you can also upload an image file.

Example

```
<logo>https://gate.ac.uk/plugins/gau-0.1/images/logo-gate.png</logo>
```

sourceOfMetadataRecord

Path MetadataRecord.sourceOfMetadataRecord

Data type component

Optionality Recommended

Explanation & Instructions

Refers to the entity (repository, catalogue, archive, etc.) from which the metadata record has been imported into the new catalogue

This element is a property of the metadata record, and it is automatically assigned by the ELG software for records automatically harvested. For records originally included in other catalogues and registered in ELG by individuals, the element can be filled in at the LRT section of the editor.

It consists of two mandatory elements, repositoryName and repositoryURL, and the optional element repositoryIdentifier.

Example

```
<sourceOfMetadataRecord>
  <repositoryName xml:lang="en">ELRC-SHARE</repositoryName>
  <repositoryURL>https://www.elrc-share.eu/</repositoryName>
</sourceOfMetadataRecord>
```

intendedApplication

Path MetadataRecord.DescribedEntity.LanguageResource.intendedApplication

Data type component

Optionality Recommended

Explanation & Instructions

Specifies an LT application for which the language resource has been created or for which it can be used or is recommended to be used

The element is important for discovery purposes.

You can use the element LTClassRecommended with one of the recommended values from the LT taxonomy (class 'Function' of the OMTD-SHARE ontology at <http://w3id.org/meta-share/omtd-share/>), or add a free text at the LTClassOther element.

You can repeat the element if the resource can be used for various applications. For instance, a part-of-speech tagger can be used as a component for Named entity recognition, for sentiment analysis, etc.

Example

```
<ms:intendedApplication>
  <ms:LTCClassRecommended>http://w3id.org/meta-share/omtd-share/
  ↳NamedEntityRecognition</ms:LTCClassRecommended>
</ms:intendedApplication>

<ms:intendedApplication>
  <ms:LTCClassRecommended>http://w3id.org/meta-share/omtd-share/
  ↳SentimentAnalysis</ms:LTCClassRecommended>
</ms:intendedApplication>

<ms:intendedApplication>
  <ms:LTCClassOther>face recognition</ms:LTCClassRecommended>
</ms:intendedApplication>
```

domain

Path MetadataRecord.DescribedEntity.LanguageResource.domain

Data type component

Optionality Recommended

Explanation & Instructions

Identifies the domain according to which a resource is classified

You must fill in the `CategoryLabel` element with a free text value. If you prefer to add a value from an established controlled vocabulary, you can also use the `DomainIdentifier` (with the attribute `DomainClassificationScheme` with the appropriate value).

Example

```
<ms:domain>
  <ms:categoryLabel xml:lang="en">EDUCATION & COMMUNICATIONS</
  ↳ms:categoryLabel>
  <ms:DomainIdentifier ms:DomainClassificationScheme="http://w3id.org/meta-
  ↳share/meta-share/EUROVOC">32</ms:DomainIdentifier>
</ms:domain>

<ms:domain>
  <ms:categoryLabel xml:lang="en">health</ms:categoryLabel>
</ms:domain>
```

keyword

Path MetadataRecord.DescribedEntity.LanguageResource.keyword

Data type multilingual string

Optionality Mandatory

Explanation & Instructions

Introduces a word or phrase considered important for the description of a language resource, person or organization and thus used to index or classify it

You can repeat the element if you want to add more keywords. Keywords are used for discovery purposes; so, try to use words or phrases that you think users will use to find similar resources to yours.

Example

```
<ms:keyword xml:lang="en">Named entity recognition</ms:keyword>
<ms:keyword xml:lang="en">person</ms:keyword>
<ms:keyword xml:lang="en">location</ms:keyword>
<ms:keyword xml:lang="en">fake news</ms:keyword>
<ms:keyword xml:lang="en">tweets</ms:keyword>
```

additionalInfo

Path MetadataRecord.DescribedEntity.LanguageResource.additionalInfo

Data type component

Optionality Mandatory

Explanation & Instructions

Introduces a point that can be used for further information (e.g. a landing page with a more detailed description of the resource or a general email that can be contacted for further queries)

It's a recommended practice to give at least a landing page (landingPage) or a general email addresss (email); if you want, you can also specify a contact person (see full schema for contactPerson)

Example

```
<ms:additionalInfo>
  <ms:landingPage>https://provider.example.com/product</ms:landingPage>
</ms:additionalInfo>

<ms:additionalInfo>
  <ms:email>product@example.com</ms:email>
</ms:additionalInfo>
```

contact

Path MetadataRecord.DescribedEntity.LanguageResource.contact

Data type component

Optionality Recommended

Explanation & Instructions

Specifies the data of the person/organization/group that can be contacted for information about a language resource

Example

```
<ms:contact>
  <ms:Person>
    <ms:actorType>Person</ms:actorType>
    <ms:surname xml:lang="en">Smith</ms:surname>
```

(continues on next page)

(continued from previous page)

```
<ms:givenName xml:lang="en">John</ms:givenName>
<ms:PersonalIdentifier ms:PersonalIdentifierScheme="http://purl.org/
↪spar/datacite/orcid">String</ms:PersonalIdentifier>
<ms:email>smith@example.com</ms:email>
</ms:Person>
</ms:contact>
```

isDocumentedBy

Path MetadataRecord.DescribedEntity.LanguageResource.document

Data type component

Optionality Recommended

Explanation & Instructions

Links a language resource to a document (e.g., research paper describing its contents or its use in a project, user manual, etc.) or any other form of documentation (e.g., a URL with support information) that is related to the resource

You can use this element to add

- supporting documentation (user manuals, training material, etc.) for the installation and use of your resource
- scientific publications that describe the resource.

If you want, you can use one of the more fine-grained relations to documents (see [full schema](#)).

You can repeat the element if you want to add more documents.

You must fill in the `title` element with the title of the document (or even an entire bibliographic record). When available, it's also recommended to add the `DocumentIdentifier` with the DOI of the document, or any other link to the document; if you do, use the attribute `DocumentIdentifierScheme` to indicate the identifier type.'

Example

```
<ms:isDocumentedBy>
  <ms:title xml:lang="en">Product User Manual</ms:title>
  <ms:DocumentIdentifier ms:DocumentIdentifierScheme="http://purl.org/spar/
↪datacite/url">https://www.company.org/product.pdf</ms:DocumentIdentifier>
</ms:isDocumentedBy>
```

replaces

Path MetadataRecord.DescribedEntity.LanguageResource.replaces

Data type component

Optionality Recommended

Explanation & Instructions

Links two Language Resources: the one being described to another which is an older version and has been replaced

You must provide the `resourceName` of the language resource and, if possible, an `LRIdentifier` that will help uniquely identify it.

Example

```
<ms:replaces>
  <ms:resourceName xml:lang="en">COVID-19 Concept Embeddings</ms:resourceName>
  <ms:LRIIdentifier ms:LRIIdentifierScheme="http://w3id.org/meta-share/meta-share/
↪doi">https://zenodo.org/record/3753531</ms:LRIIdentifier>
</ms:replaces>
```

isVersionOf

Path MetadataRecord.DescribedEntity.LanguageResource.isVersionOf

Data type component

Optionality Recommended

Explanation & Instructions

Links two Language Resources: the one being described to another which is a version (corrected, annotated, enriched, processed, etc.) of it

You must provide the `resourceName` of the language resource and, if possible, an `LRIIdentifier` that will help uniquely identify it.

Example

```
<ms:isVersionOf>
  <ms:resourceName xml:lang="en">COVID-19 Concept Embeddings</ms:resourceName>
  <ms:LRIIdentifier ms:LRIIdentifierScheme="http://w3id.org/meta-share/meta-share/
↪doi">https://zenodo.org/record/3753531</ms:LRIIdentifier>
</ms:isVersionOf>
```

isPartOf

Path MetadataRecord.DescribedEntity.LanguageResource.isPartOf

Data type component

Optionality Recommended

Explanation & Instructions

Links two Language Resources: the one being described to another containing it (e.g., a monolingual corpus which is a part of a bilingual corpus)

You must provide the `resourceName` of the language resource and, if possible, an `LRIIdentifier` that will help uniquely identify it.

Example

```
<ms:isPartOf>
  <ms:resourceName xml:lang="en">Multilingual Example corpus</ms:resourceName>
  <ms:LRIIdentifier ms:LRIIdentifierScheme="http://w3id.org/meta-share/meta-share/
↪doi">https://zenodo.org/record/123456789</ms:LRIIdentifier>
</ms:PartOf>
```

isSimilarTo

Path MetadataRecord.DescribedEntity.LanguageResource.isSimilarTo

Data type component

Optionality Recommended

Explanation & Instructions

Links two Language Resources: the one being described to another that bears resemblances with. Examples are: two resources which have been built with the same theoretical principles; the same resource which comes in different formats, or processed at the same level with different tools.

You must provide the `resourceName` of the language resource and, if possible, an `LRIdentifier` that will help uniquely identify it.

Example

```
<ms:isSimilarTo>
  <ms:resourceName xml:lang="en">Multilingual Example corpus</ms:resourceName>
  <ms:LRIdentifier ms:LRIdentifierScheme="http://w3id.org/meta-share/meta-share/
↪doi">https://zenodo.org/record/123456789</ms:LRIdentifier>
</ms:isSimilarTo>
```

relation

Path MetadataRecord.DescribedEntity.LanguageResource.relation

Data type component

Optionality Recommended

Explanation & Instructions

Links two Language Resources specifying the type of relation as well

You must provide the `relationType` (free text) and for the `relatedLR`, the `resourceName` of the language resource and, if possible, an `LRIdentifier` that will help uniquely identify it.

Example

```
<ms:relation>
  <ms:relationType xml:lang="en">new relation</ms:relationType>
  <ms:relatedLR>
    <ms:resourceName xml:lang="en">COVID-19 Concept Embeddings</
↪ms:resourceName>
    <ms:LRIdentifier ms:LRIdentifierScheme="http://w3id.org/meta-share/
↪meta-share/doi">https://zenodo.org/record/3753531</ms:LRIdentifier>
    </ms:relatedLR>
  </ms:relation>
```

1.52.4 Minimal elements for tools/services

This page describes the minimal metadata elements specific to **tools/services**.

1. Overview

Element name	Optionality	Section	Tab
function	M	Tool/Service	categories
mlFramework1	R	Tool/Service	categories
implementationLanguage	R	Tool/Service	categories
languageDependent	M	Tool/Service	technical
<i>inputContentResource</i>	M	Tool/Service	technical
processingResourceType	M	Tool/Service	technical
language	MA	Tool/Service	technical
mediaType	R	Tool/Service	technical
dataFormat	R	Tool/Service	technical
annotationType	R	Tool/Service	technical
<i>outputResource</i>	R	Tool/Service	technical
processingResourceType	M	Tool/Service	technical
language	MA	Tool/Service	technical
mediaType	R	Tool/Service	technical
dataFormat	R	Tool/Service	technical
annotationType	R	Tool/Service	technical
requiredHardware	R	Tool/Service	technical
evaluated	M	Tool/Service	evaluation
trl	R	Tool/Service	evaluation
<i>SoftwareDistribution</i>	M	distribution	technical
SoftwareDistributionForm	M	distribution	technical
webServiceType	MA	distribution	technical
dockerDownloadLocation	RA	distribution	technical
serviceAdapterDownloadLocation	RA	distribution	technical
downloadLocation	RA	distribution	technical
executionLocation	RA	distribution	technical
accessLocation	RA	distribution	technical
demoLocation	R	distribution	technical
additionalHWRequirements	R	distribution	technical
isDescribedBy	R	distribution	technical
licenceTerms	M	distribution	technical
cost	R	distribution	technical
membershipInstitution	R	distribution	technical

2. Element presentation

In this section all the aforementioned elements are presented each one separately. The presentation follows the order of the elements in the table of the previous section.

function

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.ToolService.function

Data type component

Optionality Mandatory

Explanation & Instructions

Specifies the operation/function/task that a software object performs

The element is important for discovery purposes.

You can fill in:

- the `LTClassRecommended` element with one of the recommended values from the [LT taxonomy](#), or
- the `LTClassOther` element with a free text.

For services that perform multiple functions (e.g., syntactic and semantic annotation) you can repeat the element.

Example

```
<ms:function>
  <ms:LTClassRecommended>http://w3id.org/meta-share/omtd-share/
  ↪NamedEntityRecognition</ms:LTClassRecommended>
</ms:function>

<ms:function>
  <ms:LTClassRecommended>http://w3id.org/meta-share/omtd-share/
  ↪MachineTranslation</ms:LTClassRecommended>
</ms:function>

<ms:function>
  <ms:LTClassOther>video segmentation</ms:LTClassRecommended>
</ms:function>
```

mlFramework1

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.ToolService.mlFramework1

Data type CV

Optionality Recommended

Explanation & Instructions

Specifies the framework that has been used for developing a model (e.g. keras, tensorflow, etc.)

Example

```
<ms:mlFramework1>http://w3id.org/meta-share/meta-share/TensorFlow</ms:mlFramework1>
```

implementationLanguage

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.ToolService.implementationLanguage

Data type string

Optionality Recommended

Explanation & Instructions

The programming language(s) used for the development of a tool/service, which is needed for running the tools/services, in case no executables are available

Example

```
<ms:implementationLanguage>Java v8</ms:implementationLanguage>
```

languageDependent

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.ToolService.languageDependent

Data type boolean

Optionality Mandatory

Explanation & Instructions

Indicates whether the operation of the tool or service is language dependent or not

For language-dependent tools/services, you will be asked to also provide the language of the input and output resources.

Example

```
<ms:languageDependent>true</ms:languageDependent>
```

inputContentResource

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.ToolService.inputContentResource

Data type component

Optionality Mandatory

Explanation & Instructions

Specifies the requirements set by a tool/service for the (content) resource that it processes

The following elements are mandatory or recommended:

- `processingResourceType` (Mandatory): Specifies the resource type that a tool/service takes as input or produces as output; you must specify, for instance, if the tool/service can process a single file, or set of files, or processes a string typed in by the users.
- `language` (Mandatory if applicable): Specifies the language that is used in the resource or supported by the tool/service, expressed according to the BCP47 recommendation. See [language](#)
- `mediaType` (Recommended): Specifies the media type of the input/output of a language processing tool/service. For ELG functional services, this will be used to fit the appropriate GUI (e.g. “audio” for ASR applications, vs. “text” for Machine Translation applications)
- `dataFormat` (Recommended): Indicates the format(s) of a data resource Please, use to indicate the data format of the resource supported by the tool/service. The [dataFormat](#) controlled vocabulary lists data formats, with their mimetype and documentation on the particularities, thus catering for variations of formats, e.g. GATE XML, TEI variants, etc.
- `characterEncoding` (Recommended if applicable): Specifies the character encoding used for the input/output text resource of an LT service
- `annotationType` (Recommended if applicable): Specifies the annotation type of the annotated version(s) of a resource or the annotation type a tool/ service requires or produces as an output. Use this element only if the tool/service processes pre-annotated corpora; for tools/services processing raw files, do not use. The element takes a value from a controlled vocabulary, see [annotationType](#).

Example

```
<!-- example for a tool with textual input -->
<ms:inputContentResource>
  <ms:processingResourceType>http://w3id.org/meta-share/meta-share/file1</
↪ms:processingResourceType>
  <ms:language>
    <ms:languageTag>en</ms:languageTag> <ms:languageId>en</ms:languageId>
  </ms:language>
  <ms:mediaType>http://w3id.org/meta-share/meta-share/text</ms:mediaType>
  <ms:dataFormat>http://w3id.org/meta-share/omtd-share/Json</ms:dataFormat>
  <ms:characterEncoding>http://w3id.org/meta-share/meta-share/UTF-8</
↪ms:characterEncoding>
</ms:inputContentResource>

<!-- example for an Automatic Speech Recognizer -->
<ms:inputContentResource>
  <ms:processingResourceType>http://w3id.org/meta-share/meta-share/file1</
↪ms:processingResourceType>
  <ms:language>
    <ms:languageTag>de</ms:languageTag> <ms:languageId>de</ms:languageId>
  </ms:language>
  <ms:mediaType>http://w3id.org/meta-share/meta-share/audio</ms:mediaType>
  <ms:dataFormat>http://w3id.org/meta-share/omtd-share/mp3</ms:dataFormat>
  <ms:dataFormat>http://w3id.org/meta-share/omtd-share/wav</ms:dataFormat>
</ms:inputContentResource>
```

outputResource

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.ToolService.outputResource

Data type component

Optionality Recommended if applicable

Explanation & Instructions

Describes the features of the output resource processed by a tool/service.

The set of elements are the same as for the *inputContentResource*.

Make sure that you add here what is relevant for your application. For instance,

- for annotation and information extraction tools/services, use the *annotationType* to indicate the results of your processing; you can repeat it to indicate multiple annotation types (e.g., part of speech, person, amount, location, etc.)
- for Machine Translation tools, indicate the input and output languages respectively.

Example

```
<!-- example for an Information Extraction tool -->
<ms:outputResource>
  <ms:processingResourceType>http://w3id.org/meta-share/meta-share/file1</
↪ms:processingResourceType>
  <ms:language>
    <ms:languageTag>en</ms:languageTag>
    <ms:languageId>en</ms:languageId>
  </ms:language>
  <ms:mediaType>http://w3id.org/meta-share/meta-share/text</ms:mediaType>
  <ms:dataFormat>http://w3id.org/meta-share/omtd-share/Json</ms:dataFormat>
  <ms:characterEncoding>http://w3id.org/meta-share/meta-share/UTF-8</
↪ms:characterEncoding>
  <ms:annotationType>http://w3id.org/meta-share/omtd-share/Person</
↪ms:annotationType>
  <ms:annotationType>http://w3id.org/meta-share/omtd-share/Location</
↪ms:annotationType>
  <ms:annotationType>http://w3id.org/meta-share/omtd-share/Organization</
↪ms:annotationType>
  <ms:annotationType>http://w3id.org/meta-share/omtd-share/Date</
↪ms:annotationType>
</ms:outputResource>

<!-- example for a Machine Translation tool -->
<ms:outputResource>
  <ms:processingResourceType>http://w3id.org/meta-share/meta-share/file1</
↪ms:processingResourceType>
  <ms:language>
    <ms:languageTag>en</ms:languageTag>
    <ms:languageId>en</ms:languageId>
  </ms:language>
  <ms:mediaType>http://w3id.org/meta-share/meta-share/text</ms:mediaType>
  <ms:dataFormat>http://w3id.org/meta-share/omtd-share/Json</ms:dataFormat>
  <ms:characterEncoding>http://w3id.org/meta-share/meta-share/UTF-8</
↪ms:characterEncoding>
</ms:outputResource>
```

language

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.ToolService.
language

Data type component

Optionality Mandatory if applicable

Explanation & Instructions

Specifies the language that is used in the resource or supported by the tool/service, expressed according to the BCP47 recommendation

The element languageTag is composed of the languageId, and optionally scriptId, regionId and variantId; you can use those elements that best describe the language(s) of your resource.

Example

```
<ms:language>
  <ms:languageTag>en</ms:languageTag>
  <ms:languageId>en</ms:languageId>
</ms:language>

<ms:language>
  <ms:languageTag>en-US</ms:languageTag>
  <ms:languageId>en</ms:languageId>
  <ms:regionId>US</ms:regionId>
</ms:language>
```

requiredHardware

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.ToolService.
requiredHardware

Data type CV (requiredHardware)

Optionality Recommended

Explanation & Instructions

Specifies the type of hardware required for running a tool and/or computational grammar

Example

```
<ms:requiredHardware>http://w3id.org/meta-share/meta-share/ocrSystem</
↪ms:requiredHardware>
```


trl

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.ToolService.trl

Data type CV (TRL)

Optionality Recommended

Explanation & Instructions

Specifies the TRL (Technology Readiness Level) of the technology according to the measurement system defined by the EC (https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf)

Example

```
<ms:trl>http://w3id.org/meta-share/meta-share/trl4</ms:trl>
```

evaluated

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.ToolService.evaluated

Data type boolean

Optionality Mandatory

Explanation & Instructions

Indicates whether the tool or service has been evaluated

If the tool/service has been evaluated, you can use the ‘evaluation’ component to give more detailed information; see [here](#) for the relevant elements.

