



EUROPEAN LANGUAGE GRID

D1.5

Base Infrastructure (Final Release)

Authors: Florian Kintzel (DFKI), Georg Rehm (DFKI)

Dissemination Level: Public

Date: 30-09-2021

About this document

Project	ELG – European Language Grid
Grant agreement no.	825627 – Horizon 2020, ICT 2018-2020 – Innovation Action
Coordinator	Prof. Dr. Georg Rehm (DFKI)
Start date, duration	01-01-2019, 42 months
Deliverable number	D1.5
Deliverable title	Base Infrastructure (Final Release)
Type	Report
Number of pages	228
Status and version	Final – Version 1.0
Dissemination level	Public
Date of delivery	Contractual: 30-09-2021
WP number and title	WP1: Grid Platform – Base Infrastructure
Task number and title	Task 1.2: Installation, setup and adaptation of base infrastructure
Authors	Florian Kintzel (DFKI), Georg Rehm (DFKI)
Reviewers	Ulrich Germann (UEDIN), Andres Garcia Silva (EXPSYS)
Consortium	Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Germany Institute for Language and Speech Processing (ILSP), Greece University of Sheffield (USFD), United Kingdom Charles University (CUNI), Czech Republic Evaluations and Language Resources Distribution Agency (ELDA), France Tilde SIA (TILDE), Latvia Hensoldt GmbH (HENS), Austria Expert System Iberia SL (EXPSYS), Spain University of Edinburgh (UEDIN), United Kingdom
EC project officers	Philippe Gelin, Miklos Druskoczi
For copies of reports and other ELG-related information, please contact:	DFKI GmbH European Language Grid (ELG) Alt-Moabit 91c D-10559 Berlin Germany Prof. Dr. Georg Rehm, DFKI GmbH georg.rehm@dfki.de Phone: +49 (0)30 23895-1833 Fax: +49 (0)30 23895-1810 http://european-language-grid.eu © 2021 ELG Consortium

Table of Contents

List of Figures	4
List of Tables	4
List of Terms	5
History of Changes	6
Abstract	6
1 Introduction	6
2 Hosting	7
2.1 Overview	7
2.2 Operational report for Years 1-3	7
2.3 Hardware Budget Planning	8
3 Managing and Deploying the ELG Platform	8
3.1 ELG Hardware Cluster Configuration	9
3.2 ELG Package Repository	10
3.3 ELG Cluster Admin Repository	10
3.4 ELG External Configuration	12
3.5 Docker Registries	12
3.6 ELG Components	13
3.6.1 Build-Bot	13
3.6.2 Cert-Manager	13
3.6.3 Cluster-Autoscaler	13
3.6.4 Knative and Kourier	13
3.6.5 Nginx-Ingress	14
3.6.6 Keycloak	14
3.6.7 Elasticsearch	14
3.6.8 Postgresql	14
3.6.9 MariaDB	14
3.6.10 s3proxy	14
3.6.11 Prometheus, Grafana, Loki and AlertManager	14
3.7 Continuous Integration	15
3.8 Slack Integration	16
4 Documentation and Training	17
5 Conclusion	17
6 Annex	18
6.1 Monthly hosting cost ELG	18
6.2 ELG documentation	19

List of Figures

Figure 1: ELG Base Infrastructure Configuration and Deployment _____ 9

Figure 2: Grafana cluster capacity dashboard _____ 15

Figure 3: ELG Continuous Integration _____ 16

List of Tables

Table 1: Hardware budget _____ 8

Table 2: Infrastructure budget consumption _____ 19

List of Terms

Cluster (Kubernetes)	A set of machines (nodes), that run containerized applications managed by Kubernetes.
Container (Kubernetes)	A running instance of an image (see below).
Controller (Kubernetes)	A loop that watches the shared state of a cluster through the API server and makes changes attempting to move the current state towards the desired state.
Deployment (Kubernetes)	An object in Kubernetes (a resource) that manages the lifecycle a replicated application.
Deployment	The action of making an installed and configured software application available to its intended internal or public users.
Docker	A software technology providing operating-system-level virtualization also known as containers.
Helm	Helm is a package manager for Kubernetes, i.e., a tool for the easy installation of pre-packaged components inside a Kubernetes cluster.
Helm chart	A configuration file for Helm listing all components of a package with their dependencies and additional configuration options.
Image	A single file containing software along with all its dependencies. Can be run as an application through software such as Docker.
Infrastructure as code	Infrastructure as code (IaC) is the process of managing and provisioning computer data centers through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools.
(Machine) Instance (Cloud Provider)	A typical cost calculation unit as delivered by cloud IaaS providers. An instance can come in different sizes, e.g., small, medium, large with respect to cores and available RAM. The instance is represented by a virtual machine with the respective hardware resources.
Kubelet (Kubernetes)	An agent that runs on each node in the cluster. It makes sure that containers are running in a pod.
Namespace (Kubernetes)	A concept in Kubernetes used to partition a Kubernetes cluster into multiple parts. Most Kubernetes objects belong to exactly one namespace. Can be used e.g., to limit access for users to certain namespaces only.
Node (Kubernetes)	An abstraction for anything offering compute resources to a Kubernetes cluster. Usually, instances (see above) are mapped into a Kubernetes cluster as nodes.
Object Storage	A storage architecture that manages data as objects with unique identifiers.
Pod (Kubernetes)	The smallest deployable units of computing that can be created and managed in Kubernetes. Consists of at least one container. Is scheduled to a fitting node automatically by Kubernetes.
S3	An object storage solution developed by Amazon. Today, often used to refer to solutions by other providers that have APIs compatible to the Amazon S3 API.
Service (Kubernetes)	An API object that describes how to access applications, such as a set of Pods, and can describe ports and load-balancers.
Volume (Kubernetes)	A directory containing data, accessible to the containers in a pod.

History of Changes

Version 1, August 2021

- Initial version (changes with regard to D1.4)
 - Updated hosting section with 2021 content
 - Updated budget calculation regarding the ELG project extension into 2022
 - Updated operational report regarding year 3
 - Updated “Cluster admin repository” with new namespaces, components, user-roles
 - Updated component description with changes in 2021
 - Updated “Continuous Integration”, describing newly configure runners
 - New section describing Slack integration
 - Updated “Documentation and training”
 - Moved “Documentation and training” to a separate section
 - “Conclusion and outlook” updated to newest state

Abstract

This document is an update to ELG Deliverable D1.4, *Base Infrastructure (Third Release)*, and describes the current state of the ELG base infrastructure. It provides information about the current hardware setup, the base software and developer tooling, i.e., continuous integration. The deliverable includes all sections from D1.4 and updates them where applicable.

1 Introduction

This deliverable is an update of Deliverable D1.4, keeping all sections and updating them as necessary. It provides a full overview of the ELG infrastructure in its current form.

The ELG development effort distinguishes three major components: the base infrastructure (WP1; this deliverable), platform (WP2; D2.x), and front-end (WP3; D3.x). The contribution of WP1 consists of the following points which correspond to the indicated sections in this document:

- Hosting situation and budget (Section 2)
- Management of ELG cluster (Sections 3.1 - 3.5)
- Integration of off-the-shelf components (such as database servers, identity management solutions) and development of a smaller set of custom infrastructure components (Section 3.6)
- Providing development tooling, i.e., continuous integration (Section 3.7)
- Training and know-how transfer to WP2-5 regarding the base infrastructure (Section 4)

The architecture of the ELG platform proper is not part of this deliverable but included in the platform deliverables (D2.x).

This deliverable updates D1.4 in all aspects changed in 2021, mainly consisting of (1) the operational and hosting situation of 2021, (2) impact of the ELG project extension into 2022, (3) changes to platform components and tooling, (4) updates to the documentation and training material.

2 Hosting

2.1 Overview

ELG contracted the cloud service provider SysEleven GmbH (Berlin)¹ in 2019 and initially provisioned one cluster named “development”. At the end of 2019 the consortium extended the contract to include a second hardware cluster, “demonstration” specifically to showcase the current state at META-FORUM on 8/9 October 2019. The demo cluster was deallocated in early 2020 and replaced by the first publicly available version of the ELG, the “live” cluster in 2020. A new contract was signed with SysEleven GmbH in 2020, running until the end of 2021, the initially planned project end date. The contract was extended again in July 2021 to cover the extended project timeline until June 2022.

The contract provides for a base amount of hardware which was calculated to be sufficient for the ELG base load regarding both development and live clusters. On top of that, resources can be added dynamically to the clusters, in peak load situations. This way, the actual capacity of the clusters can be about doubled on short notice.

As of August 2021, we utilize both the “development” and “live” clusters. Additionally, a smaller cluster “continuous integration” has been set up, to support the ongoing development of the platform software. The two clusters “development” and “live” have a total base capacity of about 50 cores/200 GB RAM/2T SSD and the option of doubling that amount on demand. Additionally, we utilize S3 compatible object storage for persistence of large datasets (e.g., LT corpora) which is scaled up dynamically, based on usage.

2.2 Operational report for Years 1-3

In the first two years of the project, no major issues were encountered during the operation of the infrastructure cluster. In the second year, a monitoring solution (Prometheus) was included in the setup, offering metrics, statistics and warning notifications. Together with existing notification and warning services offered by SysEleven GmbH regarding CVE² announcements, maintenance windows and incident notifications, this forms the basis of the ELG operations. The hardware setup was sufficient for the continued use of the ELG up to now and is expected to remain so up until the project end date.

In 2021, with the extended usage of the platform, scheduled or unexpected downtimes started becoming a topic of higher priority. Therefore, a major focus in 2021 was to increase reliability and observability of the platform. Specifically, all software components of the ELG platform were configured to run in a replicated setup, to be able to tolerate compute node failures. As of August 2021, this has been mostly completed, with only a few components needing adjustments to be able to run with replication enabled.

¹ <https://www.syseleven.de>

² CVE: Common Vulnerabilities and Exposures: Information regarding software security concerns

Regarding observability, relevant alerting and status messages were configured to be sent to the newly introduced Slack channel (see 3.8), where the ELG technical team can coordinate to investigate any upcoming issues.

2.3 Hardware Budget Planning

The budget distribution, was originally planned in Deliverable D1.1³ and had so far served well regarding the actual hardware needs encountered. Due to the project end date being extended into 2022, it was decided to stretch the available budget into 2022. This was possible as the actual incurred costs in 2019-2021 were well below estimation. The updated budget planning is:

Year	Budget (original)	Budget (updated calculation)	Actual
2019	11,000€	9,000€	8,590€
2020	33,000€	24,000€	23,350€
2021	66,000€	44,000€	20,464€ (up to Aug 21)
2022 (Jan-July)	n/a	23,000€	n/a

Table 1: Hardware Budget

A detailed overview about monthly costs can be found in the appendix.

3 Managing and Deploying the ELG Platform

While WP2 handles the ELG platform development, its deployment and configuration are part of WP1. For managing and deploying the ELG platform (core components and LT services) the following packages and configurations are required.

- Hardware cluster configuration (networking, storage, compute)
- ELG package repository (infrastructure-as-code)
- ELG cluster administrator repository (continuous integration)
- ELG external configuration (management of access credentials)

They are described in their following Sections 3.1 to 3.4. Sections 3.5 and 3.6 describe additional components needed to deploy a complete ELG platform instance. Together, these components form the basis of the infrastructure configuration. The components do not only form one cluster, but, rather, can be combined to create multiple instances of the ELG platform (Figure 1).

³ ELG Deliverable D1.1 (M4): *Requirements and Architectural Specification of the Base Infrastructure*

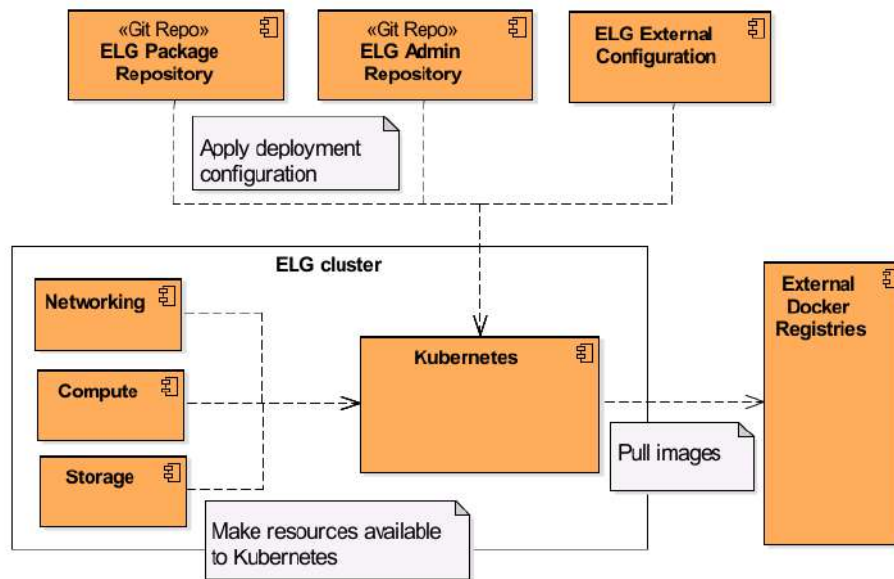


Figure 1: ELG Base Infrastructure Configuration and Deployment

The currently available instances are:

- ELG development cluster (“dev cluster”) - Not publicly available
- ELG public cluster (“live cluster”) <https://live.european-language-grid.eu/>
- Various ELG instances on the local machines or clusters of the ELG development team

The current setup of each cluster is maintained as infrastructure-as-code inside the ELG Package Repository (see Sec. 3.2). Using this approach, we can easily replicate the infrastructure to serve different purposes (stable deployment vs development). There are no manual changes being done to any of the clusters, the whole deployment is exclusively controlled by the state of the Git repository and is therefore fully versioned and easily rolled back to a previous version in case this becomes necessary.

The rationale for the way we configured the platform is to have a core common part, that is identical in all instances (the ELG package repository) as well as specific parts (“configurations”), for each of the instances of the ELG (hardware cluster configuration, cluster administration repository, external configuration). The split of the latter (the specific instance configuration) into three parts is due to technical (the hardware cluster configuration) and security reasons (administration repository and ELG external configuration).

When creating or updating an instance of the ELG, it is necessary to combine the core configuration with the respective specific configuration for the cluster in question. All these parts are described in the following section.

3.1 ELG Hardware Cluster Configuration

This configuration manages the (virtual) hardware of each cluster. This configuration is maintained with the help of the tools provided by the respective cloud provider (see annex). It handles the provisioning of compute, storage, networking resources as well as the setup of the base Kubernetes system on top of this hardware.

The cloud-based ELG instances (in the SysEleven data centre) are managed and maintained via the SysEleven MetaKube Dashboard. The hardware setup thus configured defines the hardware limitations of the instance (i.e., CPU cores, RAM, storage, etc.), which is made known to Kubernetes to manage. Kubernetes manages the

distribution of workloads (containers) to the available hardware. Kubernetes does *not* manage the actual hardware as configured by the cloud provider's toolbox. While the actual hardware configuration is specific to each cluster, the Kubernetes API offers a unified interface for workload management.

Setting up or configuring the hardware configuration of a cluster is a manual process involving the provider's UI. This configuration determined the hardware limitations, and accordingly the infrastructure costs. Certain parts of the hardware configuration do not need to be configured statically but are always available in any quantity and priced strictly on usage; in the case of the ELG that is the object storage solution.

Finally, a cluster configuration also needs to be done for local testing and development of the ELG. In this case, usually minikube⁴ is installed on the developer's machine.

3.2 ELG Package Repository

The ELG package repository⁵ consists of a set of configuration files, mostly in the form of Helm charts. Helm charts define which packages, i.e., containers, the ELG system consists of, as well as numerous additional configuration parameters for these packages, e.g., the number of replicas and package-specific configuration.

The ELG package repository can be used to set up multiple clusters, as there is no cluster-specific configuration inside the repository. However, we maintain different branches within the repository to hold different versions of the ELG package repository, usually at least one for the development and one for the live cluster. The branches are not used to distinguish between specific configurations for each of the clusters, but rather present different versions of the ELG system as it matures during development. This is used to facilitate a staged rollout to the production cluster, i.e., first rolling out a new version on a test cluster and later, once successfully tested, to the production cluster.

This Git repository is currently marked as private, but might later, be made public to facilitate third parties to make use of the ELG base system, e.g., ICT-29b projects and others.

The content configured within the ELG package repository mainly consists of:

- All third-party components like databases, authentication servers etc., their version to use and various configuration options
- All ELG custom developed components, like the catalogue frontend and backend components (these are described in deliverables 2.x of work package 2), also along with their version and configuration
- All LT services from all providers
- The configuration necessary to link all those microservices together

For obvious security reasons, all sensitive information like password for the database etc. cannot be stored in a git repository. For those, only templates exist (see Section 3.4 for more details).

It should be noted here that the actual source code of all those components is not part of this repository. This repository includes only references to the docker registries holding the specific image versions. When installing any ELG cluster, these images are then downloaded ("pulled") from one of these registries.

3.3 ELG Cluster Admin Repository

⁴ <https://kubernetes.io/de/docs/setup/minikube/>

⁵ <https://gitlab.com/european-language-grid/platform/infra>

The private ELG Cluster Admin repository⁶ holds cluster-specific configurations for each ELG instance:

- The list of active users for each instance
- Their roles and access rights within each instance
- The code and configuration for build-bot, the ELG continuous integration utility
- Utility scripts for the cluster administrators to, e.g., create new users for the cluster
- The infrastructure-as-code configuration for the “continuous integration” cluster

The users mentioned above are users that need access to the actual Kubernetes API and resources, i.e., administration users, ELG developers for debugging, etc. They are different and managed separately from actual users of the ELG platform, i.e., those accessing certain sections of the ELG platform proper or calling LT services. The latter set of users is not in the scope of the infrastructure configuration but of the platform itself.

This repository is not needed for local deployment of the ELG, as such a deployment is usually only meant for a single user and does not participate in continuous deployment.

We use Kubernetes namespaces to separate the different platform components from one another. A namespace in Kubernetes terminology is a set of resources exclusively grouped within it, i.e., a resource is only part of a single namespace. A resource in Kubernetes terms means a container, pod, ingress-role or any other Kubernetes object except for cluster-wide resources, e.g., persistent volumes.

Currently, we have defined the following namespaces:

- elg-admin: Admin-only namespace dealing mostly with the continuous integration
- elg-backend-dev: The ELG backend components (e.g., the catalogue)
- elg-core-dev: ELG core components (currently mostly nginx-ingres and ingress rules)
- elg-portal-dev: ELG portal components (i.e., the portal website)
- elg-srv-dev: All open source LT services components
- elg-srv-expsys: Closed source or otherwise sensitive LT service components from ExpertSystems
- elg-srv-saillabs: Closed source otherwise sensitive LT service components from SailLabs
- elg-srv-usfd: Closed source otherwise sensitive LT services components from University of Sheffield
- Various other elg-srv-X namespaces for additional third parties (i.e., ELG pilot projects)
- users: Definitions for all users of the cluster (i.e., the content of the ELG cluster admin repository)
- syseleven-kubernetes-dashboard: Reserved by SysEleven GmbH
- syseleven-certmanager: Management of TLS certificates
- kourier-system: For installation of the kourier⁷ API gateway (Gloo-system was used before, but it was changed for kourier for better support regarding kubernetes updates)
- knative-serving: For installation of knative⁸ (dynamic pod scheduling)
- Elg-monitoring: Holding the prometheus monitoring solution and supporting services
- velero: Reserved by SysEleven GmbH (backup)
- webterminal: Reserved by SysEleven GmbH
- a small set of kube-* namespaces, reserved for Kubernetes itself

⁶ <https://gitlab.com/european-language-grid/platform-admin/env-dev>

⁷ <https://github.com/knative-sandbox/net-kourier>

⁸ <https://knative.dev/>

The current set of defined user roles is as follows:

- `elg-backend-dev-role`: Role for ELG backend developers, giving access to the `elg-backend-dev` namespace
- `elg-portal-dev-role`: Role assigned to portal developers, giving access to the `elg-portal-dev` namespace
- `expsys-role`: Specific role for partner Expert System, giving access to their respective namespace
- `saillabs-role`: Specific role for partner Sail Labs
- `usfd-role`: Specific role for the University of Sheffield
- `srv-provider-role`: Standard role, assigned as default to all users, allows access to the namespaces `elg-srv-dev` namespace (the open-source LT services)
- Cluster-admin roles: For administrators with full access

3.4 ELG External Configuration

This external platform configuration consists of certain parts of the cluster configuration that are not meant to be stored in any repository for security reasons (“external” means here not being part of the package repository). These mostly consists of:

- The user access credentials (for access to the technical cluster infrastructure). These are distributed to their specific user securely. They are specific to each cluster.
- Access credentials to third-party closed source images that need to be installed inside the cluster. This currently consists mostly of the credentials to access the private docker registries of the commercial partners ExpertSystems, SailLabs and Tilde.
- Access credentials to cluster internal components like backend databases, mail servers etc.