Example

```
<ms:evaluated>false</ms:evaluated>
```

SoftwareDistribution

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.ToolService.SoftwareDistribution

Data type component

Optionality Mandatory

Explanation & Instructions

Any form with which software is distributed (e.g., web services, executable or code files, etc.)

This element groups together information that pertains to the physical form of a tool/service that is made available through the catalogue. For software that is distributed with multiple forms (e.g., as source code, as a web service, etc.), you can repeat this group of elements. The access location and the licensing conditions may differ for each distribution.

The following list includes the mandatory and recommended elements:

- `SoftwareDistributionForm` (Mandatory): The medium, delivery channel or form (e.g., source code, API, web service, etc.) through which a software object is distributed. Use the value `http://w3id.org/meta-share/meta-share/dockerImage` for ELG integrated services.
- `webServiceType` (Recommended if applicable): The type of a web service following the web service communication protocols. Recommended for web services.
- `dockerDownloadLocation` (Mandatory if applicable): A location where the the LT tool docker image is stored. For ELG integrated services, add the location from where the ELG team can download the docker image in order to test it.
- `serviceAdapterDownloadLocation` (Mandatory if applicable): he URL where the docker image of the service adapter can be downloaded from. Required only for ELG integrated services implemented with an adapter.
- `executionLocation` (Mandatory if applicable): A URL where the resource (mainly software) can be directly executed. Add here the REST endpoint at which the LT tool is exposed within the Docker image. It is also used for software available in the form of executable code or web services.
- `downloadLocation` (Mandatory if applicable): A URL where a tool can be downloaded from. To be used only for direct links, i.e. for links that require no extra actions on the part of the user.
- `accessLocation` (Mandatory if applicable): A URL where a tool can be accessed. It can be used, for instance, for links to tools that are included in a web page, or for tools that require authentication and authorization before being accessed.
- `demoLocation` (Recommended if applicable): A URL providing access to a demo version of the tool/service. For ELG integrated services, this does not have to be filled in, since ELG provides a demo version at the “Try out” tab of the metadata record.
- `additionalHwRequirements` (Mandatory if applicable): A short text where you specify additional requirements for running the service, e.g. memory requirements, etc. The recommended format for this is: ‘limits_memory: X limits_cpu: Y’
- `licenceTerms` (Mandatory): See [licenceTerms](#)
- `cost` (Recommended if applicable): The cost for accessing a resource or the overall budget of a project, formally described as a set of amount (*amount*) and currency unit (*currency*). Fill in this element only if the tool/service can be accessed on a fee.
- `membershipInstitution` (Recommended if applicable): Introduces an institution with members that can benefit from specific conditions on the use of a resource (e.g. discount, unlimited access, etc.). Use this element only if such specific conditions apply.

Example

```
<ms:SoftwareDistribution>
  <ms:SoftwareDistributionForm>http://w3id.org/meta-share/meta-share/dockerImage
</ms:SoftwareDistributionForm>
  <ms:executionLocation>http://localhost:8080/mt/process/</ms:executionLocation>
  <ms:dockerDownloadLocation>registry.gitlab.com/EXAMPLE</
<ms:dockerDownloadLocation>
  <ms:serviceAdapterDownloadLocation>registry.gitlab.com/serviceAdapter</
<ms:serviceAdapterDownloadLocation>
  <ms:isDescribedBy>
    <ms:title xml:lang="en">description article</ms:title>
    <ms:DocumentIdentifier ms:DocumentIdentifierScheme="http://purl.org/
<ms:DocumentIdentifier>String</ms:DocumentIdentifier>
  </ms:isDescribedBy>
  <ms:additionalHwRequirements>terabytes</ms:additionalHwRequirements>
  <ms:licenceTerms>
```

(continues on next page)

(continued from previous page)

```

        <ms:licenceTermsName xml:lang="en">GNU Lesser General Public License v
↪v3.0 only</ms:licenceTermsName>
        <ms:licenceTermsURL>https://spdx.org/licenses/LGPL-3.0-only.html</
↪ms:licenceTermsURL>
        <ms:LicenceIdentifier ms:LicenceIdentifierScheme="http://w3id.org/
↪meta-share/meta-share/SPDX">LGPL-3.0-only</ms:LicenceIdentifier>
        <ms:conditionOfUse>http://w3id.org/meta-share/meta-share/unspecified</
↪ms:conditionOfUse>
    </ms:licenceTerms>
    <ms:cost>
        <ms:amount>14500</ms:amount>
        <ms:currency>http://w3id.org/meta-share/meta-share/euro</ms:currency>
    </ms:cost>
    <ms:membershipInstitution>http://w3id.org/meta-share/meta-share/ELRA</
↪ms:membershipInstitution>
</ms:SoftwareDistribution>

```

licenceTerms

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.ToolService.
SoftwareDistribution.licenceTerms

Data type component

Optionality Mandatory

Explanation & Instructions

Links the distribution (distributable form) of a language resource to the licence or terms of use/service (a specific legal document) with which it is distributed

The recommended practice is to add a licence name and identifier from the SPDX list of licences (<https://spdx.org/licenses/>). For proprietary licences or licences not included in the above list, please add a (unique) licence name and the URL where the text of the licence can be found.

You must also fill in the *conditionOfUse* element. For popular standard licences, we have already included the conditions of use. So, you can add the element with the value <http://w3id.org/meta-share/meta-share/unspecified>. For proprietary licences, you can add the conditions of user or use the same value.

Example

```

<ms:licenceTerms>
    <ms:licenceTermsName xml:lang="en">GNU Lesser General Public License v3.0 only
↪</ms:licenceTermsName>
    <ms:licenceTermsURL>https://spdx.org/licenses/LGPL-3.0-only.html</
↪ms:licenceTermsURL>
    <ms:LicenceIdentifier ms:LicenceIdentifierScheme="http://w3id.org/meta-share/
↪meta-share/SPDX">LGPL-3.0-only</ms:LicenceIdentifier>
    <ms:conditionOfUse>http://w3id.org/meta-share/meta-share/unspecified</
↪ms:conditionOfUse>
</ms:licenceTerms>

<ms:licenceTerms>
    <ms:licenceTermsName xml:lang="en">publicDomain</ms:licenceTermsName>
    <ms:licenceTermsURL>https://elrc-share.eu/terms/publicDomain.html</
↪ms:licenceTermsURL>

```

(continues on next page)

(continued from previous page)

```

    <ms:conditionOfUse>http://w3id.org/meta-share/meta-share/noConditions</
↪ms:conditionOfUse>
</ms:licenceTerms>

<ms:licenceTerms>
    <ms:licenceTermsName xml:lang="en">Creative Commons Attribution 4.0_
↪International</ms:licenceTermsName>
    <ms:licenceTermsURL>https://creativecommons.org/licenses/by/4.0/legalcode</
↪ms:licenceTermsURL>
    <ms:LicenceIdentifier ms:LicenceIdentifierScheme="http://w3id.org/meta-share/
↪meta-share/SPDX">CC-BY-4.0</ms:LicenceIdentifier>
    <ms:conditionOfUse>http://w3id.org/meta-share/meta-share/attribution</
↪ms:conditionOfUse>
</ms:licenceTerms>

```

1.52.5 Minimal elements for corpora

This page describes the minimal metadata elements specific to **corpora**.

1. Overview

Corpora are collections of text documents, audio transcripts, audio and video recordings, etc. To cater for the representation of multimedia/multimodal language resources (e.g. a corpus of videos and their subtitles, or corpus of audio recordings and their transcripts), the notion of “**media part**” is introduced in the model. Thus, a corpus consists of *at least one* text, audio, video, image and numerical text parts. Depending on the media part type, the DatasetDistribution component includes a set of text, audio, video, etc. **distribution features**.

The first table below has all the elements (mandatory and recommended) for a Corpus. The second table presents the mandatory and recommended elements for each media part. The third table presents the mandatory and recommended elements for the Distribution component, which includes elements that are specific to each media part.

Table 1 - Corpus common

Element name	Optionality	Section	Tab
corpusSubclass	M	Corpus	Technical
personalDataIncluded	M	Corpus	Technical
personalDataDetails	RA	Corpus	Technical
sensitiveDataIncluded	RA	Corpus	Technical
sensitiveDataDetails	M	Corpus	Technical
anonymized	MA	Corpus	Technical
anonymizationDetails	RA	Corpus	Technical
isAnnotatedVersionOf	R	Corpus	Technical

Table 2 - Media parts

Element name	Optionality	Section	Tab
lingualityType	M	Corpus	text part
multilingualityType	MA	Corpus	text part
multilingualityTypeDetails	R	Corpus	text part
language	M	Corpus	text part
textType	R	Corpus	text part

continues on next page

Table 2 – continued from previous page

Element name	Optionality	Section	Tab
annotation	RA	Corpus	text part
lingualityType	M	Corpus	audio part
multilingualityType	MA	Corpus	audio part
multilingualityTypeDetails	RA	Corpus	audio part
language	M	Corpus	audio part
AudioGenre	R	Corpus	audio part
SpeechGenre	R	Corpus	audio part
numberOfParticipants	R	Corpus	audio part
dialectAccentOfParticipants	R	Corpus	audio part
geographicDistributionOfParticipants	R	Corpus	audio part
annotation	RA	Corpus	audio part
lingualityType	M	Corpus	video part
multilingualityType	MA	Corpus	video part
multilingualityTypeDetails	RA	Corpus	video part
language	M	Corpus	video part
typeOfVideoContent	M	Corpus	video part
VideoGenre	R	Corpus	video part
numberOfParticipants	R	Corpus	video part
dialectAccentOfParticipants	R	Corpus	video part
geographicDistributionOfParticipants	R	Corpus	video part
annotation	RA	Corpus	video part
lingualityType	M	Corpus	image part
multilingualityType	RA	Corpus	image part
multilingualityTypeDetails	RA	Corpus	image part
language	M	Corpus	image part
typeOfImageContent	M	Corpus	image part
ImageGenre	R	Corpus	image part
annotation	RA	Corpus	image part
typeOfTextNumericalContent	M	Corpus	numerical text part
numberOfParticipants	R	Corpus	numerical text part
dialectAccentOfParticipants	R	Corpus	numerical text part
geographicDistributionOfParticipants	R	Corpus	numerical text part
annotation	RA	Corpus	numerical text part

Table 3 - Distribution

Element name	Optionality	Section	Tab
DatasetDistribution	M	Distribution	Technical
DatasetDistributionForm	M	Distribution	Technical
downloadLocation	MA	Distribution	Technical
accessLocation	MA	Distribution	Technical
distributionLocation	MA	Distribution	Technical
samplesLocation	R	Distribution	Technical
distributionTextFeature	MA	Distribution	Technical
distributionAudioFeature	MA	Distribution	Technical
distributionVideoFeature	MA	Distribution	Technical
distributionImageFeature	MA	Distribution	Technical
distributionTextNumericalFeature	MA	Distribution	Technical
licenceTerms	M	Distribution	Technical
cost	R	Distribution	Technical
membershipInstitution	R	Distribution	Technical

2. Element presentation

In this section all the aforementioned elements are presented each one separately. The presentation follows the order of the elements in the tables of the previous section.

Corpus

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corpus

Data type component

Optionality Mandatory

Explanation & Instructions

Wraps together the set of elements that is specific to corpora

Example

```
<ms:LRSubclass>
  <ms:Corpus>
    <ms:lrType>Corpus</ms:lrType>
  </ms:Corpus>
</ms:LRSubclass>
```

corpusSubclass

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.corpusSubclass

Data type CV (corpusSubclass)

Optionality Mandatory

Explanation & Instructions

Introduces a classification of corpora into types (used for descriptive reasons)

Use one of the values for raw corpora, annotated corpora (mixed raw with annotations), annotations (only annotations without the original corpus)

Example

```
<ms:corpusSubclass>http://w3id.org/meta-share/meta-share/rawCorpus</ms:corpusSubclass>

<ms:corpusSubclass>http://w3id.org/meta-share/meta-share/annotatedCorpus</
↪ms:corpusSubclass>
```

personalDataIncluded

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.personalDataIncluded

Data type boolean

Optionality Mandatory

Explanation & Instructions

Specifies whether the language resource contains personal data (mainly in the sense falling under the GDPR)

If the resource contains personal data, you can use the (recommended) `personalDataDetails` to provide more information

Example

```
<ms:personalDataIncluded>true</ms:personalDataIncluded>
<ms:personalDataDetails>The corpus contains data on the place of living and place of
↪birth of participants</ms:personalDataDetails>
```

sensitiveDataIncluded

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.sensitiveDataIncluded

Data type boolean

Optionality Mandatory

Explanation & Instructions

Specifies whether the language resource contains sensitive data (e.g., medical/health-related, etc.) and thus requires special handling

If the resource contains sensitive data, you can use the (recommended) `sensitiveDataDetails` to provide more information.

Example

```
<ms:sensitiveDataIncluded>true</ms:sensitiveDataIncluded>
<ms:sensitiveDataDetails>The corpus contains medical data for persons with_
↳disabilities</ms:sensitiveDataDetails>
```

anonymized

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corpus.anonymized

Data type boolean

Optionality Mandatory if applicable

Explanation & Instructions

Indicates whether the language resource has been anonymized

The element is mandatory if either `personalDataIncluded` or `sensitiveDataIncluded` have 'true' as value; `anonymizationDetails` must also be filled in with information on the anonymization method, etc.

Example

```
<ms:anonymized>true</ms:anonymized>
<ms:anonymizationDetails>pseudonymization performed manually</ms:anonymizationDetails>
```

isAnnotatedVersionOf

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corpus.isAnnotatedVersionOf

Data type component

Optionality Recommended when applicable

Explanation & Instructions

Links to a corpus B which is the raw corpus that has been annotated (corpus A, the one being described)

You must provide the `resourceName` of the language resource and, if possible, an `LRIdentifier` that will help uniquely identify it.

Example

```
<ms:isAnnotatedVersionOf>
  <ms:resourceName xml:lang="en">MTP Annotated German corpus - untagged version
  ↳</ms:resourceName>
  <ms:LRIdentifier ms:LRIdentifierScheme="http://w3id.org/meta-share/meta-share/
  ↳islrn">417-827-623-669-9</ms:LRIdentifier>
</ms:isAnnotatedVersionOf>
```


CorpusTextPart

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corpora.CorporaMediaPart.CorporaTextPart

Data type component

Optionality Mandatory if applicable

Explanation & Instructions

The part of a corpus (or a whole corpus) that consists of textual segments (e.g., a corpus of publications, or transcriptions of an oral corpus, or subtitles , etc.)

You can repeat the group of elements for multiple textual parts.

The mandatory or recommended elements for the text part are:

- **mediaType** (Mandatory): Specifies the media type of a language resource (the physical medium of the contents representation). For text parts, always use the value ‘text’.
- **lingualityType** (Mandatory): Indicates whether the resource includes one, two or more languages.
- **multilingualityType** (Mandatory if applicable): Indicates whether the resource (part) is parallel, comparable or mixed. If **lingualityType** = bilingual or multilingual, it is required; select one of the values for parallel (e.g., original text and its translations), comparable (e.g. corpus of the same domain in multiple languages) and multilingualSingleText (for corpora that consist of segments including text in two or more languages (e.g., the transcription of a European Parliament session with MPs speaking in their native language).
- **language** (Mandatory): Specifies the language that is used in the resource part , expressed according to the BCP47 recommendation. See [language](#).
- **languageVariety** (Mandatory if applicable): Relates a language resource that contains segments in a language variety (e.g., dialect, jargon) to it. Please use for dialect corpora.
- **modalityType** (Recommended if applicable): Specifies the type of the modality represented in the resource. For instance, you can use ‘spoken language’ to describe transcribed speech corpora.
- **TextGenre** (Recommended): A category of text characterized by a particular style, form, or content according to a specific classification scheme. See [TextGenre](#).
- **annotation** (Mandatory if applicable): A set of features describing the annotated parts of a resource. See [annotation](#).

Example

```
<ms:CorpusTextPart>
  <ms:corpusMediaType>CorpusTextPart</ms:corpusMediaType>
  <ms:mediaType>http://w3id.org/meta-share/meta-share/text</ms:mediaType>
  <ms:lingualityType>http://w3id.org/meta-share/meta-share/monolingual</
  <ms:lingualityType>
    <ms:language>
      <ms:languageTag>es</ms:languageTag>
      <ms:languageId>es</ms:languageId>
    </ms:language>
  </ms:CorpusTextPart>
<ms:CorpusTextPart>
```

(continues on next page)

(continued from previous page)

```

<ms:corpusMediaType>CorpusTextPart</ms:corpusMediaType>
<ms:mediaType>http://w3id.org/meta-share/meta-share/text</ms:mediaType>
<ms:lingualityType>http://w3id.org/meta-share/meta-share/bilingual</
↪ms:lingualityType>
  <ms:language>
    <ms:languageTag>es</ms:languageTag>
    <ms:languageId>es</ms:languageId>
  </ms:language>
  <ms:language>
    <ms:languageTag>en</ms:languageTag>
    <ms:languageId>en</ms:languageId>
  </ms:language>
  <ms:multilingualityType>http://w3id.org/meta-share/meta-share/parallel</
↪ms:multilingualityType>
  <ms:TextGenre>
    <ms:CategoryLabel>administrative texts</ms:CategoryLabel>
  </ms:TextGenre>
</ms:CorpusTextPart>

<ms:CorpusTextPart>
  <ms:corpusMediaType>CorpusTextPart</ms:corpusMediaType>
  <ms:mediaType>http://w3id.org/meta-share/meta-share/text</ms:mediaType>
  <ms:lingualityType>http://w3id.org/meta-share/meta-share/monolingual</
↪ms:lingualityType>
  <ms:language>
    <ms:languageTag>en</ms:languageTag>
    <ms:languageId>en</ms:languageId>
  </ms:language>
  <ms:modalityType>http://w3id.org/meta-share/meta-share/spokenLanguage</
↪ms:modalityType>
</ms:CorpusTextPart>

```

CorpusAudioPart

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.CorporusMediaPart.CorporusAudioPart

Data type component

Optionality Mandatory if applicable

Explanation & Instructions

The part of a corpus (or whole corpus) that consists of audio segments

You can repeat the group of elements for multiple audio parts.

The mandatory or recommended elements for the audio part are:

- **mediaType** (Mandatory): Specifies the media type of a language resource (the physical medium of the contents representation). For audio parts, always use the value ‘audio’
- **lingualityType** (Mandatory): Indicates whether the resource includes one, two or more languages
- **multilingualityType** (Mandatory if applicable): Indicates whether the resource (part) is parallel, comparable or mixed. If **lingualityType** = bilingual or multilingual, it is required; select one of the values for parallel (e.g., original text and its translations), comparable (e.g. corpus of the same domain in multiple languages) and

multilingualSingleText (for corpora that consist of segments with content in two or more languages (e.g., the transcription of a European Parliament session with MPs speaking in their native language))

- `language` (Mandatory): Specifies the language that is used in the resource part, expressed according to the BCP47 recommendation. See [language](#)
- `languageVariety` (Mandatory if applicable): Relates a language resource that contains segments in a language variety (e.g., dialect, jargon) to it. Please use for dialect corpora.
- `modalityType` (Recommended if applicable): Specifies the type of the modality represented in the resource. For instance, you can use ‘spoken language’ to describe transcribed speech corpora.
- `AudioGenre` (Recommended if applicable): A category of audio characterized by a particular style, form, or content according to a specific classification scheme. See [AudioGenre](#)
- `SpeechGenre` (Recommended if applicable): A category for the conventionalized discourse of the speech part of a language resource, based on extra-linguistic and internal linguistic criteria. See [SpeechGenre](#)
- `annotation` (Mandatory if applicable): A set of features describing the annotated parts of a resource. See [annotation](#).

Example

```
<ms:CorpusAudioPart>
  <ms:corpusMediaType>CorpusAudioPart</ms:corpusMediaType>
  <ms:mediaType>http://w3id.org/meta-share/meta-share/audio</ms:mediaType>
  <ms:lingualityType>http://w3id.org/meta-share/meta-share/monolingual</
↪ms:lingualityType>
  <ms:language>
    <ms:languageTag>en</ms:languageTag>
    <ms:languageId>en</ms:languageId>
  </ms:language>
  <ms:AudioGenre>
    <ms:CategoryLabel>conference noises</ms:CategoryLabel>
  </ms:AudioGenre>
</ms:CorpusAudioPart>

<ms:CorpusAudioPart>
  <ms:corpusMediaType>CorpusAudioPart</ms:corpusMediaType>
  <ms:mediaType>http://w3id.org/meta-share/meta-share/audio</ms:mediaType>
  <ms:lingualityType>http://w3id.org/meta-share/meta-share/monolingual</
↪ms:lingualityType>
  <ms:language>
    <ms:languageTag>en</ms:languageTag>
    <ms:languageId>en</ms:languageId>
  </ms:language>
  <ms:modalityType>http://w3id.org/meta-share/meta-share/spokenLanguage</
↪ms:modalityType>
  <ms:SpeechGenre>
    <ms:CategoryLabel>monologue</ms:CategoryLabel>
  </ms:SpeechGenre>
</ms:CorpusAudioPart>
```

CorpusVideoPart

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.CorporusMediaPart.CorporusVideoPart

Data type component

Optionality Mandatory if applicable

Explanation & Instructions

The part of a corpus (or a whole corpus) that consists of video segments (e.g., a corpus of video lectures, a part of a corpus with news, a sign language corpus, etc.)

You can repeat the group of elements for multiple video parts.

The mandatory or recommended elements for the video part are:

- `mediaType` (Mandatory): Specifies the media type of a language resource (the physical medium of the contents representation). For video parts, always use the value ‘video’.
- `lingualityType` (Mandatory): Indicates whether the resource includes one, two or more languages.
- `multilingualityType` (Mandatory if applicable): Indicates whether the resource (part) is parallel, comparable or mixed. If `lingualityType` = bilingual or multilingual, it is required; select one of the values for parallel (e.g., original text and its translations), comparable (e.g. corpus of the same domain in multiple languages) and multilingualSingleText (for corpora that consist of segments with content in two or more languages (e.g., the transcription of a European Parliament session with MPs speaking in their native language).
- `language` (Mandatory): Specifies the language that is used in the resource part, expressed according to the BCP47 recommendation. See [language](#).
- `languageVariety` (Mandatory if applicable): Relates a language resource that contains segments in a language variety (e.g., dialect, jargon) to it. Please use for dialect corpora.
- `modalityType` (Recommended if applicable): Specifies the type of the modality represented in the resource. For instance, you can use ‘spoken language’ to describe transcribed speech corpora.
- `VideoGenre` (Recommended): A classification of video parts based on extra-linguistic and internal linguistic criteria and reflected on the video style, form or content. See [VideoGenre](#)
- `typeOfVideoContent` (Mandatory): Main type of object or people represented in the video.
- `annotation` (Mandatory if applicable): A set of features describing the annotated parts of a resource. See [annotation](#).

Example

```
<ms:CorpusVideoPart>
  <ms:corpusMediaType>CorpusVideoPart</ms:corpusMediaType>
  <ms:mediaType>http://w3id.org/meta-share/meta-share/video</ms:mediaType>
  <ms:lingualityType>http://w3id.org/meta-share/meta-share/monolingual</
↪ms:lingualityType>
  <ms:language>
    <ms:languageTag>en</ms:languageTag>
    <ms:languageId>en</ms:languageId>
  </ms:language>
  <ms:modalityType>http://w3id.org/meta-share/meta-share/bodyGesture</
↪ms:modalityType>
  <ms:modalityType>http://w3id.org/meta-share/meta-share/facialExpression</
↪ms:modalityType>
  <ms:modalityType>http://w3id.org/meta-share/meta-share/spokenLanguage</
↪ms:modalityType>
```

(continues on next page)

(continued from previous page)

```

        <ms:typeOfVideoContent>people eating at a restaurant</ms:typeOfVideoContent>
</ms:CorpusVideoPart>

<ms:CorpusVideoPart>
  <ms:corpusMediaType>CorpusVideoPart</ms:corpusMediaType>
  <ms:mediaType>http://w3id.org/meta-share/meta-share/video</ms:mediaType>
  <ms:lingualityType>http://w3id.org/meta-share/meta-share/monolingual</
↪ms:lingualityType>
  <ms:language>
    <ms:languageTag>fr</ms:languageTag>
    <ms:languageId>fr</ms:languageId>
  </ms:language>
  <ms:VideoGenre>
    <ms:CategoryLabel>documentary</ms:CategoryLabel>
  </ms:VideoGenre>
  <ms:typeOfVideoContent>birds, wild animals, plants</ms:typeOfVideoContent>
</ms:CorpusVideoPart>

```

CorpusImagePart

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.CorporusMediaPart.CorporusImagePart

Data type component

Optionality Mandatory if applicable

Explanation & Instructions

The part of a corpus (or whole corpus) that consists of images (e.g., g a corpus of photographs and their captions)

You can repeat the group of elements for multiple image parts.

The mandatory or recommended elements for the image part are:

- `mediaType` (Mandatory): Specifies the media type of a language resource (the physical medium of the contents representation). For image parts, always use the value 'image'.
- `lingualityType` (Mandatory): Indicates whether the resource includes one, two or more languages.
- `multilingualityType` (Mandatory if applicable): Indicates whether the resource (part) is parallel, comparable or mixed. If `lingualityType` = bilingual or multilingual, it is required; select one of the values for parallel (e.g., original text and its translations), comparable (e.g. corpus of the same domain in multiple languages) and multilingualSingleText (for corpora that consist of segments with content in two or more languages (e.g., the transcription of a European Parliament session with MPs speaking in their native language).
- `language` (Mandatory): Specifies the language that is used in the resource part, expressed according to the BCP47 recommendation. See [language](#).
- `languageVariety` (Mandatory if applicable): Relates a language resource that contains segments in a language variety (e.g., dialect, jargon) to it. Please use for dialect corpora.
- `modalityType` (Recommended if applicable): Specifies the type of the modality represented in the resource.
- `ImageGenre` (Recommended): A category of images characterized by a particular style, form, or content according to a specific classification scheme. See [ImageGenre](#).
- `typeOfImageContent` (Mandatory): Main type of object or people represented in the image.

- `annotation` (Mandatory if applicable): A set of features describing the annotated parts of a resource. See [annotation](#).

Example

```
<ms:CorpusImagePart>
  <ms:corpusMediaType>CorpusImagePart</ms:corpusMediaType>
  <ms:mediaType>http://w3id.org/meta-share/meta-share/image</ms:mediaType>
  <ms:lingualityType>http://w3id.org/meta-share/meta-share/monolingual</
↪ms:lingualityType>
  <ms:language>
    <ms:languageTag>el</ms:languageTag>
    <ms:languageId>el</ms:languageId>
  </ms:language>
  <ms:ImageGenre>
    <ms:CategoryLabel>comics</ms:CategoryLabel>
  </ms:ImageGenre>
  <ms:typeOfImageContent>human figures</ms:typeOfImageContent>
</ms:CorpusImagePart>
```

CorpusTextNumericalPart

Path `MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.CorporusMediaPart.CorporusTextNumericalPart`

Data type component

Optionality Mandatory if applicable

Explanation & Instructions

The part of a corpus (or whole corpus) that consists of sets of textual representations of measurements and observations linked to sensorimotor recordings

You can repeat the group of elements for multiple numerical text parts.

The mandatory or recommended elements for this part are:

- `mediaType` (Mandatory): Specifies the media type of a language resource (the physical medium of the contents representation). For numerical text parts, always use the value ‘textNumerical’.
- `typeOfTextNumericalContent` (Mandatory): Main type of object or people represented in this part.
- `numberOfParticipants` (Recommended): The number of the persons participating in the part of the resource
- `dialectAccentOfParticipants` (Recommended): Provides information on the dialect accent of the group of participants
- `geographicDistributionOfParticipants` (Recommended): Gives information on the geographic distribution of the participants
- `annotation` (Mandatory if applicable): A set of features describing the annotated parts of a resource. See [annotation](#).

Example

```

<ms:CorpusTextNumericalPart>
  <ms:corpusMediaType>CorpusImagePart</ms:corpusMediaType>
  <ms:mediaType>http://w3id.org/meta-share/meta-share/textNumerical</
↪ms:mediaType>
  <ms:typeOfTextNumericalContent>temperature measures</
↪ms:typeOfTextNumericalContent>
</ms:CorpusTextNumericalPart>

```

TextGenre

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.CorporusMediaPart.CorporusTextPart.TextGenre

Data type component

Optionality Recommended

Explanation & Instructions

A category of text characterized by a particular style, form, or content according to a specific classification scheme

You can add only a free text value at the `CategoryLabel` element; if you have used a value from an established controlled vocabulary, you can use the `TextGenreIdentifier` and the attribute `TextGenreClassificationScheme`.

Example

```

<ms:TextGenre>
  <ms:CategoryLabel>movie subtitles</ms:CategoryLabel>
</ms:TextGenre>

<ms:TextGenre>
  <ms:CategoryLabel>news articles</ms:CategoryLabel>
</ms:TextGenre>

```

AudioGenre

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.CorporusMediaPart.CorporusAudioPart

Data type component

Optionality Recommended if applicable

Explanation & Instructions

A category of audio characterized by a particular style, form, or content according to a specific classification scheme

You can add only a free text value at the `CategoryLabel` element; if you have used a value from an established controlled vocabulary, you can use the `AudioGenreIdentifier` and the attribute `AudioGenreClassificationScheme` to provide further details.

Example

```
<ms:AudioGenre>
  <ms:CategoryLabel>conference noises</ms:CategoryLabel>
</ms:AudioGenre>
```

SpeechGenre

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.CorporusMediaPart.CorporusAudioPart.SpeechGenre

Data type component

Optionality Recommended if applicable

Explanation & Instructions

A category for the conventionalized discourse of the speech part of a language resource, based on extra-linguistic and internal linguistic criteria

You can add only a free text value at the `CategoryLabel` element; if you have used a value from an established controlled vocabulary, you can use the `SpeechGenreIdentifier` and the attribute `SpeechGenreClassificationScheme` to provide further details.

Example

```
<ms:SpeechGenre>
  <ms:CategoryLabel>broadcast news</ms:CategoryLabel>
</ms:SpeechGenre>

<ms:SpeechGenre>
  <ms:CategoryLabel>monologue</ms:CategoryLabel>
</ms:SpeechGenre>
```

VideoGenre

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.CorporusMediaPart.CorporusVideoPart.VideoGenre

Data type string (+ id + scheme)

Optionality Recommended if applicable

Explanation & Instructions

A classification of video parts based on extra-linguistic and internal linguistic criteria and reflected on the video style, form or content

You can add only a free text value at the `CategoryLabel` element; if you have used a value from an established controlled vocabulary, you can use the `VideoGenreIdentifier` and the attribute `VideoClassificationScheme`

Example


```

<ms:videoGenre>
  <ms:CategoryLabel>documentaries</ms:CategoryLabel>
</ms:videoGenre>

<ms:videoGenre>
  <ms:CategoryLabel>video lectures</ms:CategoryLabel>
</ms:videoGenre>

```

ImageGenre

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.CorporusMediaPart.CorporusImagePart.ImageGenre

Data type component

Optionality Recommended

Explanation & Instructions

A category of images characterized by a particular style, form, or content according to a specific classification scheme

You can add only a free text value at the `CategoryLabel` element; if you have used a value from an established controlled vocabulary, you can use the `ImageGenreIdentifier` and the attribute `ImageClassificationScheme` to provide further details.

Example

```

<ms:imageGenre>
  <ms:CategoryLabel>human faces</ms:CategoryLabel>
</ms:imageGenre>

<ms:imageGenre>
  <ms:CategoryLabel>landscape</ms:CategoryLabel>
</ms:imageGenre>

```

annotation

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.annotation

Data type component

Optionality Mandatory if applicable

Explanation & Instructions

Links a corpus to its annotated part(s)

You must use it for annotated corpora and annotations. You can repeat it for corpora that have separate files for each annotation type, or if you want to give information such as the use of different annotation tools for each annotation level.

Enter at least the annotation type(s); if you want, you can give a more detailed description of the annotated parts - see the [annotation](#) component of the full schema.

Example

```
<ms:annotation>
  <ms:annotationType>http://w3id.org/meta-share/omtd-share/Lemma</
↪ms:annotationType>
  <ms:annotationStandoff>>false</ms:annotationStandoff>
  <ms:annotationMode>http://w3id.org/meta-share/meta-share/mixed</
↪ms:annotationMode>
  <ms:isAnnotatedBy>
    <ms:resourceName xml:lang="en">Lemmatizer</ms:resourceName>
  </ms:isAnnotatedBy>
</ms:annotation>

<ms:annotation>
  <ms:annotationType>http://w3id.org/meta-share/omtd-share/PartOfSpeech</
↪ms:annotationType>
  <ms:annotationStandoff>>false</ms:annotationStandoff>
  <ms:tagset>
    <ms:resourceName xml:lang="en">Universal Dependencies</
↪ms:resourceName>
    </ms:tagset>
    <ms:isAnnotatedBy>
      <ms:resourceName xml:lang="en">PoS tagger</ms:resourceName>
    </ms:isAnnotatedBy>
</ms:annotation>

<ms:annotation>
  <ms:annotationType>http://w3id.org/meta-share/omtd-share/
↪SyntacticAnnotationType</ms:annotationType>
</ms:annotation>
```

DatasetDistribution

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.
DatasetDistribution

Data type component

Optionality Mandatory

Explanation & Instructions

Any form with which a dataset is distributed, such as a downloadable form in a specific format (e.g., spreadsheet, plain text, etc.) or an API with which it can be accessed

You can repeat the element for multiple distributions.

The list of mandatory and recommended elements are:

- **DatasetDistributionForm** (Mandatory): The form (medium/channel) used for distributing a language resource consisting of data (e.g., a corpus, a lexicon, etc.). The typical values are ‘downloadable’, ‘accessibleThroughInterface’, ‘accessibleThroughQuery’ (see more at [DatasetDistributionForm](#)).
- **downloadLocation** (Mandatory if applicable): A URL where the language resource (mainly data but also downloadable software programmes or forms) can be downloaded from. Use this element if the value of **DatasetDistributionForm** is ‘downloadable’ and only for direct download links (i.e., from which the dataset is downloaded without the need of further actions such as clicks on a page).

- `accessLocation` (Mandatory if applicable): A URL where the resource can be accessed from; it can be used for landing pages or for cases where the resource is accessible via an interface, i.e. cases where the resource itself is not provided with a direct link for downloading. Use if the value of `DatasetDistributionForm` is 'accessibleThroughInterface' or 'accessibleThroughQuery' but also for links used for downloading corpora which are mentioned on a landing page or require some kind of action on the part of the user.
- `samplesLocation` (Recommended): Links a resource to a url (or url's) with samples of a data resource or of the input of output resource of a tool/service.
- `licenceTerms` (Mandatory): See [licenceTerms](#)
- `cost` (Mandatory if applicable): Introduces the cost for accessing a resource, formally described as a set of amount and currency unit. Please use only for resources available at a cost and not for free resources.

Depending on the parts of the corpus, you must also use one or more of the following:

- `distributionTextFeature`: See [distributionTextFeature](#)
- `distributionAudioFeature`: See [distributionAudioFeature](#)
- `distributionVideoFeature`: See [distributionVideoFeature](#)
- `distributionImageFeature`: See [distributionImageFeature](#)
- `distributionTextNumericalFeatureFeature`: See [distributiontextNumericalFeature](#)

Example

```
<ms:DatasetDistribution>
  <ms:DatasetDistributionForm>http://w3id.org/meta-share/meta-share/downloadable
  </ms:DatasetDistributionForm>
  <ms:accessLocation>https://www.someAccessURL.com</ms:accessLocation>
  <ms:samplesLocation>https://www.URLwithsamples.com</ms:samplesLocation>
  <ms:distributionTextFeature>
    <ms:size>
      <ms:amount>17601</ms:amount>
      <ms:sizeUnit>http://w3id.org/meta-share/meta-share/unit</
    </ms:sizeUnit>
    </ms:size>
    <ms:dataFormat>http://w3id.org/meta-share/omtd-share/Xml</
  </ms:dataFormat>
    <ms:characterEncoding>http://w3id.org/meta-share/meta-share/UTF-8</
  </ms:characterEncoding>
    </ms:distributionTextFeature>
    <ms:licenceTerms>
      <ms:licenceTermsName xml:lang="en">openUnder-PSI</ms:licenceTermsName>
      <ms:licenceTermsURL>https://elrc-share.eu/terms/openUnderPSI.html</
    </ms:licenceTermsURL>
    </ms:licenceTerms>
  </ms:DatasetDistribution>

<ms:DatasetDistribution>
  <ms:DatasetDistributionForm>http://w3id.org/meta-share/meta-share/
  accessibleThroughInterface</ms:DatasetDistributionForm>
  <ms:accessLocation>https://www.someAccessURL.com</ms:accessLocation>
  <ms:distributionTextFeature>
    <ms:size>
      <ms:amount>100</ms:amount>
      <ms:sizeUnit>http://w3id.org/meta-share/meta-share/text1</
    </ms:sizeUnit>
    </ms:size>
```

(continues on next page)

(continued from previous page)

```

        <ms:dataFormat>http://w3id.org/meta-share/omtd-share/Pdf</
↪ms:dataFormat>
        <ms:characterEncoding>http://w3id.org/meta-share/meta-share/UTF-8</
↪ms:characterEncoding>
        </ms:distributionTextFeature>
        <ms:licenceTerms>
            <ms:licenceTermsName xml:lang="en">some commercial licence</
↪ms:licenceTermsName>
            <ms:licenceTermsURL>https://elrc-share.eu/terms/someCommercialLicence.
↪html</ms:licenceTermsURL>
        </ms:licenceTerms>
        <ms:cost>
            <ms:amount>10000</ms:amount>
            <ms:currency>http://w3id.org/meta-share/meta-share/euro</ms:currency>
        </ms:cost>
    </ms:DatasetDistribution>

```

distributionTextFeature

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.
DatasetDistribution.distributionTextFeature

Data type component

Optionality Mandatory if applicable

Explanation & Instructions

Links to a feature that can be used for describing distinct distributable forms of text resources/parts

The following are mandatory or recommended:

- **size** (Mandatory): The size of the text part, expressed as a combination of **amount** and **sizeUnit** (with a value from a CV for **sizeUnit**).
- **dataFormat** (Mandatory): Indicates the format(s) of a data resource; it takes a value from a CV (**dataFormat**); the dataFormat includes the IANA mimetype and pointers to additional documentation for specialized formats (e.g., GATE XML, CONLL formats, etc.).
- **characterEncoding** (Recommended): Specifies the character encoding used for a language resource data distribution.

Example

```

<ms:distributionTextFeature>
    <ms:size>
        <ms:amount>9139</ms:amount>
        <ms:sizeUnit>http://w3id.org/meta-share/meta-share/sentence</
↪ms:sizeUnit>
    </ms:size>
    <ms:size>
        <ms:amount>40</ms:amount>
        <ms:sizeUnit>http://w3id.org/meta-share/meta-share/file</ms:sizeUnit>
    </ms:size>
    <ms:dataFormat>http://w3id.org/meta-share/omtd-share/Xml</ms:dataFormat>

```

(continues on next page)

(continued from previous page)

```

    <ms:characterEncoding>http://w3id.org/meta-share/meta-share/UTF-8</
↪ms:characterEncoding>
</ms:distributionTextFeature>

```

distributionAudioFeature

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.
DatasetDistribution.distributionAudioFeature

Data type component

Optionality Mandatory if applicable

Explanation & Instructions

Links to a feature that can be used for describing distinct distributable forms of audio resources/parts

The following are mandatory or recommended:

- **size** (Mandatory): The size of the audio part, expressed as a combination of `amount` and `sizeUnit` (with a value from a CV for `sizeUnit`).
- **durationOfAudio** (Recommended): Specifies the duration of the audio recording including silences, music, pauses, etc., expressed as a combination of `amount` and `durationUnit` (with a value from the CV for `durationUnit`).
- **durationOfEffectiveSpeech** (Recommended): Specifies the duration of effective speech of the audio (part of a) resource, expressed as a combination of `amount` and `durationUnit` (with a value from the CV for `durationUnit`).
- **dataFormat** (Mandatory): Indicates the format(s) of a data resource; it takes a value from a CV (`dataFormat`); the `dataFormat` includes the IANA mimetype and pointers to additional documentation for specialized formats (e.g., GATE XML, CONLL formats, etc.).
- **audioFormat** (Recommended): Indicates the format(s) of the audio (part of a) data resource, expressed as a value of `dataFormat` (with a value from a CV for `dataFormat`) and `compressed`.