This information is only stored inside each cluster locally (as Kubernetes secrets) and is accessible only to authorised users with the specific role to do so. They cannot be automatically configured by the CI and need to be applied manually to the cluster.

The secrets are maintained by the respective partners or by the ELG administration team.

3.5 Docker Registries

The images for instantiating containers inside the ELG cluster are stored in various Docker registries. There are multiple registries from which Kubernetes pulls the specified images, including:

- The ELG GitLab project registry⁹ that hosts the images for all ELG core platform components (e.g., UI, backend components) and for several ELG LT services.
- The public registries for third-party components like `cert-manager`, `postgresql`, `keycloak` etc (e.g., DockerHub).
- Private Docker registries from partners who do not publish their LT services under an open-source license
- Various other public registries for open-sourced LT services

These registries are not part of the ELG infrastructure as such. The building of images for example is using different continuous integration solutions, e.g., GitLab CI for the ELG backend components and other in-house CI

⁹ See registries under <https://gitlab.com/european-language-grid>

solutions for the components developed by ELG's commercial partners. For private image registries, access credentials need to be installed into a cluster for Kubernetes to be able to access them.

3.6 ELG Components

Apart from the custom core components, developed by the ELG project that form the ELG platform and LT services, a set of third-party components is also installed as is, providing their functionality to the cluster. A few more infrastructure related components are also developed in WP1. All of them are described below.

3.6.1 Build-Bot

Build-bot is a small utility that is responsible for delivering the last part of the continuous integration chain, i.e., the update of the given cluster with the latest state of the respective branch of the ELG package repository. The continuous integration is described in more detail in the section about continuous integration.

3.6.2 Cert-Manager

Cert-manager¹⁰ is a tool to manage issuing and updating of TLS certificates (from Let's Encrypt¹¹). It is used to install and refresh TLS certificates to allow for the encryption of all HTTPS traffic reaching the cluster via one of the configured ingress-rules.

3.6.3 Cluster-Autoscaler

The Horizontal Pod Autoscaler is a standard Kubernetes component used to scale pods based on their current load and runtime behaviour. For scalability and load monitoring, Kubernetes collects certain metrics, e.g., CPU and memory load, from each pod. Therefore, it is necessary to have at least one instance of each type of pod to be up and running at all times. Otherwise, no metrics can be collected. This setup is useful to scale ELG core components, e.g., the portal website and backend. It cannot be utilized as-is to scale the large number of LT services offered by the platform, as these need to be scaled down to zero replicas in case they are not needed to not exceed the cluster capacity. Therefore, knative (see below) was introduced, which is feeding the standard autoscaler with a new metric "*concurrency*", based on the number of active requests to that LT service. Scaling those services still makes use of cluster-autoscaler functionality, but with the new metric also being available if no active replica of an LT services is instantiated.

3.6.4 Knative and Kourier

Knative¹² together with Kourier¹³ give us the possibility to scale down LT services based on the current number of parallel requests to them (concurrency). The concurrency metric is available even if there is no active replica of the LT services instantiated. Knative buffers HTTP requests to one of the ELG APIs until the specific LT services container has started in the background and keeps track of the concurrency metric to terminate the replica if it is no longer needed. We cannot overstate the importance of this functionality for the ELG platform as in its final release, the platform consists of hundreds of individual LT service components where most of them are expected to see relatively little traffic (LT services deployed for demonstration or trial purposes), and it would not be efficient to have all of them consume resources while in idle state. Starting up a container takes a certain amount of time though, while the service initializes. Using a service after it has not been used in a while

¹⁰ <https://github.com/jetstack/cert-manager>

¹¹ <https://letsencrypt.org>

¹² <https://knative.dev/>

¹³ <https://github.com/knative-sandbox/net-kourier>

therefore requires a certain spin-up time. Depending on the service implementation, this is usually a few seconds to a few minutes.

If frequent traffic is expected for a particular service, it can easily be configured to have one or more instances running at any given time. This will eventually not be possible for all services without a price though. In a later project phase, and with introduction of a billing component to the ELG, a mechanism is to be developed where users and providers will be asked to contribute to the runtime costs of high-load LT services.

With Knative and Kourier though, we are confident to have the technical solution at hand to adapt the scaling profile of LT services easily.

3.6.5 Nginx-Ingress

Ingress-Ningx¹⁴ is installed to act as ingress-controller, i.e., handling HTTP traffic received and forwarding them to their respective endpoint within the cluster.

3.6.6 Keycloak

Keycloak¹⁵ is an open-source solution for authentication and authorization. It interfaces with front-end, back-end and LT services to provide a single-sign on experience.

3.6.7 Elasticsearch

Elasticsearch¹⁶ is currently used exclusively to index the catalogue database for fast faceted search.

3.6.8 Postgresql

The main ELG backend database is based on postgresql¹⁷, a well-supported open-source database engine. It holds all relevant data concerning the ELG catalogue, e.g., projects, organisations, LT resources, LT service as well as user information.

3.6.9 MariaDB

MariaDB is used for persistence of the Drupal CMS powering the ELG portal pages. It is envisioned to move this over to Postgresql for ease of maintenance.

3.6.10 s3proxy

Not an off-the-shelf component, but rather specifically adapted for the ELG, the s3proxy¹⁸ is necessary for technical reasons to facilitate the upload of LT resources (models, corpora, but also project and organisation logos etc) to the ELG. It acts as a proxy to the SysEleven-hosted s3-compatible object storage, that takes care of validating upload authorization with the ELG backend and streams data onwards to the object storage.

3.6.11 Prometheus, Grafana, Loki and AlertManager

¹⁴ <https://kubernetes.github.io/ingress-nginx/>

¹⁵ <https://www.keycloak.org>

¹⁶ <https://www.elastic.co/de/elasticsearch/>

¹⁷ <https://www.postgresql.org/>

¹⁸ <https://gitlab.com/european-language-grid/platform/s3proxy>

Prometheus¹⁹, Grafana²⁰, Loki²¹ and AlertManager²² together complement each other to form the ELG monitoring solution. With them, it is possible to:

- Collect and analyse logs from all running components inside the cluster
- Collect various metrics from hardware to custom defined ones
- Provide visualizations for those metrics in the form of various dashboards and diagrams
- Configure thresholds for each metric, and send warning notifications to the ELG administrative team in case any metric exceeds those thresholds



Figure 2: Grafana cluster capacity dashboard

3.7 Continuous Integration

Continuous Integration during development of the ELG is achieved using a combination of Git repositories, GitLab CI, webhooks and Docker registries.

Continuous integration regarding the grid infrastructure only deals with updating a given ELG cluster with the latest set of images (as specified by their version number) and configuration. It does *not* deal with the building of the respective images. The latter is out of scope for this document and depends on the CI workflows of the respective service providers. Figure 3 gives an overview of the setup.

¹⁹ <https://prometheus.io/>

²⁰ <https://grafana.com/>

²¹ <https://grafana.com/oss/loki/>

²² <https://prometheus.io/docs/alerting/latest/alertmanager/>

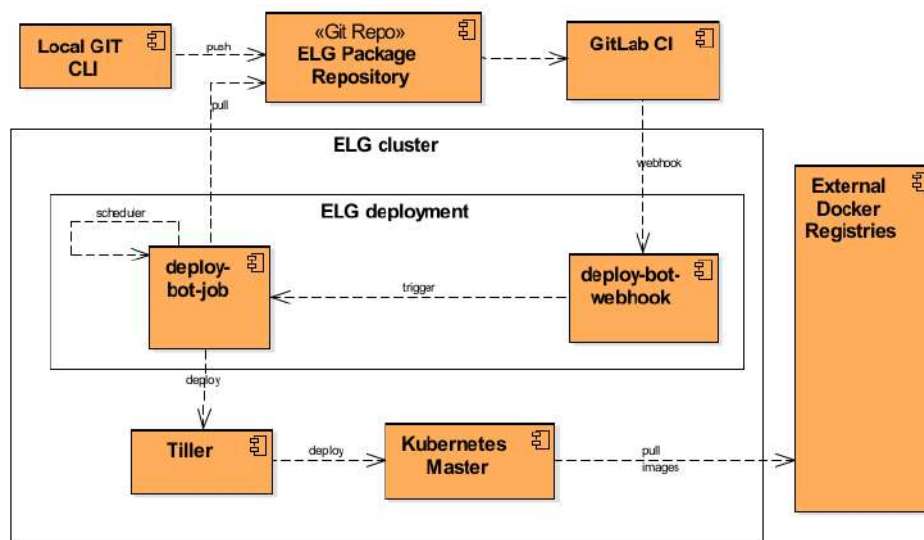


Figure 3: ELG Continuous Integration

Each push to the ELG infrastructure Git repository triggers a webhook on GitLab, which notifies the respective cluster as identified by the branch where the push is being made (development and live). Inside the respective cluster, the notification webhook is received by `deploy-bot-webhook` components which are, in turn, queried by the `build-bot-job` which is triggered at regular intervals.

If a new version of the infrastructure setup is detected, the `build-bot-job` checks out the respective branch and executes the deployment scripts from within the cluster. As the configuration mostly consists of a set of container images and their versions, the respective images are then pulled from their distributed docker registries when needed. The Kubernetes cluster is updated with the latest configuration and takes care of gracefully shutting down and instantiating new pods whenever a configuration has changed.

The ELG has been using Gitlab’s²³ public shared runners (compute nodes compiling and assembling the software) which are free to use for open-source projects up until mid 2021, but has recently installed their own runners, hosted in a newly created cluster “continuous integration”. Gitlab has in the past reduced the capacity available free of charge for all projects and that has necessitated the change. This has been set-up in such a way, that runners installed on the “continuous integration” cluster register at gitlab.com to take over specific build jobs from the ELG main repository.

3.8 Slack Integration

The chat platform Slack²⁴ had been used for some time already as a discussion forum for technical issues regarding the ELG. But in 2021, this has been extended to with a tighter integration that makes it possible to push messages from the continuous integration directly to Slack. Therefore, developers no longer need to log into different services to be up to date with deployments, maintenance and other topics.

Slack is also used extensively to get in touch with ELG power users, most notably the participants from the ELG pilot projects (ICT-29b).

²³ <https://about.gitlab.com/>

²⁴ <https://slack.com>

4 Documentation and Training

WP1 is not only responsible for the hardware setup, the base software and developer tooling but also takes care of the internal knowledge transfer within the consortium regarding matters that relate to the platform infrastructure. To train consortium members in the usage of the base infrastructure, development infrastructure and continuous integration the technical team of ELG held regular meetings:

- Training and hands-on assistance during the three one-week hackathons
 - Berlin, 20-24 May 2019
 - Sheffield, 28 July-2 August 2019
 - Athens, 2-6 December 2019
- Bi-weekly virtual meetings of the task force infrastructure, open for participation for all consortium members
 - Virtual session on demand in 2020 and 2021

Much of the discussion has been moved online due to COVID-19 restrictions. The chat platform Slack has been a major cornerstone to keep the communication going among the technical team. All discussion takes place in dedicated channels²⁵.

In addition, WP1 also administers the ELG intranet²⁶, where a host of resources and documentation is available.

The training material used for this transfer of information is based on the documentation of the infrastructure and platform. It consists of the following parts:

- Platform usage documentation (<https://european-language-grid.readthedocs.io/en/latest/>)²⁷
 - Focuses on the integration of LT services and resources into the ELG
 - Is maintained by the consortium
 - Is intended for suppliers of LT services and resource providers
 - Is constantly extended and adapted to new developments by the ELG technical team
- Technical documentation of the Kubernetes / OpenStack system
 - Online resource under <https://docs.syseleven.de/syseleven-stack/en>²⁸
 - Focuses on the configuration and administration of the cloud platform the ELG is running on
 - Is maintained by the provider of the cloud platform (SysEleven GmbH)

It was originally conceived to have separate training sessions managed by the hosting provider. Up to now, though the inhouse-knowledge regarding the virtualization infrastructure was sufficient. We nevertheless keep this option a possibility for the rest of 2021 and 2022, for architecture security review and consulting. With this approach, we are confident to utilize the available budget to best effect.

5 Conclusion

²⁵ see <https://europeanlanguagegrid.slack.com/>

²⁶ With the knowledge resources available in the ELG internal intranet

²⁷ A PDF export of the documentation can be found in the annex.

²⁸ This resource is also available for download at <https://github.com/syseleven/grav-docs-syseleven-stack>

This document describes the current state of the ELG base infrastructure as of August 2021. The hardware setup has been adapted and contractually secured until the end of the extended project end date. The adopted solution for zero-downscaling of LT services has continued to support the ELG use case well. Additionally, the operational readiness of the ELG has significantly increased with the improvements to the Prometheus monitoring and extended replication of platform software components.

6 Annex

6.1 Monthly hosting costs ELG

Month	Cost	Comment
ELG Platform 08/2021	2,585.98€	New cluster “continuous integration” created
ELG Platform 07/2021	2,580.94€	Contract extended until July 2022
ELG Platform 06/2021	2,584.04€	
ELG Platform 05/2021	2,585.44€	
ELG Platform 04/2021	2,591.55€	
ELG Platform 03/2021	2,594.35€	
ELG Platform 02/2021	2,519.14€	
ELG Platform 01/2021	2,350.50€	
ELG Platform 12/2020	2,613.58€	
ELG Platform 11/2020	2,592.02€	
ELG Platform 10/2020	2,586.24€	
ELG Platform 09/2020	2,581.18€	
ELG Platform 08/2020	2,581.83€	
ELG Platform 07/2020	2,350.40€	
ELG Platform 06/2020	3,196.09€	New contract signed; cluster “live” extended
ELG Platform 05/2020	1,980.82€	
ELG Platform 04/2020	1,785.68€	New cluster created (“live”)
ELG Platform 03/2020	1,280.88€	Cluster “demo” deallocated
ELG Platform 02/2020	1,207.58€	
ELG Platform 01/2020	1,207.66€	
ELG Platform 12/2019	1,205.38€	
ELG Platform 11/2019	1,702.66€	Extended hardware for META-FORUM 2019 (on-demand hardware is billed in month following their usage)
ELG Platform 10/2019	1,131.12€	Second cluster created (“demo”)

ELG Platform 09/2019	910.28€ –
ELG Platform 08/2019	910.28€ –
ELG Platform 07/2019	910.28€ –
ELG Platform 06/2019	910.28€ –
ELG Platform 05/2019	910.28€ First cluster (“dev”) created beginning of May 2019
Total	54,946.46€

Table 2: Infrastructure budget consumption

6.2 ELG documentation

European Language Grid

Release 2

Sep 29, 2021

Chapter 1: Introduction

1	Introduction	3
2	Types of catalogue entries	5
3	Using ELG	7
4	Browse the catalogue	9
4.1	View catalogue	9
4.2	View catalogue entry	9
5	Register as a user	19
6	Use an LT service	21
6.1	Try-out UI	21
6.2	Call a service via the API	21
7	Download a resource	25
8	Contributing to ELG	29
9	Register as a provider	31
10	Contribute a service	33
10.1	Overview: how an LT Service is integrated to ELG	33
10.2	Before you start	35
10.3	Step 0: Dockerize your service	36
10.4	Step 1: create metadata	39
10.5	Step 2: register the service with the platform	44
10.6	Step 3: wait for approval	44
10.7	Frequently asked questions	48
11	Contribute downloadable software	49
11.1	Before you start	49
11.2	Step 1: create metadata	49
11.3	Step 2: upload	49
11.4	Step 3: wait for approval	50
12	Contribute a corpus/dataset	53

12.1	Before you start	53
12.2	Step 1: create metadata	53
12.3	Step 2: upload	61
12.4	Step 3: wait for approval	63
13	Contribute a model	65
13.1	Before you start	65
13.2	Step 1: create metadata	65
13.3	Step 2: upload	69
13.4	Step 3: wait for approval	70
14	Contribute a grammar	71
14.1	Before you start	71
14.2	Step 1: create metadata	71
14.3	Step 2: upload	74
14.4	Step 3: wait for approval	74
15	Contribute a lexical/conceptual resource	77
15.1	Before you start	77
15.2	Step 1: create metadata	77
15.3	Step 2: upload	87
15.4	Step 3: wait for approval	89
16	Contribute a project	91
16.1	Before you start	91
16.2	Step 1: create metadata	91
16.3	Step 2: upload	97
16.4	Step 3: wait for approval	98
17	Contribute an organisation	99
17.1	Before you start	99
17.2	Step 1: create metadata	99
17.3	Step 2: upload	104
17.4	Step 3: wait for approval	104
18	Update / Delete a catalogue entry	107
19	Contribute via an external repository	109
19.1	Metadata harvesting	109
20	Metadata schema	111
20.1	Basic concepts	111
20.2	Full schema documentation	113
20.3	Minimal version	113
21	Minimal elements for all entities	115
21.1	MetadataRecord	115
22	Minimal elements for all language resources and technologies	117
22.1	resourceName	117
22.2	resourceShortName	117
22.3	description	118
22.4	LRIdentifier	118
22.5	logo	119
22.6	version	119
22.7	additionalInfo	119

22.8	keyword	120
22.9	domain	120
22.10	resourceProvider	121
22.11	publicationDate	122
22.12	resourceCreator	122
22.13	fundingProject (RA)	123
22.14	intendedApplication	123
22.15	isDocumentedBy	124
23	Minimal elements for tools/services	125
23.1	ToolService	125
23.2	function	125
23.3	SoftwareDistribution	126
23.4	licenceTerms	127
23.5	languageDependent	128
23.6	inputContentResource	128
23.7	outputResource	129
23.8	language	130
23.9	framework	131
23.10	implementationLanguage	131
23.11	requiredHardware	131
23.12	trl	132
23.13	evaluated	132
24	Minimal elements for corpora	133
24.1	Corpus	133
24.2	corpusSubclass	133
24.3	CorpusTextPart	134
24.4	CorpusAudioPart	135
24.5	CorpusVideoPart	137
24.6	CorpusImagePart	138
24.7	TextGenre	139
24.8	AudioGenre	139
24.9	SpeechGenre	140
24.10	VideoGenre	140
24.11	ImageGenre	141
24.12	DatasetDistribution	141
24.13	distributionTextFeature	143
24.14	distributionAudioFeature	144
24.15	distributionVideoFeature	145
24.16	distributionImageFeature	146
24.17	personalDataIncluded	146
24.18	sensitiveDataIncluded	147
24.19	anonymized	147
24.20	annotation	147
25	Minimal elements for language descriptions	149
25.1	LanguageDescription	149
25.2	LanguageDescriptionSubclass	149
25.3	MLModel	150
25.4	NGramModel	151
25.5	Grammar	151
26	Minimal elements for lexical/conceptual resources	153
26.1	LexicalConceptualResource	153