Example

```

<ms:distributionAudioFeature>
  <ms:size>
    <ms:amount>10</ms:amount>
    <ms:sizeUnit>http://w3id.org/meta-share/meta-share/file</ms:sizeUnit>
  </ms:size>
  <ms:durationOfAudio>
    <ms:amount>3</ms:amount>
    <ms:durationUnit>http://w3id.org/meta-share/meta-share/hour</
↪ms:durationUnit>
  </ms:durationOfAudio>
  <ms:dataFormat>http://w3id.org/meta-share/omtd-share/wav</ms:dataFormat>
  <ms:audioFormat>
    <ms:dataFormat>http://w3id.org/meta-share/omtd-share/wav</
↪ms:dataFormat>
    <ms:compressed>true</ms:compressed>
  </ms:audioFormat>
</ms:distributionAudioFeature>

```

distributionVideoFeature

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.
DatasetDistribution.distributionVideoFeature

Data type component

Optionality Mandatory if applicable

Explanation & Instructions

Links to a feature that can be used for describing distinct distributable forms of video resources/parts

The following are mandatory or recommended:

- **size** (Mandatory): The size of the video part, expressed as a combination of `amount` and `sizeUnit` (with a value from a CV for `sizeUnit`).
- **durationOfVideo** (Recommended): Specifies the duration of the video recording, expressed as a combination of `amount` and `durationUnit` (with a value from the CV for `durationUnit`).
- **dataFormat** (Mandatory): Indicates the format(s) of a data resource; it takes a value from a CV (`dataFormat`); the `dataFormat` includes the IANA mimetype and pointers to additional documentation for specialized formats (e.g., GATE XML, CONLL formats, etc.).
- **videoFormat** (Recommended): Indicates the format(s) of the video (part of a) data resource, expressed as a value of `dataFormat` (with a value from a CV for `dataFormat`) and `compressed`.

Example

```
<ms:distributionVideoFeature>
  <ms:size>
    <ms:amount>9139</ms:amount>
    <ms:sizeUnit>http://w3id.org/meta-share/meta-share/screen</
↪ms:sizeUnit>
  </ms:size>
  <ms:size>
    <ms:amount>40</ms:amount>
    <ms:sizeUnit>http://w3id.org/meta-share/meta-share/file</ms:sizeUnit>
  </ms:size>
  <ms:durationOfVideo>
    <ms:amount>40</ms:amount>
    <ms:durationUnit>http://w3id.org/meta-share/meta-share/hour</
↪ms:durationUnit>
  </ms:durationOfVideo>
  <ms:dataFormat>http://w3id.org/meta-share/omtd-share/wav</ms:dataFormat>
  <ms:videoFormat>
    <ms:dataFormat>http://w3id.org/meta-share/omtd-share/wav</
↪ms:dataFormat>
    <ms:compressed>true</ms:compressed>
  </ms:videoFormat>
```

distributionImageFeature

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corpus.DatasetDistribution.distributionImageFeature

Data type component

Optionality Mandatory if applicable

Explanation & Instructions

Links to a feature that can be used for describing distinct distributable forms of image resources/parts

The following are mandatory or recommended:

- **size** (Mandatory): The size of the image part, expressed as a combination of **amount** and **sizeUnit** (with a value from a CV for **sizeUnit**).
- **dataFormat** (Mandatory): Indicates the format(s) of a data resource; it takes a value from a CV (**dataFormat**); the **dataFormat** includes the IANA mimetype and pointers to additional documentation for specialized formats (e.g., GATE XML, CONLL formats, etc.).
- **imageFormat** (Mandatory): Indicates the format(s) of the image (part of a) data resource, expressed as a value of **dataFormat** (with a value from a CV for **dataFormat**) and **compressed**.

Example

```
<ms:distributionImageFeature>
  <ms:size>
    <ms:amount>100</ms:amount>
    <ms:sizeUnit>http://w3id.org/meta-share/meta-share/file</ms:sizeUnit>
  </ms:size>
  <ms:dataFormat>http://w3id.org/meta-share/omtd-share/Pdf</ms:dataFormat>
  <ms:imageFormat>
    <ms:dataFormat>http://w3id.org/meta-share/omtd-share/Pdf</
  <ms:dataFormat>
    <ms:compressed>true</ms:compressed>
  </ms:imageFormat>
</ms:distributionImageFeature>
```

distributiontextNumericalFeature

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corpus.DatasetDistribution.distributiontextNumericalFeature

Data type component

Optionality Mandatory if applicable

Explanation & Instructions

Links to a feature that can be used for describing distinct distributable forms of image resources/parts

The following are mandatory or recommended:

- **size** (Mandatory): The size of the image part, expressed as a combination of **amount** and **sizeUnit** (with a value from a CV for **sizeUnit**).

- `dataFormat` (Mandatory): Indicates the format(s) of a data resource; it takes a value from a CV (`dataFormat`); the `dataFormat` includes the IANA mimetype and pointers to additional documentation for specialized formats (e.g., GATE XML, CONLL formats, etc.).

Example

```
<ms:distributionTextNumericalFeature>
  <ms:size>
    <ms:amount>30</ms:amount>
    <ms:sizeUnit>http://w3id.org/meta-share/meta-share/file</ms:sizeUnit>
  </ms:size>
  <ms:dataFormat>http://w3id.org/meta-share/omtd-share/Pdf</ms:dataFormat>
  <ms:imageFormat>
    <ms:dataFormat>http://w3id.org/meta-share/omtd-share/Pdf</
  <ms:dataFormat>
    <ms:compressed>true</ms:compressed>
  </ms:imageFormat>
</ms:distributionTextNumericalFeature>
```

1.52.6 Minimal elements for language descriptions

This page describes the minimal metadata elements specific to **language descriptions**.

1. Overview

Under language descriptions we subsume models and computational grammars, distinguished with the `languageDescriptionSubclass` element. Depending on the value, a further set of features pertinent to the subclass is recommended for them. In addition, as for corpora, we also cater for multimedia resources, which include not only text but also audio, video and image files. To cater for these cases, the notion of “**media part**” is introduced in the model. Thus, a language description consists of *at least one* text, video and image parts. Depending on the media part type, the `DatasetDistribution` component includes a set of text, video, etc. **distribution features**.

The first table below has all the elements (mandatory and recommended) for a language description. The second table presents the mandatory and recommended elements for each subclass. The third table presents the mandatory and recommended elements for each media part. Finally, the fourth table presents the mandatory and recommended elements for the `Distribution` component, which includes elements that are specific to each media part.

Table 1 - Language description common elements

Element name	Optionality	Section	Tab
<code>languageDescriptionSubclass</code>	M	LD	technical
<code>MLModel</code>	MA	LD	technical
<code>Grammar</code>	MA	LD	technical
<code>Ngram model</code>	MA	LD	technical
<code>personalDataIncluded</code>	M	LD	technical
<code>personalDataDetails</code>	RA	LD	technical
<code>sensitiveDataIncluded</code>	M	LD	technical
<code>sensitiveDataDetails</code>	RA	LD	technical
<code>anonymized</code>	MA	LD	technical
<code>anonymizationDetails</code>	RA	LD	technical
<code>requiresHardware</code>	R	LD	technical

Table 2 - Language description subclass elements

Element name	Optionality	Section	Tab
<i>MLModel</i>	MA	LD	technical
modelVariant	R	LD	technical
typesystem	R	LD	technical
method	R	LD	technical
MLframework1	R	LD	technical
trainingCorpusDetails	R	LD	technical
<i>Grammar</i>	MA	LD	technical
encodingLevel	M	LD	technical
compliesWith	R	LD	technical
Formalism	R	LD	technical
ldTask	R	LD	technical
<i>Ngrammodel</i>	MA	LD	technical
baseItem	M	LD	technical
order	M	LD	technical
perplexity	R	LD	technical

Table 3 - Media parts

Element name	Optionality	Section	Tab
<i>textPart</i>	MA	LD	Part
lingualityType	M	LD	Part
multilingualityType	MA	LD	Part
multilingualityTypeDetails	R	LD	Part
language	M	LD	Part
metalanguage	R	LD	Part
<i>videoPart</i>	MA	LD	Part
lingualityType	M	LD	Part
multilingualityType	MA	LD	Part
multilingualityTypeDetails	RA	LD	Part
language	M	LD	Part
metalanguage	R	LD	Part
typeOfVideoContent	M	LD	Part
<i>imagePart</i>	MA	LD	Part
lingualityType	M	LD	Part
multilingualityType	RA	LD	Part
multilingualityTypeDetails	RA	LD	Part
language	M	LD	Part
metalanguage	R	LD	Part
typeOfImageContent	M	LD	Part

Table 4 - Distribution

Element name	Optionality	Section	Tab
DatasetDistribution	M	Distribution	Technical
DatasetDistributionForm	M	Distribution	Technical
downloadLocation	MA	Distribution	Technical
accessLocation	MA	Distribution	Technical
distributionLocation	MA	Distribution	Technical
samplesLocation	R	Distribution	Technical
distributionTextFeature	MA	Distribution	Technical
distributionVideoFeature	MA	Distribution	Technical
distributionImageFeature	MA	Distribution	Technical
licenceTerms	M	Distribution	Technical
cost	R	Distribution	Technical
membershipInstitution	R	Distribution	Technical

2. Element presentation

In this section all the aforementioned elements are presented each one separately. The presentation follows the order of the elements in the tables of the previous section.

LanguageDescription

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.
LanguageDescription

Data type component

Optionality Mandatory

Explanation & Instructions

Wraps together elements for language descriptions

Example

```
<ms:LRS subclass>
  <ms:LanguageDescription>
    <ms:lrType>LanguageDescription</ms:lrType>
    ...
  </ms:LanguageDescription>
</ms:LRS subclass>
```

LanguageDescriptionSubclass

Path `MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.
LanguageDescription.LanguageDescriptionSubclass`

Data type component

Optionality Mandatory

Explanation & Instructions

The type of the language description (used for documentation purposes)

It wraps the set of elements that must be used for the Language Description subclasses:

- Machine Learning Model: See [MLModel](#)
- N-gram model: See [NGramModel](#)
- Computational grammar: See [Grammar](#)

Example

```
<ms:LanguageDescriptionSubclass>
  ...
</ms:LanguageDescriptionSubclass>
```

MLModel

Path `MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.
LanguageDescription.LanguageDescriptionSubclass.MLModel`

Data type Component

Optionality Mandatory if applicable

Explanation & Instructions

Mandatory for Machine Learning (ML) models; a ML model, for our purposes, is defined as “The model artifact that is created through a training process involving an ML algorithm (that is, the learning algorithm) and the training data to learn from”

The following set of elements are mandatory or recommended for ML models:

- `ldSubclassType` (Mandatory): Used to mark the subclass of a language description. For ML models, the value is fixed to ‘MLModel’.
- `modelVariant` (Recommended): Introduces a label that can be used to identify the variant of a ML model.
- `typesystem` (Recommended): Specifies the typesystem (preferably through an identifier or URL) that has been used for the annotation of a resource or that is required for the input resource of a tool/service or that should be used (dependency) for the annotation or used in the training of a ML model.
- `method` (Recommended): Specifies the method used for the development of a tool/service or the ML model. You must use one of the values from the CV.
- `mlFramework` (Recommended): Specifies the framework that has been used for developing a model (e.g. keras, tensorflow, etc.).
- `trainingCorpusDetails` (Recommended): Provides a detailed description of the training corpus (e.g., size, number of features , etc.).

Example

```
<ms:MLModel>
  <ms:ldSubclassType>MlModel</ms:ldSubclassType>
  <ms:modelVariant>factored</ms:modelVariant>
  <ms:typesystem>
    <ms:resourceName xml:lang="en">Universal dependencies</
↪ms:resourceName>
    <ms:version>undefined</ms:version>
  </ms:typesystem>
  <ms:method>http://w3id.org/meta-share/omtd-share/DeepLearning</ms:method>
  <ms:mlFramework>tensorflow</ms:mlFramework>
  <ms:trainingCorpusDetails xml:lang="en">Trained on a corpus of tweets</
↪ms:trainingCorpusDetails>
</ms:MLModel>
```

Grammar

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.
LanguageDescription.LanguageDescriptionSubclass.Grammar

Data type Component

Optionality Mandatory if applicable

Explanation & Instructions

Mandatory for grammars; grammar for our purposes is defined as “A set of rules governing what strings are valid or allowable in a language or text” [<https://en.oxforddictionaries.com/definition/grammar>]

The following set of elements are mandatory or recommended for computational grammars:

- `ldSubclassType` (Mandatory): Used to mark the subclass of a language description. For grammars, the value is fixed to ‘Grammar.’
- `encodingLevel` (Mandatory): Classifies the contents of a lexical/conceptual resource or language description as regards the linguistic level of analysis it caters for.
- `compliesWith` (Recommended): Specifies the vocabulary/standard/best practice to which a resource is compliant with.
- `formalism` (Recommended): Specifies the formalism (bibliographic reference, URL, name) used for the creation/enrichment of the resource (grammar or tool/service).
- `ldTask` (Recommended): Specifies the task performed by the language description.

Example

```
<ms:Grammar>
  <ms:ldSubclassType>Grammar</ms:ldSubclassType>
  <ms:encodingLevel>http://w3id.org/meta-share/meta-share/morphology</
↪ms:encodingLevel>
  <ms:compliesWith>http://w3id.org/meta-share/meta-share/GrAF</ms:compliesWith>
</ms:Grammar>
```

NGramModel

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.
LanguageDescription.LanguageDescriptionSubclass.NGramModel

Data type Component

Optionality Mandatory if applicable

Explanation & Instructions

Mandatory for n-gram models; n-gram model for our purposes is defined as “A language model consisting of n-grams, i.e. specific sequences of a number of words”

The following set of elements are mandatory or recommended for Machine Learning models:

- `ldSubclassType` (Mandatory): Used to mark the subclass of a language description. For ML models, the value is fixed to ‘NGramModel’.
- `baseItem` (Mandatory): Type of item that is represented in the n-gram resource.
- `order` (Mandatory): Specifies the maximum number of items in the sequence.
- `perplexity` (Recommended): Provides information on the perplexity derived from running on test set taken from the same corpus.

Example

```
<ms:NGramModel>
  <ms:ldSubclassType>NGramModel</ms:ldSubclassType>
  <ms:baseItem>http://w3id.org/meta-share/meta-share/word</ms:baseItem>
  <ms:order>5</ms:order>
</ms:NGramModel>
```

1.52.7 Minimal elements for lexical/conceptual resources

This page describes the minimal metadata elements specific to **lexical/conceptual resources**.

1. Overview

Lexical/Conceptual resources comprise computational lexica, gazetteers, ontologies, term lists, etc. Under this class, we also include multimedia dictionaries, sign language resources, etc. which include not only text but also audio, video and image files. To cater for these cases, the notion of “**media part**” is introduced in the model. Thus, a lexical/conceptual resource consists of *at least one* text, audio, video, image and numerical text parts. Depending on the media part type, the DatasetDistribution component includes a set of text, audio, video, etc. **distribution features**.

The first table below has all the elements (mandatory and recommended) for a lexical/conceptual resource. The second table presents the mandatory and recommended elements for each media part. The third table presents the mandatory and recommended elements for the Distribution component, which includes elements that are specific to each media part.

Table 1 - Lexical/Conceptual resource common elements

Element name	Optionality	Section	Tab
lcrSubclass	R	LCR	technical
encodingLevel	M	LCR	technical
contentType	R	LCR	technical
compliesWith	R	LCR	technical
personalDataIncluded	M	LCR	technical
personalDataDetails	RA	LCR	technical
sensitiveDataIncluded	M	LCR	technical
sensitiveDataDetails	RA	LCR	technical
anonymized	MA	LCR	technical
anonymizationDetails	RA	LCR	technical

Table 2 - Media parts

Element name	Optionality	Section	Tab
<i>textPart</i>	MA	LCR	Part
lingualityType	M	LCR	Part
multilingualityType	MA	LCR	Part
multilingualityTypeDetails	R	LCR	Part
language	M	LCR	Part
metalanguage	R	LCR	Part
<i>audioPart</i>	MA	LCR	Part
lingualityType	M	LCR	Part
multilingualityType	MA	LCR	Part
multilingualityTypeDetails	RA	LCR	Part
language	M	LCR	Part
metalanguage	R	LCR	Part
<i>videoPart</i>	MA	LCR	Part
lingualityType	M	LCR	Part
multilingualityType	MA	LCR	Part
multilingualityTypeDetails	RA	LCR	Part
language	M	LCR	Part
metalanguage	R	LCR	Part
typeOfVideoContent	M	LCR	Part
<i>imagePart</i>	MA	LCR	Part
lingualityType	M	LCR	Part
multilingualityType	RA	LCR	Part
multilingualityTypeDetails	RA	LCR	Part
language	M	LCR	Part
metalanguage	R	LCR	Part
typeOfImageContent	M	LCR	Part

Table 3 - Distribution

Element name	Optionality	Section	Tab
DatasetDistribution	M	Distribution	Technical
DatasetDistributionForm	M	Distribution	Technical
downloadLocation	MA	Distribution	Technical
accessLocation	MA	Distribution	Technical
distributionLocation	MA	Distribution	Technical
samplesLocation	R	Distribution	Technical
distributionTextFeature	MA	Distribution	Technical
distributionAudioFeature	MA	Distribution	Technical
distributionVideoFeature	MA	Distribution	Technical
distributionImageFeature	MA	Distribution	Technical
distributionTextNumericalFeature	MA	Distribution	Technical
licenceTerms	M	Distribution	Technical
cost	R	Distribution	Technical
membershipInstitution	R	Distribution	Technical

2. Element presentation

In this section all the aforementioned elements are presented each one separately. The presentation follows the order of the elements in the tables of the previous section.

LexicalConceptualResource

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.
LexicalConceptualResource

Data type component

Optionality Mandatory

Explanation & Instructions

Wraps together elements for lexical/conceptual resources

Example

```
<ms:LRSubclass>
  <ms:LexicalConceptualResource>
    <ms:lrType>LexicalConceptualResource</ms:lrType>
    ...
  </ms:LexicalConceptualResource>
</ms:LRSubclass>
```

lcrSubclass

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.
LexicalConceptualResource.lcrSubclass

Data type CV (lcrSubclass)

Optionality Recommended

Explanation & Instructions

Introduces a classification of lexical/conceptual resources into types (used for descriptive reasons)

Example

```
<lcrSubclass>http://w3id.org/meta-share/meta-share/computationalLexicon</lcrSubclass>  
<lcrSubclass>http://w3id.org/meta-share/meta-share/ontology</lcrSubclass>
```

encodingLevel

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.
LexicalConceptualResource.encodingLevel

Data type CV (encodingLevel)

Optionality Mandatory

Explanation & Instructions

Classifies the contents of a lexical/conceptual resource or language description as regards the linguistic level of analysis it caters for

You can repeat the element for multiple encoding levels.

Example

```
<ms:encodingLevel>http://w3id.org/meta-share/meta-share/phonology</ms:encodingLevel>  
<ms:encodingLevel>http://w3id.org/meta-share/meta-share/semantics</ms:encodingLevel>
```

ContentType

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.
LexicalConceptualResource.ContentType

Data type CV (ContentType)

Optionality Recommended

Explanation & Instructions

A more detailed account of the linguistic information contained in the lexical/conceptual resource

You can repeat the element for multiple content types.