26.2	lcrSubclass	153
26.3	encodingLevel	154
26.4	ContentType	154
26.5	compliesWith	155
26.6	LexicalConceptualResourceTextPart	155
27	Minimal elements for projects	157
27.1	Project	157
27.2	ProjectIdentifier	157
27.3	projectName	158
27.4	projectShortName	158
27.5	projectAlternativeName	159
27.6	fundingType	159
27.7	funder	159
27.8	fundingCountry	160
27.9	website	160
27.10	logo	161
27.11	LTArea	161
27.12	domain	161
27.13	keyword	162
27.14	socialMediaOccupationalAccount	162
27.15	projectSummary	163
28	Minimal elements for organisations	165
28.1	Organization	165
28.2	OrganizationIdentifier	165
28.3	organizationName	166
28.4	organizationShortName	166
28.5	organizationAlternativeName	167
28.6	organizationBio	167
28.7	logo	168
28.8	LTArea	168
28.9	serviceOffered	169
28.10	domain	169
28.11	keyword	169
28.12	email	170
28.13	website	170
28.14	headOfficeAddress	171
28.15	socialMediaOccupationalAccount	171
28.16	hasDivision	171
29	API specifications	175
30	Internal LT Service API specification	177
30.1	Basic API pattern	178
30.2	Utility datatypes	178
30.3	Request structure	179
30.4	Response structure	183
30.5	Progress Reporting	186
30.6	Appendix: Standard status message codes	188
31	Public LT API specification	189
31.1	Input formats	189
31.2	Service responses	190
31.3	Asynchronous processing	191

32 Terms of use	193
33 Publications and reports	195
33.1 Scientific publications	195
33.2 Related publications	196
33.3 Deliverables	197
34 Indices and tables	199
Index	201

Welcome to the user manual of the European Language Grid.

CHAPTER 1

Introduction

The [European Language Grid \(ELG\)](#) is a catalogue and cloud 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, organisations, and groups*.

This manual aims to guide

- **consumers:** learn how to browse the ELG catalogue and find the language resources and technologies you need. Also find organisations and projects to connect with.
- **providers:** learn how to contribute your language resources and technologies to the ELG catalogue and how to host and sell access to your services via the ELG cloud.
- **moderators:** if you are part of the ELG technical team, learn how to review and ingest submitted language resources and technologies.

The current version of the manual documents the first official release of the ELG platform, launched in May 2020, which comes with a limited set of functionalities. More functionalities are continuously added, and this manual keeps on being updated following the evolution of the ELG platform.

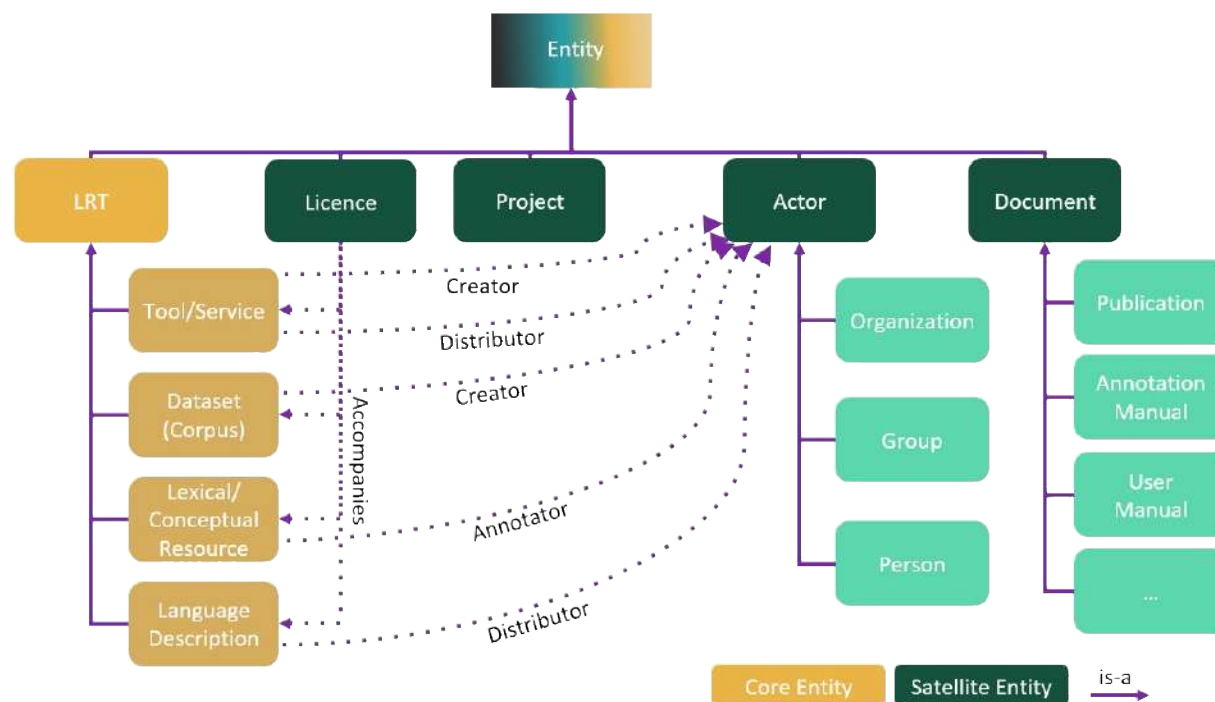
If you have any questions or want to share feedback, please send an email to contact@european-language-grid.eu.

Types of catalogue entries

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 entries:



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.

CHAPTER 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.

Browse the catalogue

4.1 View catalogue

You can

- browse through the catalogue and see all the entries,
- search for entries using the free text bar,
- filter entries by type, language, service function, license, etc.

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


4.2 View catalogue entry

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

The following figure shows the catalogue entry 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 the cloud): 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.



ALPHA

[Upload](#)
[Technologies](#)
[Resources](#)
[Community](#)
[Events](#)
[About ELG](#)
[Help](#)

Kilian

Clear all filters

455 search results

Language resources & technologies

+ Corpus (252)

+ Tool/Service (157)

+ Lexical/Conceptual resource (21)

+ Language description (7)

Languages

+ English (300)

+ Polish (64)

+ German (54)

+ French (52)

+ Modern Greek (1453-) (44)

Show more

Service functions

+ Part-of-Speech Tagging (72)

+ Tokenization (67)

+ Dependency parsing (60)

+ Lemmatization (60)

+ Morphological annotation (60)

Show more

Licences

+ Open under PSI (90)

+ Creative Commons Attribution Non Commercial Share Alike 4.0 International (64)

+ Creative Commons Attribution 4.0 International (63)

+ Mozilla Public License 2.0 (62)

+ Public Domain (44)

Show more

Related entities

+ Project (10)

+ Organization (8)

2006 CoNLL Shared Task - Ten Languages

Corpus

19 views

2 downloads

2006 CoNLL Shared Task - Ten Languages consists of dependency treebanks in ten languages used as part of the CoNLL 2006 shared task on multi-lingual dependency parsing. The languages covered in this release are: Bulgaria ...

Keyword:

corpus

Languages:

Swedish Danish Portuguese Spanish Turkish ...

Licences:

ELRA-END-USER-COMMERCIAL-MEMBER-NONCOMMERCIALUSE-1.0

ELRA-END-USER-COMMERCIAL-NOMEMBER-NONCOMMERCIALUSE-1.0 ...

2007 CoNLL Shared Task - Basque, Catalan, Czech & Turkish

Corpus

1 views

0 downloads

2007 CoNLL Shared Task - Basque, Catalan, Czech & Turkish consists of dependency treebanks in four languages used as part of the CoNLL 2007 shared task on multi-lingual dependency parsing and domain adaptation. The langu ...

Keyword:

corpus

Languages:

Catalan Czech Turkish Basque

Licences:

ELRA-END-USER-ACADEMIC-MEMBER-NONCOMMERCIALUSE-1.0

ELRA-END-USER-COMMERCIAL-MEMBER-NONCOMMERCIALUSE-1.0 ...

2007 CoNLL Shared Task - Greek, Hungarian & Italian

Corpus

2 views

0 downloads

2007 CoNLL Shared Task - Greek, Hungarian & Italian consists of dependency treebanks in three languages used as part of the CoNLL 2007 shared task on multi-lingual dependency parsing and domain adaptation. The languages ...

Keyword:

corpus

Languages:

Modern Greek (1453-) Italian Hungarian

Licences:

ELRA-END-USER-COMMERCIAL-MEMBER-NONCOMMERCIALUSE-1.0

ELRA-END-USER-COMMERCIAL-NOMEMBER-NONCOMMERCIALUSE-1.0 ...

BMI Brochures 2011-2015 (Processed)

Corpus

11 views

BMI Brochures 2011-2015 (Processed)

English translations of German BMI brochures from the last four years, in TMX format.

10

Chapter 4. Browse the catalogue


[← Back to catalogue](#)


Cogito Discover Named Entity Recognizer ESI_NER

Version: 14.3.0

ToolService

Overview

Download/Run

Test/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

Reconocimiento de entidades nombradas

Intended application

Named Entity Recognition

Resource provider

Expert System
Website

Additional info

[Landing page](#)

Contact

Gómez Pérez Jose Manuel



Input content resource

Language
English Spanish; Castilian
German French

Data format
JSON text/plain HTML XML

Processing resource type
file

[more](#)



Function

Function
Named Entity Recognition

Language dependent
true



Output resource

Language
English

Data format
JSON

Processing resource type
file

[more](#)

Documentations

Is documented by
[Cogito Discover Services Documentation](#)

Resource creator



Expert System
Website

Evaluated

Evaluated: true

TRL: TRL9



[← Back to catalogue](#)



INTERA English-Slovene SVEZ ACQUIS Corpus

Version: v1.0.0 (automatically assigned)

Corpus

Overview

Download

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

Keyword

corpus

Intended application

machineTranslation

Domain

law

Corpus subclass

annotated corpus

Funding project

[Integrated European language data Repository Area](#)

Additional info

Contact

Gavrilidou Maria

Relations to other resources

more

Corpus parts

TEXT	Linguality type	bilingual	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
	Multilinguality type	parallel	
	Language	English Slovenian	
	Modality type	written language	

Documentations

Is documented by
[Building Multilingual Terminological Resources](#)

Is documented by
[Building parallel corpora for eContent professionals](#)

Is documented by
[Language resources production models: the case of INTERA multilingual corpus and terminology](#)

Is documented by
[D5.2 - Report on the multilingual resources production](#)

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

Actual use

Used in application:
[terminologyExtraction](#)

Actual use details
[nlpApplications](#)

The following figure shows the entry of a **corpus**.