Example


```
<ms:ContentType>http://w3id.org/meta-share/meta-share/collocation</ms:ContentType>
<ms:ContentType>http://w3id.org/meta-share/meta-share/definition</ms:ContentType>
```

compliesWith

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.LexicalConceptualResource.ContentType

Data type CV (**compliesWith**)

Optionality Recommended

Explanation & Instructions

Specifies the vocabulary/standard/best practice to which a resource is compliant with

Example

```
<ms:compliesWith>http://w3id.org/meta-share/meta-share/LMF</ms:compliesWith>
```

LexicalConceptualResourceTextPart

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.CorporusMediaPart.LexicalConceptualResourceTextPart

Data type component

Optionality Mandatory if applicable

Explanation & Instructions

A part (or whole set) of a lexical/conceptual resource that consists of textual elements

You can repeat the group of elements for multiple textual parts.

The mandatory or recommended elements for the text part of lexical/conceptual resources are:

- **mediaType** (Mandatory): Specifies the media type of a language resource (the physical medium of the contents representation). For text parts, always use the value ‘text’.
- **lingualityType** (Mandatory): Indicates whether the resource includes one, two or more languages.
- **multilingualityType** (Recommended if applicable): Indicates whether the resource (part) is parallel, comparable or mixed. If **lingualityType** = bilingual or multilingual, it is recommended for lexical/conceptual resources; select one of the values for parallel (e.g., bilingual dictionaries with source and translation equivalents), comparable (e.g. lexica of the same domain in multiple languages).
- **language** (Mandatory): Specifies the language that is used in the resource part, expressed according to the BCP47 recommendation. See [language](#).
- **languageVariety** (Mandatory if applicable): Relates a language resource that contains segments in a language variety (e.g., dialect, jargon) to it. Please use for dialect corpora.
- **metalanguage** (Recommended if applicable): Specifies the language that is used as support for the resource (e.g., English for a grammar of French described in English or for a French dictionary with English definitions), expressed according to the BCP47 recommendation. See [language](#).

- `modalityType` (Recommended if applicable): Specifies the type of the modality represented in the resource. For instance, you can use ‘spoken language’ to describe transcribed speech corpora.

Example

```
<ms:LexicalConceptualResourceMediaPart>
  <ms:LexicalConceptualResourceTextPart>
    <ms:lcrMediaType>LexicalConceptualResourceTextPart</ms:lcrMediaType>
    <ms:mediaType>http://w3id.org/meta-share/meta-share/text</
↪ms:mediaType>
    <ms:lingualityType>http://w3id.org/meta-share/meta-share/bilingual</
↪ms:lingualityType>
    <ms:multilingualityType>http://w3id.org/meta-share/meta-share/parallel
↪</ms:multilingualityType>
    <ms:language>
      <ms:languageTag>en-US</ms:languageTag>
      <ms:languageId>en</ms:languageId>
      <ms:regionId>US</ms:regionId>
    </ms:language>
    <ms:language>
      <ms:languageTag>es</ms:languageTag>
      <ms:languageId>es</ms:languageId>
    </ms:language>
    <ms:metalanguage>
      <ms:languageTag>es</ms:languageTag>
      <ms:languageId>es</ms:languageId>
    </metalanguage>
    </ms:language>
  </ms:LexicalConceptualResourceTextPart>
</ms:LexicalConceptualResourceMediaPart>
```

LexicalConceptualResourceAudioPart

Path `MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.CorporusMediaPart.LexicalConceptualResourceAudioPart`

Data type component

Optionality Mandatory if applicable

Explanation & Instructions

A part (or whole set) of a lexical/conceptual resource that consists of audio elements

You can repeat the group of elements for multiple audio parts.

The mandatory or recommended elements for the audio part of lexical/conceptual resources are:

- `mediaType` (Mandatory): Specifies the media type of a language resource (the physical medium of the contents representation). For audio parts, always use the value ‘audio’.
- `lingualityType` (Mandatory): Indicates whether the resource includes one, two or more languages.
- `multilingualityType` (Recommended if applicable): Indicates whether the resource (part) is parallel, comparable or mixed. If `lingualityType` = bilingual or multilingual, it is recommended for lexical/conceptual resources; select one of the values for parallel (e.g., bilingual dictionaries with source and translation equivalents), comparable (e.g. lexica of the same domain in multiple languages).

- `language` (Mandatory): Specifies the language that is used in the resource part, expressed according to the BCP47 recommendation. See *language*.
- `languageVariety` (Mandatory if applicable): Relates a language resource that contains segments in a language variety (e.g., dialect, jargon) to it. Please use for dialect corpora.
- `metalanguage` (Recommended if applicable): Specifies the language that is used as support for the resource (e.g., English for a grammar of French described in English or for a French dictionary with English definitions), expressed according to the BCP47 recommendation. See *language*.
- `modalityType` (Recommended if applicable): Specifies the type of the modality represented in the resource. For instance, you can use 'spoken language' to describe transcribed speech corpora.

Example

```
<ms:LexicalConceptualResourceMediaPart>
  <ms:LexicalConceptualResourceAudioPart>
    <ms:lcrMediaType>LexicalConceptualResourceAudioPart</ms:lcrMediaType>
    <ms:mediaType>http://w3id.org/meta-share/meta-share/audio</
↪ms:mediaType>
    <ms:lingualityType>http://w3id.org/meta-share/meta-share/bilingual</
↪ms:lingualityType>
    <ms:multilingualityType>http://w3id.org/meta-share/meta-share/parallel
↪</ms:multilingualityType>
    <ms:language>
      <ms:languageTag>en-US</ms:languageTag>
      <ms:languageId>en</ms:languageId>
      <ms:regionId>US</ms:regionId>
    </ms:language>
    <ms:language>
      <ms:languageTag>es</ms:languageTag>
      <ms:languageId>es</ms:languageId>
    </ms:language>
    <ms:metalanguage>
      <ms:languageTag>es</ms:languageTag>
      <ms:languageId>es</ms:languageId>
    </metalanguage>
    </ms:language>
  </ms:LexicalConceptualResourceAudioPart>
</ms:LexicalConceptualResourceMediaPart>
```

LexicalConceptualResourceVideoPart

Path `MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.CorporusMediaPart.LexicalConceptualResourceVideoPart`

Data type component

Optionality Mandatory if applicable

Explanation & Instructions

A part (or whole set) of a lexical/conceptual resource that consists of video elements

You can repeat the group of elements for multiple video parts.

The mandatory or recommended elements for the video part of lexical/conceptual resources are:

- `mediaType` (Mandatory): Specifies the media type of a language resource (the physical medium of the contents representation). For video parts, always use the value ‘video’.
- `lingualityType` (Mandatory): Indicates whether the resource includes one, two or more languages.
- `multilingualityType` (Recommended if applicable): Indicates whether the resource (part) is parallel, comparable or mixed. If `lingualityType` = bilingual or multilingual, it is recommended for lexical/conceptual resources; select one of the values for parallel (e.g., bilingual dictionaries with source and translation equivalents), comparable (e.g. lexica of the same domain in multiple languages).
- `language` (Mandatory): Specifies the language that is used in the resource part, expressed according to the BCP47 recommendation. See [language](#).
- `languageVariety` (Mandatory if applicable): Relates a language resource that contains segments in a language variety (e.g., dialect, jargon) to it. Please use for dialect corpora.
- `metalinguage` (Recommended if applicable): Specifies the language that is used as support for the resource (e.g., English for a grammar of French described in English or for a French dictionary with English definitions), expressed according to the BCP47 recommendation. See [language](#).
- `modalityType` (Recommended if applicable): Specifies the type of the modality represented in the resource. For instance, you can use ‘spoken language’ to describe transcribed speech corpora.

Example

```
<ms:LexicalConceptualResourceMediaPart>
  <ms:LexicalConceptualResourceVideoPart>
    <ms:lcrMediaType>LexicalConceptualResourceVideoPart</ms:lcrMediaType>
    <ms:mediaType>http://w3id.org/meta-share/meta-share/video</
↪ms:mediaType>
    <ms:lingualityType>http://w3id.org/meta-share/meta-share/bilingual</
↪ms:lingualityType>
    <ms:multilingualityType>http://w3id.org/meta-share/meta-share/parallel
↪</ms:multilingualityType>
    <ms:language>
      <ms:languageTag>en-US</ms:languageTag>
      <ms:languageId>en</ms:languageId>
      <ms:regionId>US</ms:regionId>
    </ms:language>
    <ms:language>
      <ms:languageTag>es</ms:languageTag>
      <ms:languageId>es</ms:languageId>
    </ms:language>
    <ms:metalinguage>
      <ms:languageTag>es</ms:languageTag>
      <ms:languageId>es</ms:languageId>
    </metalinguage>
    </ms:language>
  </ms:LexicalConceptualResourceVideoPart>
</ms:LexicalConceptualResourceMediaPart>
```

LexicalConceptualResourceImagePart

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.CorporusMediaPart.LexicalConceptualResourceImagePart

Data type component

Optionality Mandatory if applicable

Explanation & Instructions

A part (or whole set) of a lexical/conceptual resource that consists of image elements

You can repeat the group of elements for multiple image parts.

The mandatory or recommended elements for the image part of lexical/conceptual resources are:

- `mediaType` (Mandatory): Specifies the media type of a language resource (the physical medium of the contents representation). For image parts, always use the value 'image'.
- `lingualityType` (Mandatory): Indicates whether the resource includes one, two or more languages.
- `multilingualityType` (Recommended if applicable): Indicates whether the resource (part) is parallel, comparable or mixed. If `lingualityType` = bilingual or multilingual, it is recommended for lexical/conceptual resources; select one of the values for parallel (e.g., bilingual dictionaries with source and translation equivalents), comparable (e.g. lexica of the same domain in multiple languages).
- `language` (Mandatory): Specifies the language that is used in the resource part, expressed according to the BCP47 recommendation. See [language](#).
- `languageVariety` (Mandatory if applicable): Relates a language resource that contains segments in a language variety (e.g., dialect, jargon) to it. Please use for dialect corpora.
- `metalanguage` (Recommended if applicable): Specifies the language that is used as support for the resource (e.g., English for a grammar of French described in English or for a French dictionary with English definitions), expressed according to the BCP47 recommendation. See [language](#).
- `modalityType` (Recommended if applicable): Specifies the type of the modality represented in the resource. For instance, you can use 'spoken language' to describe transcribed speech corpora.

Example

```
<ms:LexicalConceptualResourceMediaPart>
  <ms:LexicalConceptualResourceImagePart>
    <ms:lcrMediaType>LexicalConceptualResourceImagePart</ms:lcrMediaType>
    <ms:mediaType>http://w3id.org/meta-share/meta-share/image</
↪ms:mediaType>
    <ms:lingualityType>http://w3id.org/meta-share/meta-share/bilingual</
↪ms:lingualityType>
    <ms:multilingualityType>http://w3id.org/meta-share/meta-share/parallel
↪</ms:multilingualityType>
    <ms:language>
      <ms:languageTag>en-US</ms:languageTag>
      <ms:languageId>en</ms:languageId>
      <ms:regionId>US</ms:regionId>
    </ms:language>
    <ms:language>
      <ms:languageTag>es</ms:languageTag>
      <ms:languageId>es</ms:languageId>
    </ms:language>
    <ms:metalanguage>
      <ms:languageTag>es</ms:languageTag>
```

(continues on next page)

(continued from previous page)

```
<ms:languageId>es</ms:languageId>
</metalanguage>
</ms:language>
</ms:LexicalConceptualResourceImagePart>
</ms:LexicalConceptualResourceMediaPart>
```

1.52.8 Minimal elements for projects

This page describes the minimal metadata elements specific to **projects**.

N.B. The interactive editor supports the full schema, i.e. it also includes optional elements.

1. Overview

Element name	Optionality	Tab
projectName	M	Identity
ProjectIdentifier	R	Identity
projectShortName	R	Identity
projectAlternativeName	R	Identity
projectSummary	R	Identity
website	R	Identity
email	R	Identity
logo	R	Identity
fundingType	R	Identity
funder	R	Identity
fundingCountry	R	Identity
socialMediaOccupationalAccount	R	Identity
LTArea	R	Categories
domain	R	Categories
keyword	R	Categories

2. Element presentation

In this section all the aforementioned elements are presented each one separately. The presentation follows the order of the elements in the table of the previous section.

Project

Path MetadataRecord.DescribedEntity.Project

Data type component

Optionality Mandatory

Explanation & Instructions

Wraps together elements for projects

Example

```
<ms:Project>
  <ms:entityType>project</ms:entityType>
  ...
</ms:Project>
```

ProjectIdentifier

Path MetadataRecord.DescribedEntity.Project.ProjectIdentifier

Data type string

Optionality Recommended

Explanations & Instructions

A string (e.g., PID, internal to an organization, issued by the funding authority, etc.) used to uniquely identify a project

You must also use the attribute `ProjectIdentifierScheme` to specify the name of the scheme according to which an identifier is assigned to a project by the authority that issues it. [ProjectIdentifierScheme](#) for details.

Example

```
<ms:ProjectIdentifier ms:ProjectIdentifierScheme="http://w3id.org/meta-share/meta-
↪share/cordis">219608</ms:ProjectIdentifier>

<ms:ProjectIdentifier ms:ProjectIdentifierScheme="http://w3id.org/meta-share/meta-
↪share/cordis">219378</ms:ProjectIdentifier>
```

projectName

Path MetadataRecord.DescribedEntity.Project.projectName

Data type multilingual string

Optionality Mandatory

Explanations & Instructions

The full name (title) of a project

Example

```
<ms:projectName xml:lang="en">Browser-based Multilingual Translation</ms:projectName>

<ms:projectName xml:lang="en">European Language Grid</ms:projectName>
```

projectShortName

Path MetadataRecord.DescribedEntity.Project.projectShortName

Data type multilingual string

Optionality Recommended

Explanations & Instructions

Introduces a short name (e.g., acronym, abbreviated form) by which a project is known

Example

```
<ms:projectShortName xml:lang="en">Bergamot</ms:projectShortName>
<ms:projectShortName xml:lang="en">ELG</ms:projectShortName>
```

projectAlternativeName

Path MetadataRecord.DescribedEntity.Project.projectAlternativeName

Data type multilingual string

Optionality Recommended

Explanations & Instructions

Introduces an alternative name (other than the short name) used for a project

Example

```
<ms:projectAlternativeName xml:lang="en">The European Language Grid</ms:projectName>
```

projectSummary

Path MetadataRecord.DescribedEntity.Project.projectSummary

Data type multilingual string

Optionality Recommended

Explanations & Instructions

Introduces a short description (in free text) of the main objectives, mission or contents of the project

Example


```
<ms:projectSummary xml:lang="en">The Bergamot project will add and improve client-
→side machine translation in a web browser. Unlike current cloud-based options,
→running directly on users' machines empowers citizens to preserve their privacy,
→and increases the uptake of language technologies in Europe in various sectors that
→require confidentiality. Free software integrated with an open-source web browser,
→such as Mozilla Firefox, will enable bottom-up adoption by non-experts, resulting
→in cost savings for private and public sector users who would otherwise procure
→translation or operate monolingually. To understand and support non-expert users,
→our user experience work package researches their needs and creates the user
→interface. Rather than simply translating text, this interface will expose
→improved quality estimates, addressing the rising public debate on algorithmic
→trust. Building on quality estimation research, we will enable users to
→confidently generate text in a language they do not speak, enabling cross-lingual
→online form filling. To improve quality overall, dynamic domain adaptation
→research addresses the peculiar writing style of a website or user by adapting
→translation on the fly using local information too private to upload to the cloud.
→These applications require adaptation and inference to run on desktop hardware with
→compact model downloads, which we address with neural network efficiency research.
→Our combined research on user experience, domain adaptation, quality estimation,
→outbound translation, and efficiency support a broad browser-based innovation plan.
→</ms:projectSummary>
```

```
<ms:projectSummary xml:lang="en">With 24 official EU and many more additional
→languages, multilingualism in Europe and an inclusive Digital Single Market can
→only be enabled through Language Technologies (LTs). European LT business is
→dominated by thousands of SMEs and a few large players. Many are world-class, with
→technologies that outperform the global players. However, European LT business is
→also fragmented by nation states, languages, verticals and sectors. Likewise,
→while much of European LT research is world-class, with results transferred into
→industry and commercial products, its full impact is held back by fragmentation.
→The key issue and challenge is the fragmentation of the European LT landscape. The
→European Language Grid (ELG) project will address this fragmentation by
→establishing the ELG as the primary platform for LT in Europe. The ELG will be a
→scalable cloud platform, providing, in an easy-to-integrate way, access to hundreds
→of commercial and non-commercial Language Technologies for all European languages,
→including running tools and services as well as data sets and resources. It will
→enable the commercial and non-commercial European LT community to deposit and
→upload their technologies and data sets into the ELG, to deploy them through the
→grid, and to connect with other resources. The ELG will boost the Multilingual
→Digital Single Market towards a thriving European LT community, creating new jobs
→and opportunities. Through open calls, up to 20 pilot projects will be financially
→supported to demonstrate the usefulness of the ELG. The proposal is rooted in the
→experience of a consortium with partners involved in all relevant initiatives.
→Based on these, 30\ national competence centres and the European LT Board will be
→set up for European coordination. The ELG will foster "language technologies for
→Europe built in Europe", tailored to our languages and cultures and to our societal
→and economical demands, benefitting the European citizen, society, innovation and
→industry.</ms:projectSummary>
```

website

Path MetadataRecord.DescribedEntity.Project.website

Data type URL

Optionality Recommended

Explanations & Instructions

Links to a URL that acts as the primary page (like a table of contents) introducing information about an organization (e.g., products, contact information, etc.) or project

Example

```
<ms:website>https://browser.mt/</ms:website>  
  
<ms:website>https://www.european-language-grid.eu/</ms:website>
```

email

Path MetadataRecord.DescribedEntity.Project.email

Data type string

Optionality Recommended

Explanation & Instructions

Points to the email address used for information purposes of a project

Example

```
<ms:email>info@project.eu</ms:email>
```

logo

Path MetadataRecord.DescribedEntity.Project.logo

Data type URL

Optionality Recommended *Explanations & Instructions*

Links to a URL with an image file containing a symbol or graphic object used to identify the entity

In the interactive editor, users can also upload an image file.

Example

```
<ms:logo>https://ufal.mff.cuni.cz/sites/default/files/styles/drupal_projects_logo_  
→style/public/bergamot_logo.png</ms:logo>  
  
<ms:logo>https://www.european-language-grid.eu/wp-content/themes/elg_theme/fab/image/  
→logo/rgb_elg__logo--colour.svg</ms:logo>
```

fundingType

Path MetadataRecord.DescribedEntity.Project.fundingType

Data type CV (fundingType)

Optionality Recommended

Explanations & Instructions

Specifies the type of funding of a project with regard to the source of the funding

Example

```
<ms:fundingType>http://w3id.org/meta-share/meta-share/euFunds</ms:fundingType>
```

funder

Path MetadataRecord.DescribedEntity.Project.funder

Data type component

Optionality Recommended

Explanations & Instructions

Identifies the person/organization/group that has financed the project

Funding information is important for acknowledgement purposes.

For organizations, you must provide the name of the organization (organizationName) and, if possible, a website (website) and/or an identifier (OrganizationIdentifier).

Example

```
<ms:funder>
  <ms:Organization>
    <ms:actorType>Organization</ms:actorType>
    <ms:organizationName xml:lang="en">European Commission</
  <ms:organizationName>
    <ms:website>https://ec.europa.eu/info/index_en</ms:website>
  </ms:Organization>
</ms:funder>
```

fundingCountry

Path MetadataRecord.DescribedEntity.Project.fundingCountry

Data type CV (regionIdType)

Optionality Recommended

Explanations & Instructions

Specifies the name of the funding country, in case of national funding as mentioned in ISO3166

Example

```
<ms:fundingCountry>EU</ms:fundingCountry>
```

socialMediaOccupationalAccount

Path MetadataRecord.DescribedEntity.Project.socialMediaOccupationalAccount

Data type multilingual string

Optionality Recommended

Explanations & Instructions

Introduces the social media or occupational account details of a person, organization or project

You must also use the attribute `socialMediaAccountType` to specify the type of social media account. See `socialMediaOccupationalAccountType` for details.