There are three 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.
- **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.
- **General:** This tab appears only for resources with a rich description and is used so as not to make the first tab too long and difficult to read.

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

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



[Upload](#)
[Technologies](#)
[Resources](#)
[Community](#)
[Events](#)
[About ELG](#)
[Help](#)

Kilian

[← Back to catalogue](#)



CEPLEXicon

Version: 1.0 (2015-04-15)

LexicalConceptualResource

[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.clul.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

Encoding level
 syntax

LCR parts

TEXT	Linguality type
	monolingual
	Language
	Portuguese

Funding project

[Complement Clauses in the Acquisition of Portuguese \(PTDC/CLE-LIN/120897/2010\)](#)

Additional info

[Landing page](#)

[Upload](#)[Technologies](#)[Resources](#)[Community](#)[Events](#)[About ELG](#)[Help](#)[Kilian](#) [← Back to catalogue](#)

Finance English grammar

Fin.en.grm

Version: v1.0.0 (automatically assigned)

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

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

Keyword

languageDescription

LD subclass

Grammar

Grammar details

Encoding level

morphology

LD parts



TEXT

Linguality type
monolingual

Language
English

Funding project

[Portdial](#)

Additional info

[Landing page](#)

Contact



Potamianos Alex


[Upload](#)
[Technologies](#)
[Resources](#)
[Community](#)
[Events](#)
[About ELG](#)
[Help](#)
[Killian](#)
[← Back to catalogue](#)

Athena Research Center ARC

Organization

[Overview](#)

Athena Research and Innovation Centre (ARC) is a scientific research and technological organisation, functioning under the auspices of the General Secretariat for Research and Technology (Greek Ministry of Education). It comprises 3 research Institutes: Institute for Language and Speech Processing (ILSP), dedicated to language technology research, development and innovation; Institute for the Management of Information Systems (IMIS), dedicated to data and information management, and Industrial Systems Institute (ISI) ... [Read More](#)

LT area

[Language Technology](#)

Has division



Institute for Language and
Speech Processing
ILSP

Division category
Institute

Organization description

The Institute for Language and Speech Processing (ILSP/Athena R.C.) is one of the institutes of Athena Research and Inno ... [Read More](#)

LT area

[Annotation](#)
[NLP Development Support](#)

Has division

Natural Language Processing and
Language Infrastructures
NLPLI

Division category
department

Organization description

NLPLI conducts basic and applied research in the fields of NLP and Knowledge Technologies. It designs, implements and in ... [Read More](#)

LT area

[Annotation](#)
[Machine translation support](#)
[Language resources infrastructures](#)

Organization information

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

Website

[Website](#)

Organization legal status


public organization

Organization role

LT user research organization


Address

Artemidos 6 and Epidavrou
Maroussi
Athens
151 25
GR



[Upload](#)
[Technologies](#)
[Resources](#)
[Community](#)
[Events](#)
[About ELG](#)
[Help](#)

[← Back to catalogue](#)



European Language Grid

Project

Overview

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

<http://w3id.org/meta-share/omtd-share/languageTechnology>

Coordinator



German Research Center for Artificial Intelligence
[Website](#)

Participants

Charles University	Website
SAIL LABS Technology	Website
Tilde	Website
THE UNIVERSITY OF SHEFFIELD	
EVALUATIONS AND LANGUAGE RESOURCES DISTRIBUTION AGENCY	
EXPERT SYSTEM IBERIA SL	
Athena Research Center	Website
University of Edinburgh	Website

Project information

Website

[Website](#)

Project start date


2019-01-01

Project end date

2021-12-31

Funder

Funder



European Commission
[Website](#)

Funding scheme category

IA

Funding country

European Union

Funding type

EU funds

Grant number: 825627

Status

SIGNED

Related call

H2020-ICT-2018-2

Related programme

H2020

Related subprogramme

ICT-29-2018

Amount

€7460206.25

EC max contribution

€6999631.25

CHAPTER 5

Register 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.

If you don't have an account yet, request one via our [contact form](#).

To sign in to your existing account, click on the user icon at the top right and follow the instructions.

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, or via ELG's API.

6.1 Try-out UI

The catalogue entry of a service has a tab called **Test/Try out**:

Test/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.

6.2 Call a service via the API

The catalogue entry 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** ALPHA

[Upload](#) [Technologies](#) [Resources](#) [Community](#) [Events](#) [About ELG](#) [Help](#) [Kilian](#)

[← Back to catalogue](#)

**ILSP Named Entity Recognizer for English** ToolService
ILSP-NER-English
Version: v1.0.0

[Overview](#) [Download/Run](#) [Test/Try out](#) [Code samples](#)

Type text to annotate

I moved to Berlin

SUBMIT

 ToolService

Version: v1.0.0

[Overview](#)
[Download/Run](#)
[Test/Try out](#)
[Code samples](#)

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


[illegible]

CHAPTER 7

Download a resource

You can download a resource provided in the ELG platform through the Download tab:


Downloading a resource is subject to agreeing with the licensing terms under which it is provided, while additional actions may also be required (for instance, access only for registered users):

**EUROPEAN
LANGUAGE
GRID**ALPHA

UploadTechnologiesResourcesCommunityEventsAbout ELGHelp

Kilian

← Back to catalogue

**2006 CoNLL Shared Task - Ten Languages**
Version: 1.0 (2015-12-02)

Corpus

OverviewDownload

D

Distribution

Dataset distribution form
downloadable


Text features
size
30 file

Data format
CoNLL-2006

Licence
ELRA-END-USER-ACADEMIC-MEMBER-NONCOMMERCIALUSE-1.0
http://catalogue.elra.info/static/from_media/metashare/licences/ELRA_END_USER.pdf

Membership institution
ELRA

Availability
2015-12-02 -

Distribution rights holder
 **ELRA**
[Website](#)

Download

D

Distribution

Dataset distribution form
downloadable


Text features
size
30 file

Data format
CoNLL-2006


Licence
ELRA-END-USER-COMMERCIAL-MEMBER-NONCOMMERCIALUSE-1.0
http://catalogue.elra.info/static/from_media/metashare/licences/ELRA_END_USER.pdf

Membership institution
ELRA

Availability
2015-12-02 -

Distribution rights holder
 **ELRA**
[Website](#)

Download


**EUROPEAN
LANGUAGE
GRID**ALPHA

UploadTechnologiesResourcesCommunityEventsAbout ELGHelp

Kilian

Accept the Licence Agreement

1 of 4Automatic Zoom



ELDA
EVALUATION AND LANGUAGE
RESOURCES DISTRIBUTION AGENCY

9, rue des Cordelières
75 013 Paris
Tél. : 01 43 13 39 39
Fax : 01 43 13 39 30
Web : www.elda.org
Email : info@elda.org

LANGUAGE RESOURCES END-USER AGREEMENT

(Agreement Ref. No. LC/ELDA/END-USER/2015/000/NAME)

This agreement is made by and between:

Membership institution
ELRA
Availability
2015-12-02 -

Membership institution
ELRA
Availability
2015-12-02 -

DisagreeAgree

CHAPTER 8

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.

CHAPTER 9

Register as a provider

In order to contribute to ELG, *Register as a user* to the platform and ask (by email to contact@european-language-grid.eu) to be granted “Provider” permissions.

Contribute a service

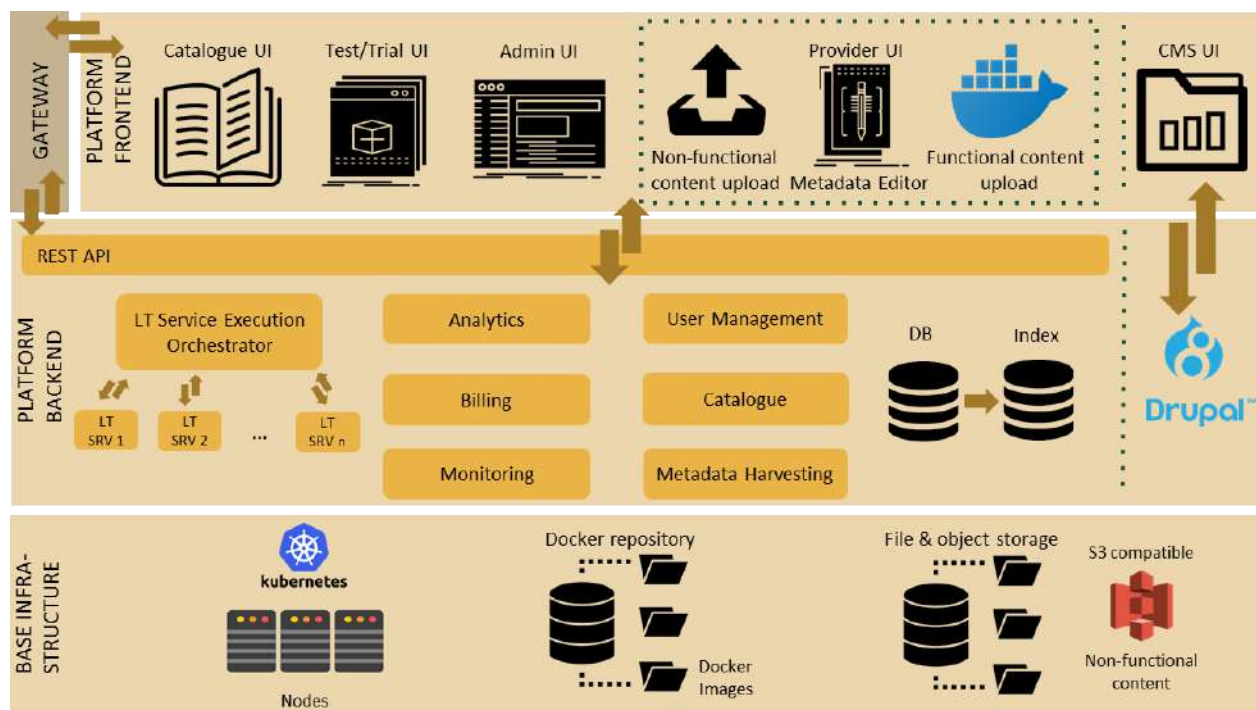
This page describes how to contribute a language technology *service* to run on the cloud platform of the European Language Grid.