Example

```
<ms:socialMediaOccupationalAccount ms:socialMediaOccupationalAccountType="http://w3id.
↪org/meta-share/meta-share/facebook">https://www.facebook.com/project</
↪ms:socialMediaOccupationalAccount>
```

LTArea

Path MetadataRecord.DescribedEntity.Project.LTArea

Data type component

Optionality Recommended

Explanations & Instructions

Introduces a Language Technology-related area that the project deals with

For details, see [LTArea](#) More specifically, you can fill in:

- the `LTClassRecommended` element with one of the recommended values from the [LT taxonomy](#), or
- the `LTClassOther` element with a free text.

Example

```
<ms:LTArea>
  <ms:LTClassRecommended>http://w3id.org/meta-share/omtd-share/
  ↪MachineTranslation</ms:LTClassRecommended>
</ms:LTArea>
<ms:LTArea>
  <ms:LTClassOther>Browser-based Machine Translation</ms:LTClassOther>
</ms:LTArea>
```

domain

Path MetadataRecord.DescribedEntity.Project.domain

Data type component

Optionality Recommended

Explanations & Instructions

Identifies a domain that the project deals with

You must fill in the CategoryLabel element with a free text value. If you prefer to add a value from an established controlled vocabulary, you can also use the DomainIdentifier (with the attribute DomainClassificationScheme with the appropriate value).

Example

```
<ms:domain>
  <ms:categoryLabel xml:lang="en">http://w3id.org/meta-share/omtd-share/
  ↪NewsMediaJournalismAndPublishing</ms:categoryLabel>
</ms:domain>
<ms:domain>
  <ms:categoryLabel xml:lang="en">General</ms:categoryLabel>
</ms:domain>
```

keyword

Path MetadataRecord.DescribedEntity.Project.keyword

Data type multilingual string

Optionality Recommended

Explanations & Instructions

Introduces a word or phrase considered important for the description of the project and thus used to index or classify it

Example

```
<ms:keyword xml:lang="en">Machine translation</ms:keyword>
<ms:keyword xml:lang="en">translation integration</ms:keyword>

<ms:keyword xml:lang="en">Language technology services</ms:keyword>
<ms:keyword xml:lang="en">Multilingualism</ms:keyword>
<ms:keyword xml:lang="en">Less-resourced languages</ms:keyword>
```

1.52.9 Minimal elements for organizations

This page describes the minimal metadata elements specific to **organizations**.

N.B. The interactive editor supports the full schema, i.e. it also includes optional elements.

1. Overview

Element name	Optionality	Tab
organizationName	M	Identity
OrganizationIdentifier	R	Identity
organizationShortName	R	Identity
organizationAlternativeName	R	Identity
organizationBio	R	Identity
logo	R	Identity
LTArea	R	Activities
serviceOffered	R	Activities
domain	R	Activities
keyword	R	Activities
email	R	Contact
website	R	Contact
headOfficeAddress	R	Contact
socialMediaOccupationalAccount	R	Contact
divisionCategory	R	Division
isDivisionOf	R	Division

2. Element presentation

In this section all the aforementioned elements are presented each one separately. The presentation follows the order of the elements in the table of the previous section.

Organization

Path MetadataRecord.DescribedEntity.Organization

Data type component

Optionality Mandatory

Explanation & Instructions

Wraps together elements for organizations

Example

```
<ms:Organization>
  <ms:entityType>organization</ms:entityType>
  ...
</ms:Organization>
```

organizationName

Path MetadataRecord.DescribedEntity.Organization.organizationName

Data type multilingual string

Optionality Mandatory

Explanation & Instructions

The full name of an organization

Example

```
<ms:organizationName xml:lang="en">Charles University</ms:organizationName>

<ms:organizationName xml:lang="en">Evaluation and Language Resources Distribution_
↪Agency</ms:organizationName>
```

OrganizationIdentifier

Path MetadataRecord.DescribedEntity.Organization.OrganizationIdentifier

Data type string

Optionality Recommended

Explanation & Instructions

A string (e.g., PID, internal to an organization, issued by the funding authority, etc.) used to uniquely identify an organization

You must also use the attribute `OrganizationIdentifierScheme` to specify the name of the scheme according to which an identifier is assigned to an organization by the authority that issues it. See [OrganizationIdentifierScheme](#) for details.

It is recommended to add an identifier issued by an authority, such as [GRID](#), if available.

Example

```
<ms:OrganizationIdentifier ms:OrganizationIdentifierScheme="http://w3id.org/meta-
↪share/meta-share/grid">https://www.grid.ac/institutes/grid.5216.0</
↪ms:OrganizationIdentifier>
```

organizationShortName

Path MetadataRecord.DescribedEntity.Organization.organizationShortName

Data type multilingual string

Optionality Recommended

Explanation & Instructions

Introduces the short name (abbreviation, acronym , etc.) used for an organization

Example

```
<ms:organizationShortName xml:lang="en">CUNI</ms:organizationName>
```

```
<ms:organizationShortName xml:lang="en">ELDA</ms:organizationName>
```

organizationAlternativeName

Path MetadataRecord.DescribedEntity.Organization.organizationAlternativeName

Data type multilingual string

Optionality Recommended

Explanation & Instructions

Introduces an alternative name (other than the short name) used for an organization

Example

```
<ms:organizationAlternativeName xml:lang="en">UNIVERZITA KARLOVA</ms:organizationAlternativeName>
```

```
<ms:organizationAlternativeName xml:lang="en">EVALUATIONS AND LANGUAGE RESOURCES DISTRIBUTION AGENCY</ms:organizationAlternativeName>
```

organizationBio

Path MetadataRecord.DescribedEntity.Organization.organizationBio

Data type multilingual string

Optionality Recommended

Explanation & Instructions

Introduces a short free-text account that provides information on an organization

Example

```
<ms:organizationBio xml:lang="en">Charles University was founded in 1348, making it one of the oldest universities in the world. Yet it is also renowned as a modern, dynamic, cosmopolitan and prestigious institution of higher education. It is the largest and most renowned Czech university, and is also the best-rated Czech university according to international rankings. There are currently 17 faculties at the University, plus 3 institutes, 6 other centres of teaching, research, development and other creative activities, a centre providing information services, 5 facilities serving the whole University, and the Rectorate - which is the executive management body for the whole University.</ms:organizationBio>
```

```
<ms:organizationBio xml:lang="en">The Evaluations and Language Resources Distribution Agency (ELDA), was created in 1995 as the organizational infrastructure with the mission of providing a central clearing house for Language Resources (LR) of the European Language Resources Association (ELRA). ELDA was set up to identify, classify, collect, validate and distribute the language resources that are needed by the Human Language Technology (HLT) community. Anticipating the evolutions in the HLT field, ELDA broadened its activities to cover multimedia/multimodal resources as well as evaluation activities, distributing the language resources needed for evaluation purposes, and conducting/coordinating evaluation campaigns. ELDA has played a significant role within the major Multimedia and Multimodal production projects that resulted in one of the most impressive collections of available data sets, embracing all aspects of Language Technologies. ELDA was also involved in evaluation initiatives, in several FPs' projects involving HLT infrastructures, as well as in national programmes. In addition to work on data production, processing and annotation, validation and quality control, several of
```


(continued from previous page)

logo*Path* `MetadataRecord.DescribedEntity.Organization.logo`*Data type* URL*Optionality* Recommended*Explanation & Instructions*

Links to a URL with an image file containing a symbol or graphic object used to identify the entity

In the interactive form, users can also upload an image file.

Example

```
<ms:logo>https://cuni.cz/UKEN-1-version1-afoto.jpg</ms:logo>

<ms:logo>https://www.european-language-grid.eu/wp-content/uploads/2019/03/logo___
↪consortium-elda.svg</ms:logo>
```

LTArea*Path* `MetadataRecord.DescribedEntity.Organization.LTArea`*Data type* component*Optionality* Recommended*Explanation & Instructions*

Introduces a Language Technology-related area that a person or organization is involved or active in

For details, see [LTArea](#) More specifically, you can fill in:

- the `LTClassRecommended` element with one of the recommended values from the [LT taxonomy](#), or
- the `LTClassOther` element with a free text.

Example

```
<ms:LTArea>
  <ms:LTClassRecommended>http://w3id.org/meta-share/omtd-share/
  ↪LanguageTechnology</ms:LTClassRecommended>
</ms:LTArea>
<ms:LTArea>
  <ms:LTClassRecommended>http://w3id.org/meta-share/omtd-share/
  ↪MachineTranslation</ms:LTClassRecommended>
</ms:LTArea>
```

serviceOffered

Path MetadataRecord.DescribedEntity.Organization.serviceOffered

Data type multilingual string

Optionality Recommended

Explanation & Instructions

Lists the service(s) offered by an organization or person

Example

```
<ms:serviceOffered xml:lang="en">Evaluation and benchmarking</ms:serviceOffered>
<ms:serviceOffered xml:lang="en">Legal support</ms:serviceOffered>
```

domain

Path MetadataRecord.DescribedEntity.Organization.domain

Data type component

Optionality Recommended

Explanation & Instructions

Identifies a domain that the organization deals with

You must fill in the `CategoryLabel` element with a free text value. If you prefer to add a value from an established controlled vocabulary, you can also use the `DomainIdentifier` (with the attribute `DomainClassificationScheme` with the appropriate value).

Example

```
<ms:domain>
  <ms:categoryLabel xml:lang="en">environment</ms:categoryLabel>
</ms:domain>
```

keyword

Path MetadataRecord.DescribedEntity.Organization.keyword

Data type multilingual string

Optionality Recommended

Explanation & Instructions

Introduces a word or phrase considered important for the description of the project and thus used to index or classify it

Example

```
<ms:keyword xml:lang="en">Research infrastructures</ms:keyword>
<ms:keyword xml:lang="en">Language Resources</ms:keyword>
<ms:keyword xml:lang="en">Digital Humanities</ms:keyword>
<ms:keyword xml:lang="en">Language Resources and Evaluation</ms:keyword>
<ms:keyword xml:lang="en">Legal support</ms:keyword>
<ms:keyword xml:lang="en">Data management</ms:keyword>
```

email

Path MetadataRecord.DescribedEntity.Organization.email

Data type string

Optionality Recommended

Explanation & Instructions

Points to the email address of a person, organization or group

Example

```
<ms:email>info@company.eu</ms:email>
```

website

Path MetadataRecord.DescribedEntity.Organization.website

Data type URL

Optionality Recommended

Explanation & Instructions

Links to a URL that acts as the primary page (like a table of contents) introducing information about an organization (e.g., products, contact information, etc.) or project

Example

```
<ms:website>https://www.cuni.cz</ms:website>  
<ms:website>http://www.elra.info/en/</ms:website>
```

headOfficeAddress

Path MetadataRecord.DescribedEntity.Organization.headOfficeAddress

Data type component

Optionality Recommended

Explanation & Instructions

Links to a set of elements that describe the full address of the head office of an or organization (i.e. including street address, zip code, etc.). The only mandatory element in this set is `country`.

Example

```
<ms:headOfficeAddress>
  <ms:address xml:lang="en">OLD COLLEGE, SOUTH BRIDGE</ms:address>
  <ms:zipCode>EH8 9YL</ms:zipCode>
  <ms:city xml:lang="en">EDINBURGH</ms:city>
  <ms:country>GB</ms:country>
</ms:headOfficeAddress>
```

socialMediaOccupationalAccount

Path MetadataRecord.DescribedEntity.Organization.socialMediaOccupationalAccount

Data type multilingual string

Optionality Recommended

Explanation & Instructions

Introduces the social media or occupational account details of a person or organization

You must also use the attribute `socialMediaAccountType` to specify the type of social media account. See https://european-language-grid.readthedocs.io/en/release1.1.2/Documentation/ELG-SHARE_xsd.html#socialMediaOccupationalAccountType for details.

Example

```
<ms:socialMediaOccupationalAccount ms:socialMediaOccupationalAccountType="http://w3id.
↪org/meta-share/meta-share/facebook">https://www.facebook.com/UFALMFFUK</
↪ms:socialMediaOccupationalAccount>
```

divisionCategory

Path MetadataRecord.DescribedEntity.Organization.divisionCategory

Data type CV

Optionality Recommended

Explanation & Instructions

Classifies the division of an organization according to a controlled vocabulary

Specify, in case the organization you describe is part of a parent organization, the category, e.g. faculty or department of a university, laboratory in a company, etc.

Example

```
<ms:divisionCategory>http://w3id.org/meta-share/meta-share/institute</
↪ms:divisionCategory>
```

isDivisionOf

Path MetadataRecord.DescribedEntity.Organization.isDivisionOf

Data type component

Optionality Recommended

Explanation & Instructions

Links an organization to the division(s) it consists of

Example

```
<ms:isDivisionOf>
  <ms:organizationName xml:lang="en">Charles University</ms:organizationName>
  <ms:website>https://www.cuni.cz</ms:website>
</ms:isDivisionOf>
```

1.53 Overview

This annex contains the specifications of ELG's internal and public application programming interfaces (APIs).

1.54 Internal LT Service API specification

Note: This specification details the API that LT tool containers need to implement in order to be runnable as functional services within the ELG infrastructure. This is distinct from (though closely related to) the public-facing service execution API that outside users use to send requests to ELG services - the *public APIs* are documented separately.

Contents

- *Internal LT Service API specification*
 - *Basic API pattern*
 - *Utility datatypes*
 - * *Status message*
 - * *Annotations*
 - *Request structure*
 - * *Text requests*
 - * *Structured text request*
 - * *Audio requests*
 - * *A note about parameters*
 - *Response structure*
 - * *Failure message*
 - * *Successful response message*

- * *Annotations response*
- * *Classification response*
- * *Texts response*
- * *Audio response*
- *Progress Reporting*
- *Appendix: Standard status message codes*

Where possible, this document SHOULD use the MUST/SHOULD/MAY terms from [RFC 2119](#) to indicate requirement levels.

1.54.1 Basic API pattern

In order to integrate an LT tool as a functional service in the ELG infrastructure, the tool MUST offer at least one endpoint that can accept HTTP (1.1 or 2 - preferably cleartext HTTP/2) POST requests conforming to the appropriate request schema, and return an appropriate response as `application/json`. This specification also details a response pattern based on Server-Sent Events (SSE, a protocol defined as [part of HTML5](#)) that long-running tools can use to report progress information - support for this mechanism is RECOMMENDED for all tools but not required.

Endpoints may be sent multiple parallel requests by the ELG platform, and there is no requirement that a service must respond to requests in any particular order - certain services may, for example, be more efficient if they can batch up several requests into one back end process (e.g. for GPU computing) and send the responses in one go. If a tool has limits on the number of concurrent requests a single instance can handle then this information should be supplied to the ELG platform administrators as part of the on-boarding process, so the platform can use this data to decide how to scale the pod replicas to match the level of load on the service at any given time.

Where a tool already has its own native HTTP API it may be more convenient for integrators to provide a separate *service adapter* image which can handle requests matching the ELG specification and transform them into calls on the tool's native API. The tool container and the adapter container will run within the same “pod” in Kubernetes and can access each other as `localhost`.

1.54.2 Utility datatypes

The following JSON structures are used in several places in this specification, they are documented here to avoid duplication.

Status message

Since the ELG is supposed to be a multilingual platform, error and other status messages are handled using an approach modelled on the `i18n` mechanism from the [Spring Framework](#) - the message is represented by a *code*, along with a template *text* with numbered placeholders that are zero-based indices into an array of *params* replacement values.

```
{
  "code": "elg.example.no.translation",
  "text": "Default text to use for the {0} if no {1} can be found",
  "params": ["message", "translation"],
  "detail": {
    // arbitrary further details that don't need translation,
    // such as a stack trace, service-native error code, etc.
  }
}
```

ELG provides a common library of fully-translated message codes for service developers to use, as detailed below - developers are free to use their own codes in their own namespaces (i.e. not prefixed `elg.`) on the understanding that it is their responsibility to provide translations. A mechanism for developers to contribute their translated messages to the platform is under development but not yet generally available.

Annotations

Many of the request and response types need to represent *annotations* - pieces of metadata about specific parts of a text or audio data stream, rather than about the stream as a whole. For example, a named entity recogniser might want to state that characters 10 to 15 in the request text represent the name of a female person, or a speech recogniser might want to state that characters 75 to 80 in the transcription represent a word, and map to the time period 1.37 to 1.6 seconds in the source audio. Such structures are represented in a consistent way across all the ELG API messages:

```
"annotations":{
  "<annotation type>":[
    {
      "start":number,
      "end":number,
      "sourceStart":number,
      "sourceEnd":number,
      "features":{ /* arbitrary JSON */ }
    }
  ]
}
```

The `<annotation type>` is an arbitrary string representing the type of annotation, e.g. “Person” or “Word” in the examples above. For each type of annotation, the matching value is a JSON *array* of objects, each object representing one annotation of that type. Note that when generating these structures in your API responses the value here **MUST** be an array even if there is only one annotation of the relevant type - some JSON generation libraries “unwrap” singleton arrays by default. The properties of each annotation object are:

start and end The position of the annotation in the main data stream to which it refers - this is typically the content directly associated with this `annotations` structure (for example the text of a translation). When the stream is text these would be Unicode character offsets from the start of the text, for audio they would typically be time points in seconds, etc. Subtracting the start value from the end value should give the length of the annotated area - there are several equivalent ways to conceptualise this, for example with text you could consider the characters as numbered from zero with the start offset *inclusive* and the end offset *exclusive*, or you could consider the offsets to represent the positions *between* characters (so 0 is before the first character, 1 is between the first and second, etc.).

sourceStart and sourceEnd Where these annotations are relative to a data stream that has been generated from another “source” data stream (e.g. a translation of text in another language, or a transcription of audio), these properties can be optionally used to link to the positions in the source stream (e.g. to align words in the translation with words in the original).

features Arbitrary JSON representing other properties of the annotation, e.g. a “Person” annotation might have a feature for “gender”, a “Word” from a morphological analyser might have “root” and “suffix”, etc.

1.54.3 Request structure

There are two main types of endpoint currently supported for this specification, one for services whose input is structured or unstructured *text* and one for services whose input is *audio*.

Text requests

Services that take plain text (or something from which plain text can be extracted, e.g. HTML) as their input are expected to offer an endpoint that accepts POST requests with `Content-Type: application/json` that conforms to the following structure.

```
{
  "type": "text",
  "params": { ... }, /* optional */
  "content": "The text of the request",
  // mimeType optional - this is the default if omitted
  "mimeType": "text/plain",
  "features": { /* arbitrary JSON metadata about this content, optional */ },
  "annotations": { /* optional */
    "<annotation type>": [
      {
        "start": number,
        "end": number,
        "features": { /* arbitrary JSON */ }
      }
    ]
  }
}
```

We expect that across the ELG from amongst the large number of possible and supported document types, a set of a smaller number of document types will emerge as being preferred and well supported (for example, plain text, HTML, XML - we do not intend to support binary formats such as PDF or Word as “text” requests, but may introduce other formats to this specification at a later date).

The only part of this request that is guaranteed to be present is the `type` (which will always be “text”) and the `content`. So a minimal request would look like this:

```
{ "type": "text", "content": "This is an example request" }
```

The optional elements are:

mimeType the MIME type of the content, if it is not simply plain text

params vendor-specific parameters - it is up to the individual service implementor to decide how (or indeed whether) to interpret these

features metadata about the input *as a whole*

annotations *as described above* - the `start` and `end` are Unicode character offsets within the `content` and the `sourceStart` and `sourceEnd` are ignored.

Tools that are able to accept `text` requests are RECOMMENDED to also offer an endpoint that can accept just the plain text (or other types of) “content” posted directly, and treat that the same as they would a message with the “content” property equal to the post data, the “mimeType” taken from the request `Content-Type` header, and no features or annotations. The “params” should be populated from the URL query string parameters. This endpoint will not be called by the ELG platform internally but it will make the service easier to test outside of the ELG platform infrastructure, and for open-source tools it will allow users to easily download and run the tool locally in Docker on their own hardware.

Structured text request

This is very similar to the plain text request, but for services that require some structure to their input, for example a list of sentences for some MT services, a list of words for a service that re-segments a stream of ASR output into a list of sentences, etc. Again, services that accept this kind of input should provide a POST endpoint that accepts Content-Type: application/json conforming to the following structure:

```
{
  "type": "structuredText",
  "params": { ... }, /* optional */
  "texts": [
    {
      "content": "The text of this node", // either
      "texts": [ /* same structure, recursive */ ], // or
      // mimeType optional - this is the default if omitted
      "mimeType": "text/plain",
      "features": { /* arbitrary JSON metadata about this node, optional */ },
      "annotations": { /* optional */
        "<annotation type>": [
          {
            "start": number,
            "end": number,
            "features": { /* arbitrary JSON */ }
          }
        ]
      }
    }
  ]
}
```

The type will always be “structuredText”, params (optional) allows for vendor-specific parameters whose interpretation is up to the individual service implementor, and texts will always be an array of at least one JSON object. The texts property forms a recursive tree-shaped data structure, each object will be either a *leaf node* containing a piece of content or a *branch node* containing another list of texts.

Leaf nodes have one required property content containing the text of this node, plus zero or more of the following optional properties:

mimeType the MIME type of the content, if it is not simply plain text

features metadata about this node as a whole

annotations *as described above* - the start and end are Unicode character offsets within the content and the sourceStart and sourceEnd are ignored.

Branch nodes have one required property texts containing an array of child nodes (which may in turn be branch or leaf nodes), plus zero or more of the following optional properties:

features metadata about this node as a whole

annotations *as described above* - the start and end are array offsets within the texts array (e.g. "start":0, "end":2 would refer to the first and second children - treat them as zero-based array indices where the start is *inclusive* and the end is *exclusive*) and the sourceStart and sourceEnd are ignored.

Here is the simplest possible example of a structured text request representing two sentences, each with several words, with no features and no annotations.

```
{
  "type": "structuredText",
  "texts": [
```

(continues on next page)

(continued from previous page)

```

{
  "texts": [
    { "content": "The" }, { "content": "European" }, { "content": "Language" }, { "content":
↪ "Grid" }
  ],
  {
    "texts": [
      { "content": "An" }, { "content": "API" }, { "content": "example" }
    ]
  }
]
}

```

Audio requests

Services that accept *audio* as input (e.g. speech recognition) are slightly more complex, given the input data cannot be easily encoded directly in JSON. Audio services must accept a POST of Content-Type: `multipart/form-data` with two parts, the first part named “request” will be `application/json` conforming to the following structure, and the second part named “content” will be `audio/x-wav` or `audio/mpeg` containing the actual audio data.

```

{
  "type": "audio",
  "params": { ... }, // optional
  "format": "string", // LINEAR16 for WAV or MP3 for MP3, other types are service_
↪ specific
  "sampleRate": number,
  "features": { /* arbitrary JSON metadata about this content, optional */ },
  "annotations": { /* optional */
    "<annotation type>": [
      {
        "start": number,
        "end": number,
        "features": { /* arbitrary JSON */ }
      }
    ]
  }
}

```

The ELG platform typically expects audio to be a single channel - this is not guaranteed, as it depends what the requesting user submits, and a service receiving multiple audio channels may handle this situation in any way it sees fit including processing only the first channel or mixing down the multi-channel stream to mono before processing.

As with text requests we expect that there will be a small number of standard audio formats that are well supported across services (e.g. 16kHz uncompressed WAV) but individual services may support other types. The format and sample rate parameters may be ignored if the audio is in a format with a self-describing file header (e.g. WAV) which specifies other values.

Optional properties of this request type are:

params vendor-specific parameters - it is up to the individual service implementor to decide how (or indeed whether) to interpret these

features metadata about the input *as a whole*

annotations *as described above* - the `start` and `end` are floating point timestamps in seconds from the start of the audio and the `sourceStart` and `sourceEnd` are ignored.

A note about parameters

All request types include an optional `params` section allowing the service to take vendor-specific parameters. While this is specified as accepting any JSON, in practice the public API will send all parameters as either a single strings (if the parameter has a single value) or an array of strings (if the parameter has multiple values). If a service requires numeric or boolean parameters then it should be written to accept string values as well as proper numeric or boolean literals in the JSON, and parse the string value to the appropriate type if possible rather than simply responding with a “type mismatch” error.

1.54.4 Response structure

Services are expected to return their responses as JSON as described in the rest of this document. The minimal requirement is for services to be able to respond with `Content-Type: application/json` containing a successful or failed response message, but long-running services may also choose to offer `Content-Type: text/event-stream` to be able to stream progress reports during processing of the request. This mechanism is described at the end of this document.

Failure message

If processing fails for any reason (whether due to bad input, overloading of the service, or internal errors during processing) then the service should return the following JSON structure to describe the failure.

```
{
  "failure":{
    "errors":[array of status messages]
  }
}
```

The `errors` property is an array of *i18n status messages* (JSON objects with properties “code”, “text” and “params”) as described above - standard message codes are given in the appendix to this document.

Successful response message

All the successful responses follow this basic format:

```
{
  "response":{
    "type":"Response type code",
    "warnings":[/* array of status messages, optional*/,
      // other properties type-specific
    ]
  }
}
```

As with the request, the response type code will likely be constant for any given service. The exact format of rest of a successful response message depends on the type of the service.

The `warnings` list is a slot to report warning messages that did not cause processing to fail entirely but may need to be fed back to the user (e.g. if the process involves several independent steps and only some of the steps failed, or the input was too long and the service chose to truncate it rather than fail altogether). Again, the individual messages in this array are *i18n status messages* as described above.

Annotations response

This response is suitable for any service that returns standoff annotations that are anchored to locations in text (e.g. named entity recognition) or time points in an audio/video stream (in general: anything compatible with a 1-dimensional coordinate system that uses a single number).

```
{
  "response":{
    "type":"annotations",
    "warnings":[...], /* optional */
    "features":{...}, /* optional */
    "annotations":{
      "<annotation type>":[
        {
          "start":number,
          "end":number,
          "features":{ /* arbitrary JSON */ }
        }
      ]
    }
  }
}
```

features (optional) metadata about the input *as a whole*

annotations (required, but may be empty "annotations": {}) *as described above* - for plain text data start and end would be character offsets into the text (Unicode code points), for audio data they would be the time point within the audio in seconds. The sourceStart and sourceEnd are ignored since there are no separate “source” and “target” data streams in this situation.

Classification response

For document-level (or more generally whole-input-level) classification services, e.g. language identification

```
{
  "response":{
    "type":"classification",
    "warnings":[...], /* optional */
    "classes":[
      {
        "class":"string",
        "score":number /* optional */
      }
    ]
  }
}
```

We allow for zero or more classifications, each with an optional score. Services should return multiple classes in whatever order they feel is most useful (e.g. “most probable class” first), this order need not correspond to a monotonic ordering by score - we don’t assume scores are all mutually comparable - and the order will be preserved by any subsequent processing steps.

Classification tools that classify *segments* of the input rather than the whole input should use the annotations or texts response formats instead of this one.

Texts response

A response consisting of one or more *new* texts with optional annotations, for example multiple alternative possible translations from an MT service or transcriptions from an ASR service.

```
{
  "response":{
    "type":"texts",
    "warnings":[...], /* optional */
    "texts":[
      {
        "role":"string", /* optional */
        "content":"string of translated/transcribed text", // either
        "texts":["same structure, recursive *//],          // or
        "score":number, /* optional */
        "features":{" /* arbitrary JSON, optional */ },
        "annotations":{" /* optional */
          "<annotation type>":[
            {
              "start":number,
              "end":number,
              "sourceStart":number, // optional
              "sourceEnd":number,   // optional
              "features":{" /* arbitrary JSON */ }
            }
          ]
        }
      }
    ]
  }
}
```

As with the structured text request format above, this texts response structure is recursive, so it is possible for each object in the list to be a branch node containing a set of child texts or a leaf node containing a single string.

Leaf nodes have one required property `content`, plus zero or more of the following optional properties:

role the role of this node in the response, “alternative” if it represents one of a list of alternative translations/transcriptions, “segment” if it represents a segment of a longer text, or “paragraph”, “sentence”, “word” etc. for specific types of text segment.

score if this is one of a list of alternatives, each alternative may have a score representing the quality of the alternative

features metadata about this node as a whole

annotations *as described above* - the `start` and `end` are Unicode character offsets within the `content` and the `sourceStart` and `sourceEnd` are the offsets into the source data (the interpretation depends on the nature of the source data).

Branch nodes have one required property `texts` containing an array of child nodes (which may in turn be branch or leaf nodes), plus zero or more of the following optional properties:

role the role of this node in the response, “alternative” if it represents one of a list of alternative translations/transcriptions, “segment” if it represents a segment of a longer text, or “paragraph”, “sentence”, “word” etc. for specific types of text segment.

features metadata about this node as a whole

annotations *as described above* - the `start` and `end` are array offsets within the `texts` array (e.g. `"start":0, "end":2` would refer to the first and second children - treat them as zero-based array indices

where the start is *inclusive* and the end is *exclusive*) and the `sourceStart` and `sourceEnd` are the offsets into the source data (the interpretation depends on the nature of the source data).

The texts response type will typically be used in two different ways, either

- the top-level list of texts is interpreted as a set of *alternatives* for the whole result - in this case we would expect the `content` property to be populated but not the `texts` one, and a “role” value of “alternative” - tools should return the alternatives in whatever order they feel is most useful, typically descending order of likelihood (though as for classification results we don’t assume scores are mutually comparable and the order of alternatives in the array need not correspond to a monotonic ordering by score).
- the top-level list of texts is interpreted as a set of *segments* of the result, where each segment can have N-best alternatives (e.g. a list of sentences, with N possible translations for each sentence). In this case we would expect `texts` to be populated but not `content`, and a “role” value of either “segment” or something more detailed indicating the nature of the segmentation such as “sentence”, “paragraph”, “turn” (for speaker detection), etc. - in this case the order of the texts should correspond to the order of the segments in the result.

Audio response

A response consisting of a piece of audio (e.g. an audio rendering of text in a text-to-speech tool), optionally with annotations linked to either or both of the source and target data.

```
{
  "response": {
    "type": "audio",
    "warnings": [...], /* optional */
    "content": "base64 encoded audio for shorter snippets",
    "format": "string",
    "features": { /* arbitrary JSON, optional */ },
    "annotations": {
      "<annotation type>": [
        {
          "start": number,
          "end": number,
          "sourceStart": number, // optional
          "sourceEnd": number,  // optional
          "features": { /* arbitrary JSON */ }
        }
      ]
    }
  }
}
```

Here the `content` property contains base64-encoded audio data, and the `format` specifies the audio format used - in this version of the ELG platform the supported formats are `LINEAR16` (uncompressed WAV) or `MP3`. In addition the response may contain zero or more of the following optional properties:

features metadata about this node as a whole

annotations *as described above* - the `start` and `end` are time offsets within the audio `content` expressed as floating point numbers of seconds, and the `sourceStart` and `sourceEnd` are the offsets into the source data (the interpretation depends on the nature of the source data).

As an alternative to embedding the audio data in base64 encoding within the JSON payload, a service MAY simply return the audio data directly with the appropriate `Content-Type` (`audio/x-wav` or `audio/mpeg`), however this approach means the service will be unable to return features or annotations over the audio, and will be unable to report partial progress.

1.54.5 Progress Reporting

Some LT services can take a long time to process each request, and in these cases it may be useful to be able to send intermediate progress reports back to the caller. This serves both to reassure the caller that processing has not silently failed, and also to ensure the HTTP connection is kept alive. The mechanism for this in ELG leverages the standard “Server-Sent Events” (SSE) protocol format - *if* the client sends an `Accept` header that announces that it is able to understand the `text/event-stream` response type, then the service may choose to *immediately* return a 200 “OK” response with `Content-Type: text/event-stream` and hold the connection open (using chunked transfer encoding in HTTP/1.1 or simply not sending a `Content-Length` in HTTP2). It may then dispatch zero or more SSE “events” with JSON data in the following structure:

```
{
  "progress":{
    "percent"://number between 0.0 and 100.0,
    "message":{
      // optional status message, with code, text and params as above
    }
  }
}
```

followed by *exactly one* successful or failed response in the usual format. Services should not send any further progress messages once the success or failure response has been sent. Note that if a message is provided in a progress report it must be an *18n status message*, not simply a plain string.

For example:

```
Content-Type: text/event-stream

data:{"progress":{"percent":0.0}}

data:{"progress":{"percent":20.0}}

data:{"progress":{
data:    "percent":70.0
data:  }
data:}

data:{"response":{...}}
```

As per the SSE specification, *each line* of data within an event is prefixed `data:`, and an event is terminated by a blank line - there **MUST** be two consecutive newlines or CRLF sequences between the end of one event and the start of the next.

One would normally expect the progress percentage to increase over time but this is not necessarily a requirement of the specification - services are free to publish progress messages *without* a “percent” property if they wish to provide a status update message but cannot quantify their progress numerically, or even with a lower percentage than the previous message if they now have information to suggest that the overall process will take longer than first estimated.

Services are **RECOMMENDED** to support this response format, and to send it if the client indicates they can accept `text/event-stream`, but it is not required. The clients which will call your services within the ELG infrastructure will accept both `text/event-stream` and `application/json` responses, and you are encouraged to return an event stream if you can, but you are free to return `application/json` if it makes more sense for your service, and you **MUST** return `application/json` if the calling client does not indicate in the `Accept` header that they can understand `text/event-stream`.

1.54.6 Appendix: Standard status message codes

```
#
#   Copyright 2019 The European Language Grid
#
#   Licensed under the Apache License, Version 2.0 (the "License");
#   you may not use this file except in compliance with the License.
#   You may obtain a copy of the License at
#
#       http://www.apache.org/licenses/LICENSE-2.0
#
#   Unless required by applicable law or agreed to in writing, software
#   distributed under the License is distributed on an "AS IS" BASIS,
#   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
#   See the License for the specific language governing permissions and
#   limitations under the License.
#
# This file contains the standard ELG status messages, translations should
# be placed in files named elg-messages_LANG.properties alongside this file.
#

# general bad request errors
elg.request.invalid=Invalid request message
elg.request.missing=No request provided in message
elg.request.type.unsupported=Request type {0} not supported by this service
elg.request.property.unsupported=Unsupported property {0} in request

elg.request.too.large=Request size too large

# Errors specific to text requests
elg.request.text.mimeType.unsupported=MIME type {0} not supported by this service

# Errors specific to audio requests
elg.request.audio.format.unsupported=Audio format {0} not supported by this service
elg.request.audio.sampleRate.unsupported=Audio sample rate {0} not supported by this_
↳service

# Errors specific to structured text requests
elg.request.structuredText.property.unsupported=Unsupported property {0} in "texts"_
↳of structuredText request

# General bad response errors
elg.response.invalid=Invalid response message
elg.response.type.unsupported=Response type {0} not supported

# Unknown property in response
elg.response.property.unsupported=Unsupported property {0} in response
elg.response.texts.property.unsupported=Unsupported property {0} in "texts" of texts_
↳response
elg.response.classification.property.unsupported=Unsupported property {0} in "classes
↳" of classification response

# User requested a service that does not exist
elg.service.not.found=Service {0} not found

# generic internal error when there's no more specific option
elg.service.internalError=Internal error during processing: {0}
```


1.55 Public LT API specification

LT services can be called via the API endpoints given on the “code samples” page, the format of the various requests and responses is closely related to the internal LT service API used within the ELG infrastructure, but the public endpoints also offer shortcuts to simplify common interactions. There are three different types of endpoint depending on the kind of data required by the service as input - *flat text*, *structured text* or *audio*.

Authentication to all endpoints is by the use of an OAuth2 Bearer Token, and a token suitable for test use can be copied from the “code samples” page. Future versions of this document will include details of how to obtain and renew access tokens programmatically. The token is passed via the HTTP Authorization header in the usual way: Authorization: Bearer <tokenValue>

1.55.1 Input formats

Services that process flat text

`https://{domain}/execution/processText/{ltServiceID}`

Services that process a single flat stream of text can be called via an endpoint of this form. Make an HTTP POST request to the endpoint with one of the following Content-Type headers:

application/json A JSON object as described in the “*text request*” section of the LT Service API specification. For example { "type": "text", "content": "The text to process", "params": { "genre": "news" } }. The type *must* be the string “text”, the content is the text to be processed, and params are specific to the individual service - see the per-service documentation for details of any parameters the service accepts.

text/plain or text/html Just the text to be processed. In this case any URL query parameters added to the endpoint URL will be passed on to the service as params

Services that process “structured” text

`https://{domain}/execution/processStructured/{ltServiceID}`

Some services require text that has been pre-segmented in some way, for example split into tokens, sentences or paragraphs. For this endpoint the following Content-Type values are supported:

application/json A JSON object as described in the “*structured text request*” section of the LT Service API specification. For example { "type": "structuredText", "texts": [{ "content": "First sentence." }, { "content": "Second sentence." }] }. As with text requests above, you may also add params to the JSON, these are specific to the individual service - see the per-service documentation for details of any parameters the service accepts.

text/plain As a convenience shortcut the endpoint can also accept a POST of plain text. In this case, how the text is segmented depends on a URL query parameter `split`

processStructured/{service} (without a split parameter) the whole text is treated as a single segment

processStructured/{service}?split=line the text is divided at line breaks, and each line is treated as a separate segment. Leading or trailing white space on each line is *not* trimmed, and blank lines become empty segments { "content": "" }

processStructured/{service}?split=paragraph the text is divided at each run of one or more *blank lines* (i.e. two or more consecutive line breaks, possibly with white space in between). Again, leading or trailing whitespace around each segment is *not* trimmed.

All query parameters (including split) are passed on to the underlying service.

Services that process audio

`https://{domain}/execution/processAudio/{ltServiceID}`

Services that process a stream of audio can be called via an endpoint of this form. Make an HTTP POST request whose body is the audio data, with an appropriate `Content-Type`: `audio/mpeg` for MP3 audio or `Content-Type: audio/x-wav` for uncompressed WAV audio.

Any URL query parameters added to the endpoint will be passed on to the service, see the per-service documentation for details of which (if any) parameters the service accepts.

A note about parameters

As noted above, when calling the “plain” endpoints that accept raw text or audio data, parameters may be passed to the service via the URL query string. The parameters (if any) accepted by a particular service are detailed in the per-service documentation. Query parameters are mapped into the JSON structure expected by the service container as follows:

- A **single** value for a parameter `paramName=paramValue` will be passed to the service as a single string `{"paramName": "paramValue"}`
- **Multiple** values for the same parameter `paramName=value1¶mName=value2` will be passed as an array `{"paramName": ["value1", "value2"]}`
- In order to force the array format even when passing a single value, append `[]` to the parameter name (note that these characters must be escaped in the query string): `param%5B%5D=value` will be passed as `{"param": ["value"]}`

1.55.2 Service responses

The response formats returned from service calls are identical to *their counterparts in the internal LT Service API* and will not be repeated here. However there is one shortcut for services such as text-to-speech that return audio data. Ordinarily these services return a response of `Content-Type: application/json` including the audio data encoded in base64, but if you supply a parameter `audioOnly` (in the `params` for a JSON request, or as a URL query parameter for an unwrapped text/HTML/audio request) with the value “true” or “yes”, then instead of receiving the full JSON response you will receive just the binary audio data with an appropriate `Content-Type` of `audio/mpeg` or `audio/x-wav`.

Failed responses return a special type of response as follows:

```
{
  "failure": {
    "errors": [array of status messages]
  }
}
```

The `errors` property is an array of *internationalization-compatible status message objects* - the ELG platform provides another endpoint `https://{domain}/i18n/resolve` to which you can POST a JSON array of these objects and receive an array of resolved message strings in response.

1.55.3 Asynchronous processing

Some services may take several seconds or more to respond, either because their processing is naturally complex or because there are many requests for the same service being processed at the same time. To avoid the risk of dropped connections in such cases, the ELG platform offers an alternative “asynchronous” interaction style. To use this, send the same POST request, but add `/async` to the endpoint URL ahead of the `/process*`, e.g.

```
https://{domain}/execution/async/processAudio/{ltServiceID}
```

When called in async mode, the initial request should return immediately with a response of the following form:

```
{
  "response": {
    "type": "stored",
    "uri": "<polling URL>"
  }
}
```

The `uri` property is a URL which you should then begin to poll on a regular basis with a GET request (using the same Authorization token). Each time you poll, if processing is still ongoing you will receive a “progress” response of the form

```
{
  "progress": {
    "percent": //number between 0.0 and 100.0,
    "message": {
      // optional status message
    }
  }
}
```

(The message is optional, if provided it is a message *object* as in the failure response case above, which can be resolved to a message *string* by the `/i18n/resolve` endpoint). Some services return true progress percentages, for those that do not provide real updates the endpoint will always return `{"progress": {"percent": 0.0}}` to show that processing is still ongoing.