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 organisations 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.

10.1 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 Test/Trial UIs that are embedded in the ELT Catalogue; however, 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 *Step 2: register the service with the platform* ('LT Service is deployed to ELG and configured' step).

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

- When 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 *Step 2: register the service with the platform* ('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.

10.2 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.

10.2.1 Technical requirements and integration options

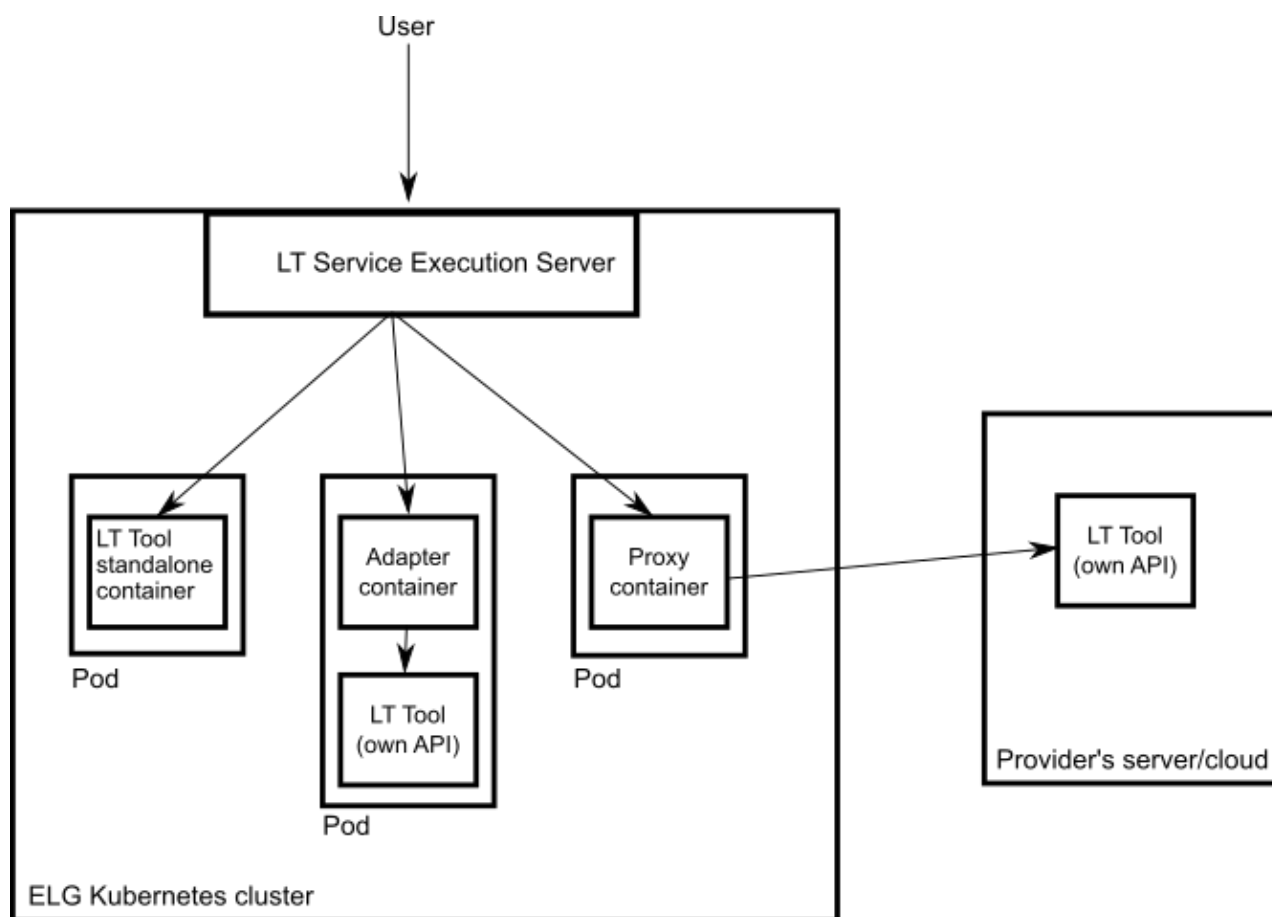
The requirements for integrating an LT tool/service are the following:

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 annex*.

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:



10.3 Step 0: Dockerize your service

10.3.1 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 on 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”); also when it completes successfully, you can 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 GitLab registry, in [DockerHub](#) (which is a public Docker registry) or any other Docker registry.

E.g 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. Before running `docker build` you have to download (clone) a copy of the project and be in 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](#).

10.3.2 Dockerization of a Python-based LT service/tool

From Python script to Service

First you need to ensure that your python script provides either a std i/o messaging or a RESTful API.

A) create a REST API using Flask Example of a [Shakespeare Bot](#)

```
from flask import Flask
from flask import render_template
from flask import request

# creates a Flask application, named app
app = Flask(__name__)

from chatterbot import ChatBot

# a route where we will display a welcome message via an HTML template
@app.route("/", methods=['POST', 'GET'])
def hello():
    if request.method == 'POST': # this block is only entered when the form is_
        ↪submitted
        user_message = request.form['user_message']
        chatbot = ChatBot("Frank")
        bot_message = chatbot.get_response(user_message).text

        data = {
            'bot_message': bot_message,
            'user_message': user_message,
            'user_message_visibility': '',
        }
        return render_template('index.html', **data)

    data = {
        'bot_message': "Speak. I am bound to hear.",
        'user_message': '',
        'user_message_visibility': 'style=visibility:hidden;',
    }
    return render_template('index.html', **data)
```

(continues on next page)

(continued from previous page)

```
# run the application
if __name__ == "__main__":
    app.run(debug=True)
```

B) create a std i/o messaging

In Python, this can easily be done via `print`.

Note: Example to be added soon.

From Python Service to Docker image

Then you can make your service a Docker image by taking the following steps:

- choose python environment, e.g. `python:3.6.4-slim-jessie` (see also: [Official Docker images](#))
- add/copy python scripts
- add/copy other resources
- install missing modules
- define entrypoint

We provide you with some dockerfile examples to see its simplicity:

Example 1: Shakespeare Bot

```
from python:3.6.4-slim-jessie

COPY shakespearebot.py .
COPY corpora/hamlet.csv corpora/

RUN pip install pandas
RUN pip install chatterbot
RUN pip install chatterbot-corpus

ENTRYPOINT ["python", "shakespearebot.py"]
```

Example 2: Legal Entity Recognition, install all requirements from .txt-file

```
FROM python:3.7

COPY . .

RUN pip install -r requirements.txt

EXPOSE 8080
ENTRYPOINT ["python", "ler-ws.py"]
```

10.3.3 Dockerization of a Java-based tool

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

10.4 Step 1: create metadata

The first step is to describe your software using ELG's metadata format, ELG-SHARE. Future releases of ELG will include an interactive editor for this. However, for now, you must create an XML file. Refer to the examples below for how to do this.

The elements you need are documented on the following pages:

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

For more information about ELG-SHARE, see:

- *Metadata schema*

At the ELG GitLab, you will find [templates](#) (that you can use to create new metadata records) and [examples](#) in XML format.

10.4.1 Example 1: information extraction service

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

```
<?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:MetadataRecordIdentifier ms:MetadataRecordIdentifierScheme="http://w3id.
↳ org/meta-share/meta-share/elg">default id</ms:MetadataRecordIdentifier>
  <ms:metadataCreationDate>2020-02-25</ms:metadataCreationDate>
  <ms:metadataLastDateUpdated>2020-02-25</ms:metadataLastDateUpdated>
  <ms:metadataCurator>
    <ms:actorType>Person</ms:actorType>
    <ms:surname xml:lang="en">Roberts</ms:surname>
    <ms:givenName xml:lang="en">Ian</ms:givenName>
    <ms:email>username1@somedomain.com</ms:email>
  </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">Roberts</ms:surname>
    <ms:givenName xml:lang="en">Ian</ms:givenName>
    <ms:email>username2@somedomain.com</ms:email>
  </ms:metadataCreator>
  <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 &lt;em>
↳ persons&lt;/em>, &lt;em>locations&lt;/em>, &lt;em>organizations&lt;/em>
↳ &lt;/em>, as well as &lt;em>money amounts&lt;/em>, &lt;em>time and date
↳ expressions&lt;/em> in English texts automatically. </ms:description>
↳ (continues on next page)
```

(continued from previous page)

```

<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:LTCClassRecommended>http://w3id.org/meta-share/
↪omtd-share/NamedEntityRecognition</ms:LTCClassRecommended>
</ms:intendedApplication>
<ms:LRS subclass>
  <ms:ToolService>
    <ms:lrType>ToolService</ms:lrType>
    <ms:function>
      <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: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>

```

(continues on next page)

(continued from previous page)

```

        </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>
        <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>

```

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

10.4.2 Example 2: machine translation service

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

```
<?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:MetadataRecordIdentifier ms:MetadataRecordIdentifierScheme="http://w3id.
↳ org/meta-share/meta-share/elg">default id</ms:MetadataRecordIdentifier>
  <ms:metadataCreationDate>2020-02-28</ms:metadataCreationDate>
  <ms:metadataLastDateUpdated>2020-02-28</ms:metadataLastDateUpdated>
  <ms:metadataCurator>
    <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: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">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:metadataCreator>
  <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:LRIdentifier ms:LRIdentifierScheme="http://w3id.org/meta-
↳ share/meta-share/elg">ELG id automatically assigned</ms:LRIdentifier>
      <ms:version>v1.0.0</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:LRS subclass>
    <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:additionalHWRRequirements>limits_
↪memory: 2048Mi limits_cpu: 1.5</ms:additionalHWRRequirements>
            <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:LicenceIdentifier
↪ms:LicenceIdentifierScheme="http://w3id.org/meta-share/meta-share/elg">ELG-ENT-LIC-
↪270220-00000097</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>de</
↪ms:languageTag>

```

(continues on next page)

(continued from previous page)

```

<ms:languageId>de</
↪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>
  </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>

```

The Docker image for this service is stored at DockerHub.