Once the processing is complete the poll URL will return the JSON response (successful or failed) exactly as you would have got from the normal synchronous API endpoint.

1.56 Terms of use

Use of the European Language Grid is subject to its [terms of use](#).

1.57 Publications and reports

This annex contains research papers, reports and other documents that describe various aspects of the work carried out in the ELG project in detail.

1.57.1 Scientific publications

If you'd like to refer to the **European Language Grid initiative and platform** in a general way, please cite the following article.

Georg Rehm, Maria Berger, Ela Elsholz, Stefanie Hegele, Florian Kintzel, Katrin Marheinecke, Stelios Piperidis, Miltos Deligiannis, Dimitris Galanis, Katerina Gkirtzou, Penny Labropoulou, Kalina Bontcheva, David Jones, Ian Roberts, Jan Hajic, Jana Hamrlová, Lukáš Kačena, Khalid Choukri, Victoria Arranz, Andrejs Vasiljevs, Orians Anvari, Andis Lagzdīņš, Jūlija Melņika, Gerhard Backfried, Erinc Dikici, Miroslav Janosik, Katja Prinz, Christoph Prinz, Severin Stampfer, Dorothea Thomas-Aniola, José Manuel Gómez Pérez, Andres Garcia Silva, Christian Berrío, Ulrich Germann, Steve Renals, Ondrej Klejch. [European Language Grid: An Overview](#). In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Christopher Cieri, Khalid Choukri, Thierry Declerck, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, [Proceedings of the 12th Language Resources and Evaluation Conference \(LREC 2020\)](#), pages 3359-3373, Marseille, France, 2020. European Language Resources Association (ELRA). [\[.bib | .pdf\]](#)

If you'd like to refer to the **metadata schema developed for and used in the European Language Grid platform**, please cite the following article.

Penny Labropoulou, Katerina Gkirtzou, Maria Gavriilidou, Miltos Deligiannis, Dimitris Galanis, Stelios Piperidis, Georg Rehm, Maria Berger, Valérie Mapelli, Michael Rigault, Victoria Arranz, Khalid Choukri, Gerhard Backfried, José Manuel Gómez Pérez, and Andres Garcia-Silva. [Making Metadata Fit for Next Generation Language Technology Platforms: The Metadata Schema of the European Language Grid](#). In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Christopher Cieri, Khalid Choukri, Thierry Declerck, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, [Proceedings of the 12th Language Resources and Evaluation Conference \(LREC 2020\)](#), pages 3421-3430, Marseille, France, 2020. European Language Resources Association (ELRA). [\[.bib | .pdf\]](#)

If you'd like to refer to a current description of the **situation of the wider Multilingual Europe community**, please cite the following article.

Georg Rehm, Katrin Marheinecke, Stefanie Hegele, Stelios Piperidis, Kalina Bontcheva, Jan Hajic, Khalid Choukri, Andrejs Vasiljevs, Gerhard Backfried, Christoph Prinz, José Manuel Gómez Pérez, Luc Meertens, Paul Lukowicz, Josef van Genabith, Andrea Lösch, Philipp Slusallek, Morten Irgens, Patrick Gatellier, Joachim Köhler, Laure Le Bars, Dimitra Anastasiou, Albina Aukšoriūtė, Núria Bel, António Branco, Gerhard Budin, Walter Daelemans, Koenraad De Smedt, Radovan Garabík, Maria Gavriilidou, Dagmar Gromann, Svetla Koeva, Simon Krek, Cvetana Krstev, Krister Lindén, Bernardo Magnini, Jan Odijk, Maciej Ogrodniczuk, Eiríkur Rögnvaldsson, Mike Rosner, Bolette Pedersen, Inguna Skadina, Marko Tadić, Dan Tufiş, Tamás Váradi, Kadri Vider, Andy Way, and François Yvon. [The European Language Technology Landscape in 2020: Language-Centric and Human-Centric AI for Cross-Cultural Communication in Multilingual Europe](#). In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Christopher Cieri, Khalid Choukri, Thierry Declerck, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, [Proceedings of the 12th Language Resources and Evaluation Conference \(LREC 2020\)](#), pages 3315-3325, Marseille, France, 2020. European Language Resources Association (ELRA). [\[.bib | .pdf\]](#)

Proceedings of the 1st International Workshop on Language Technology Platforms

Georg Rehm, Kalina Bontcheva, Khalid Choukri, Jan Hajic, Stelios Piperidis, and Andrejs Vasiljevs, editors. [Proceedings of the 1st International Workshop on Language Technology Platforms \(IWLTP 2020\)](#), co-located with LREC 2020), Marseille, France, 2020. 16 May 2020. [\[.bib | .pdf\]](#)

If you'd like to refer to our initial thoughts and plans **how to make a number of European AI platforms interoperable**, please cite the following article.

Georg Rehm, Dimitrios Galanis, Penny Labropoulou, Stelios Piperidis, Martin Weiß, Ricardo Usbeck, Joachim Köhler, Miltos Deligiannis, Katerina Gkirtzou, Johannes Fischer, Christian Chiarcos, Nils Feldhus, Julián Moreno-Schneider, Florian Kintzel, Elena Montiel, Víctor Rodríguez Doncel, John P. McCrae, David Laqua, Irina Patricia Theile, Christian Dittmar, Kalina Bontcheva, Ian Roberts, Andrejs Vasiljevs, and Andis Lagzdīņš. [Towards an Interoperable Ecosystem of AI and LT Platforms: A Roadmap for the Implementation of Different Levels of Interoperability](#).

In Georg Rehm, Kalina Bontcheva, Khalid Choukri, Jan Hajić, Stelios Piperidis, and Andrejs Vasiljevs, editors, *Proceedings of the 1st International Workshop on Language Technology Platforms (IWLTP 2020, co-located with LREC 2020)*, pages 96-107, Marseille, France, 2020. 16 May 2020. [.bib | .pdf]

1.57.2 Related publications

Andrus Ansip. *How multilingual is Europe’s Digital Single Market?*. 2016.

Communication from the commission to the European Parliament, the council, the European economic and social committee and the committee of the regions. *A Digital Single Market Strategy for Europe*. COM (Communication), 192. Brussels, Belgium, 2015.

European Parliament. *Report on language equality in the digital age..* Jill Evans, rapporteur, Committee on Culture and Education (CULT), Committee on Industry, Research and Energy (ITRE). Strasbourg, France, 2018.

Eurostat. *Internet access and use statistics – households and individuals..* 2016.

Andreas Kornai. *Digital Language Death*. In PLoS ONE, volume 8, 2013.

Georg Rehm and Stefanie Hegele. *Language Technology for Multilingual Europe: An Analysis of a Large-Scale Survey regarding Challenges, Demands, Gaps and Needs*. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odić, Stelios Piperidis, Takenobu Tokunaga, editors, *Proceedings of the 11th Language Resources and Evaluation Conference (LREC 2018)*, pages 3282–3289, Miyazaki, Japan, 2018. European Language Resources Association (ELRA). [.bib | .pdf]

Georg Rehm, Hans Uszkoreit, editors. *META-NET White Paper Series: Europe’s Languages in the Digital Age*, 32 volumes on 31 European languages. 2012. Springer, Heidelberg.

Georg Rehm and Hans Uszkoreit, editors. *The META-NET Strategic Research Agenda for Multilingual Europe 2020*. Dodrecht, New York, London, 2013. Springer, Heidelberg.

Georg Rehm, Hans Uszkoreit, Ido Dagan, Vartkes Goetcherian, Mehmet Ugur Dogan, Coskun Mermer, Tamás Váradi, Sabine Kirchmeier-Andersen, Gerhard Stickel, Meirion Pryds Jones, Stefan Oeter and Sigve Gramstad. *An Update and Extension of the META-NET Study “Europe’s Languages in the Digital Age”*. In Laurette Pretorius, Claudia Soria, Paola Baroni, editors, *Proceedings of the Workshop on Collaboration and Computing for Under-Resourced Languages in the Linked Open Data Era (CCURL 2014)*, pages 30–37, Reykjavik, Iceland, 2014.

Georg Rehm, Jan Hajić, Josef van Genabith and Andrejs Vasiljevs. *Fostering the Next Generation of European Language Technology: Recent Developments – Emerging Initiatives – Challenges and Opportunities*. Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odić, Stelios Piperidis, editors, *Proceedings of the 10th Language Resources and Evaluation Conference (LREC 2016)*, pages 1586–1592, Portorož, Slovenia, 2016. European Language Resources Association (ELRA). [.bib | .pdf]

Rehm, G., editor. *Language Technologies for Multilingual Europe: Towards a Human Language Project*. Strategic Research and Innovation Agenda. Unveiled at META-FORUM 2017. Prepared by the Cracking the Language Barrier federation, supported by the EU project CRACKER. Brussels, Belgium, 2017.

Georg Rehm, Hans Uszkoreit, Sophia Ananiadou, Núria Bel, Audronė Bielevičienė, Lars Borin, António Branco, Gerhard Budin, Nicoletta Calzolari, Walter Daelemans, Radovan Garabík, Marko Grobelnik, Carmen García-Mateo, Josef van Genabith, Jan Hajić, Inma Hernández, John Judge, Svetla Koeva, Simon Krek, Cvetana Krstev, Krister Lindén, Bernardo Magnini, Joseph Mariani, John McNaught, Maite Melero, Monica Monachini, Asunción Moreno, Jan Odić, Maciej Ogrodniczuk, Piotr Pezik, Stelios Piperidis, Adam Przepiórkowski, Eiríkur Rögnvaldsson, Michael Rosner, Bolette Pedersen, Inguna Skadina, Koenraad De Smedt, Marko Tadić, Paul Thompson, Dan Tufiş, Tamás Váradi, Andrejs Vasiljevs, Kadri Vider and Jolanta Zabarskaite. *The strategic impact of META-NET on the regional, national and international level*. Lang. Resour. Evaluation, 50(2), In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odić, Stelios Piperidis, Editors. *Proceedings*

of the Ninth International Conference on Language Resources and Evaluation 2014 (LREC'14),. pages 1517–1524, Reykjavik, Iceland, 2014. European Language Resources Association (ELRA). [.bib | .pdf]

[Riga Declaration, Declaration of Common Interests](#). Prepared and signed by 12 language and language technology stakeholders at the Riga Summit 2015 on the Multilingual Digital Single Market. 2015.

Benjamin Sargent. [The 116 Most Economically Active Languages Online](#). CommonSenseAdvisory. 2013.

Rafael Rivera Pastor, Carlotta Tarín Quirós, Juan Pablo Villar García, Toni Badia Cardús and Maite Melero Nogués. [Language equality in the digital age – Towards a Human Language Project](#). A study of the Scientific Foresight Unit (STOA) carried out by Iclaves within the Directorate-General for Parliamentary Research Services (DG EPRS) of the European Parliament. 2017.

Andrejs Vasiljevs, Khalid Choukri, Luc Meertens and Stefani Aguzzi. [Final study report on CEF Automated Translation value proposition in the context of the European LT market/ecosystem](#). A study prepared for the European Commission, DG Communications Networks, Content & Technology by Crosslang, Tilde, ELDA, IDC. Luxembourg. 2019.

Roberto Viola and Rytis Martikonis. [Multilingualism in the Digital Age: a barrier or an opportunity](#). 2017.

1.57.3 Deliverables

Note: These publications will be provided shortly.

1.58 Instructions for the registration of metadata records

1.58.1 Introduction and general recommendations

The following instructions are meant to help you provide the required metadata records for:

- The project
- Your organization, and
- The resources you will provide to the ELG catalogue.

You will find general information on how to register metadata records and examples in [Chapter 3](#). This section includes instructions adapted to the requirements of the Open Call projects.

More specifically, we include below a step-by-step process with tips that will make the registration of your metadata records easier. These take into account

- specific metadata elements required only for Open Call projects
- registration of resources that are in the process of being created and, thus, their descriptions are in-complete
- creating links between organization, project and Language Resources/Technologies, which, in ELG, is supported via retrieval mechanisms of previously added and published records.

As a general recommendation, before creating the metadata records, have a look at the examples, as they show in the ELG catalogue, to see what is expected from you. You can check out the entries for Open Call I projects through the hyperlinks from the dedicated page for [Open Call I projects](#).

You can describe all items with the ELG interactive editor or create metadata files in XML format using the templates we provide at the [ELG GitLab repository](#) and upload them at the platform. If you want, you can copy the XML templates locally, change the name of the organization/project/resource with the name of your own organization/project/resource, upload it at ELG, and continue its editing with the interactive editor.

When you finish editing your record(s), you must submit them for publication, according to the *ELG publication lifecycle*; once you do, we will be notified, check the records and publish them or contact you if edits are required.

1.58.2 Step-by-step instructions

Step 1 – Register/Sign in as a provider

First, register/sign in at <https://live.european-language-grid.eu/> and request for the *provider role*, if you don't already have such rights. All members of a project team can become providers and add metadata records. But each metadata record can be curated by only one person. So, for organizations and projects, you must select one person that will describe them.

Step 2 – Describe your organization

Describe your organization first.

If the organization has a division, describe first the parent organization, submit it for publication and wait. Once it is published, you will receive an automatic email notification and you can then proceed to add the division(s) in the same way. In this way, when you fill in the value of the metadata element “division of”, your parent organization will be included in the lookup field and the appropriate link will be created.

Step 3 – Describe your project

Describe the project, making sure you add the funding-related metadata elements with the following values:

- Funding type: EU funds
- Funder: European Commission
- Funding country: Europe
- Cost: the total budget of the project
- Funding scheme category: European Language Grid Open Call
- Related call: ELG Open Call II
- Status: approved

Please, submit it for publication and wait. Once it is published, you will receive an automatic email notification and you can then proceed to the next steps.

Step 4 – Describe your Language Technology service or data resource

Normally, you would contribute

- a service (in the format of a docker image according to the ELG technical specifications) together with its description (metadata record); however, the dockerization and integration will be done through the course of your project
- a data resource (corpus, lexical/conceptual resource, language description) in the form of a physical file with its description (metadata record); again, the creation and packaging of the file will be done during the project.

For now, you must only provide a description of each resource/service. However, in order to be published, all records must comply with the ELG minimal schema. This includes elements, such as licence, where the docker image must be downloaded from, size of the physical file, etc. that you may not know yet. These elements are not displayed for resources marked as “Work in Progress” (see, for example, an [Open Call 1 resource](#)).

For this reason, depending on your choice for entering the records, follow the following tips:

- the XML templates provided at GitLab include some “dummy” values that will ensure their import into the platform without problems. When you upload the files, make sure you check the “Work in progress” box

- if you use the interactive editor, fill in the values of the elements with any values you prefer, but check the “Work in progress” box.

Specific requirements for metadata elements on services/resources:

- add both the European Language Grid project and your Open Call project as funding projects; if you have already published the project (see Step 3 above), this will be included in the lookup field for the “funding project” metadata element.
- add your organization as “resource provider”.

When you have finished the dockerization of the service(s) and/or packaging of the resource(s), and you are ready to proceed to the next step, you will notify us so that we give you the rights to and update it with the proper information.

1.58.3 Useful links

- Templates for LRTs from Open Call projects: <https://gitlab.com/european-language-grid/platform/ELG-SHARE-schema/-/tree/master/open%20call%20templates>
- Video on ELG platform: <https://www.youtube.com/watch?v=LD6QadkkZiM>

1.59 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

PYTHON MODULE INDEX

e

- `elg.authentication`, [121](#)
- `elg.benchmark`, [129](#)
- `elg.catalog`, [119](#)
- `elg.corpus`, [123](#)
- `elg.entity`, [122](#)
- `elg.pipeline`, [132](#)
- `elg.service`, [126](#)

Symbols

`__call__()` (*elg.benchmark.Benchmark method*), 130
`__call__()` (*elg.pipeline.Pipeline method*), 133
`__call__()` (*elg.service.Service method*), 127

A

Authentication (*class in elg.authentication*), 121

B

Benchmark (*class in elg.benchmark*), 129
BenchmarkResult (*class in elg.benchmark*), 131

C

Catalog (*class in elg.catalog*), 119
Code sample, 13
`compare()` (*elg.benchmark.BenchmarkResult method*), 131
`compare_response_times()` (*elg.benchmark.BenchmarkResult method*), 131
`compare_results()` (*elg.benchmark.BenchmarkResult method*), 131
Corpus (*class in elg.corpus*), 123
`create()` (*elg.authentication.Authentication method*), 121
`create_authentication_url()` (*elg.authentication.Authentication class method*), 121
`create_docker_files()` (*elg.FlaskService class method*), 134
`create_requirements()` (*elg.FlaskService class method*), 134

D

Distribution (*class in elg.corpus*), 125
`docker_build_image()` (*elg.FlaskService class method*), 134
`docker_push_image()` (*elg.FlaskService class method*), 134
Download, 13
`download()` (*elg.corpus.Corporis method*), 124

E

`elg.authentication`
module, 121
`elg.benchmark`
module, 129
`elg.catalog`
module, 119
`elg.corpus`
module, 123
`elg.entity`
module, 122
`elg.pipeline`
module, 132
`elg.service`
module, 126
Entity (*class in elg.entity*), 122

F

FlaskService (*class in elg*), 134
`from_data()` (*elg.corpus.Distribution class method*), 125
`from_docker_image()` (*elg.service.Service class method*), 127
`from_entities()` (*elg.benchmark.Benchmark class method*), 130
`from_entities()` (*elg.pipeline.Pipeline class method*), 133
`from_entity()` (*elg.corpus.Corporis class method*), 124
`from_entity()` (*elg.service.Service class method*), 128
`from_id()` (*elg.corpus.Corporis class method*), 124
`from_id()` (*elg.entity.Entity class method*), 122
`from_id()` (*elg.service.Service class method*), 128
`from_ids()` (*elg.benchmark.Benchmark class method*), 131
`from_ids()` (*elg.pipeline.Pipeline class method*), 133
`from_json()` (*elg.authentication.Authentication class method*), 121
`from_search_result()` (*elg.entity.Entity class method*), 123

`from_success_code()` *Service (class in `elg.service`), 126*
*(`elg.authentication.Authentication` class `set_colwidth()` (`elg.benchmark.BenchmarkResult`
method), 121 *method), 131**

I

`init()` *(`elg.authentication.Authentication` class `to_json()` (`elg.authentication.Authentication`
method), 121 *method), 122**
`interactive_search()` *(`elg.catalog.Catalog` Try Out, 13
method), 119*
Internal LT API, 209
`invalid_request_error()` *(`elg.FlaskService`
method), 134*
`is_downloadable()` *(`elg.corpus.Distribution`
method), 125*

L

Licence *(class in `elg.corpus`), 125*

M

module
 `elg.authentication`, 121
 `elg.benchmark`, 129
 `elg.catalog`, 119
 `elg.corpus`, 123
 `elg.entity`, 122
 `elg.pipeline`, 132
 `elg.service`, 126

N

`need_authentication()` *(in module
`elg.authentication`), 122*
`NeedAuthentication` *(class in `elg.authentication`),
122*

P

`Pipeline` *(class in `elg.pipeline`), 132*
`process()` *(`elg.FlaskService` method), 134*
`process_audio()` *(`elg.FlaskService` method), 134*
`process_request()` *(`elg.FlaskService` method), 134*
`process_structured_text()` *(`elg.FlaskService`
method), 134*
`process_text()` *(`elg.FlaskService` method), 134*
Public API, 220

R

`refresh()` *(`elg.authentication.Authentication`
method), 122*
`refresh_if_needed()`
 *(`elg.authentication.Authentication` method),
 122*
`run()` *(`elg.FlaskService` method), 134*

S

`search()` *(`elg.catalog.Catalog` method), 120*