10.5 Step 2: register the service with the platform

From the ELG catalogue, click the “Upload” link as shown below:

Now upload the file you created in Step 1:

If there are any errors in your XML file, these will be shown to you. Fix them and try the upload again. Eventually, a success message will be shown to you and the metadata will be imported into the database.

10.6 Step 3: wait for approval

- **LT Service is assigned to a validator:** The administrator will assign it to a validator that will check mainly the metadata and technical compliance of the service; during the validation process, the metadata record is visible only to you (LT provider) and the validator.
- **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 screenshot shows the European Language Grid Catalogue search results page. The header includes the logo, navigation links (Administration, Upload, Technologies, Resources, Community, Events, About ELG, Help), and a user profile (Dimitris). A search bar with the text "Search for services, tools, datasets, organizations..." and a "Search" button is present. Below the search bar, there are filters for "Language resources & technologies" and "Languages". The search results section displays "480 search results" and lists several items, including "2006 CoNLL Shared Task - Ten Languages". A blue arrow points to the "Upload" link in the top navigation bar.

European Language Grid Catalogue

live.european-language-grid.eu/catalogue/#/

Administration Upload Technologies Resources Community Events About ELG Help

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

Search

Clear all filters

Language resources & technologies

480 search results

+ Corpus (252)

+ Tool/Service (151)

+ Lexical/Conceptual resource (21)

+ Language description (7)

Languages

+ English (304)

+ Polish (64)

+ German (54)

+ French (52)

2006 CoNLL Shared Task - Ten Languages

published

Corpus

2 views

2 downloads

2006 CoNLL Shared Task - Ten Languages consists of dependency treebanks in ten languages used as part of the CoNLL 2006 shared task on multi-lingual dependency parsing. The languages covered in this release are: Bulgaria ...

Keyword: corpus

Languages: Dutch Slovenian Spanish Turkish Portuguese ...

Licences: ELRA-END-USER-COMMERCIAL-NOMEMBER-NONCOMMERCIALUSE-1.0 ELRA-END-USER-COMMERCIAL-MEMBER-NONCOMMERCIALUSE-1.0 ...

https://live.european-language-grid.eu/catalogue/#/upload

The screenshot shows the European Language Grid Upload Resources page. The header includes the logo, navigation links (Administration, Upload, Technologies, Resources, Community, Events, About ELG, Help), and a user profile (Dimitris). A "Back to catalogue" link is visible. The main section is titled "Upload Resources" and contains a "1" icon, the text "Upload XML", and a large button labeled "Drag & Drop your files or Browse". Below the button, it says "Uploaded an XML file".

ELG - Upload

live.european-language-grid.eu/catalogue/#/upload

Administration Upload Technologies Resources Community Events About ELG Help

Back to catalogue

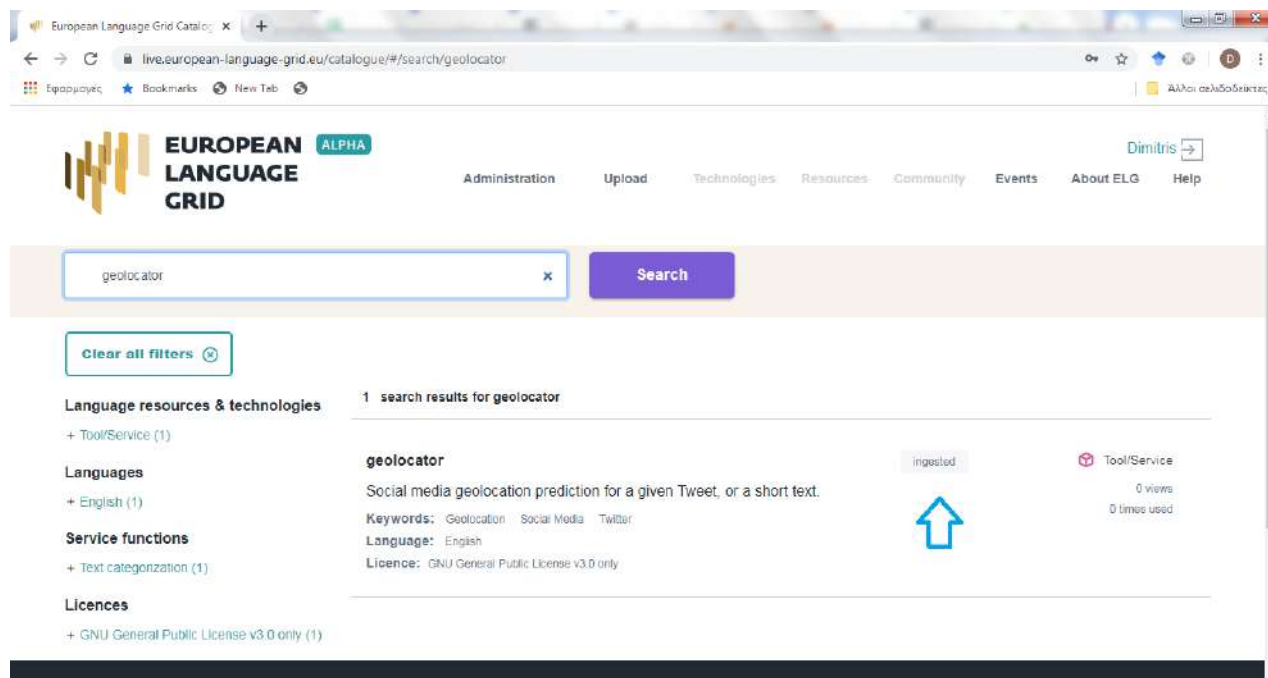
Upload Resources

1

Upload XML

Drag & Drop your files or Browse

Uploaded an XML file



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. The endpoint 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 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; both can 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.



[Upload](#)

[Technologies](#)

[Resources](#)

[Community](#)

[Events](#)

[About ELG](#)

[Help](#)

Kilian

[← Back to catalogue](#)



ILSP Named Entity Recognizer for English

ILSP-NER-English

Version: v1.0.0

ToolService

[Overview](#)

[Download/Run](#)

[Test/Try out](#)

[Code samples](#)

Type text to annotate

I moved to Berlin

SUBMIT

10.7 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).

Contribute downloadable software

This page describes how to contribute *downloadable software* to the European Language Grid.

11.1 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*.

11.2 Step 1: create metadata

The first step is to describe your software using ELG’s metadata format, ELG-SHARE. Future releases of ELG will include an interactive editor for this. However, for now, you must create an XML file.

The elements you need are documented on the following pages:

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

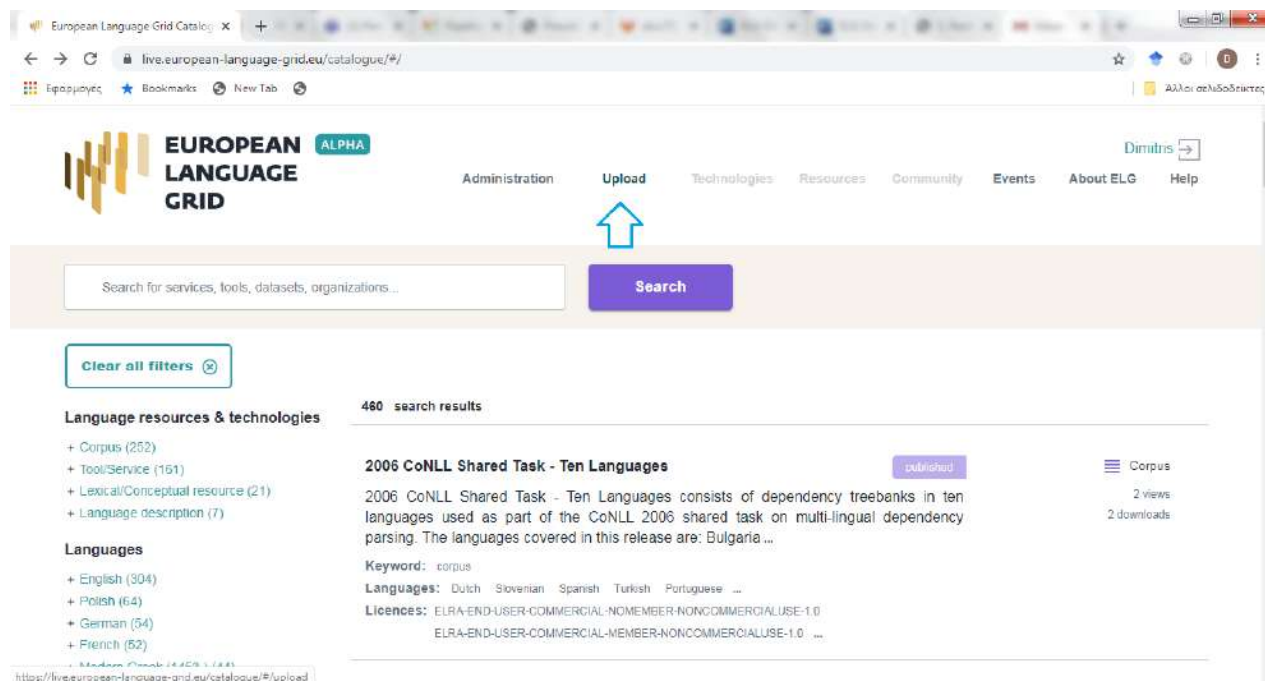
For more information about ELG-SHARE, see:

- *Metadata schema*

At the ELG GitLab, you will find *templates* (that you can use to create new metadata records) and *examples* in XML format.

11.3 Step 2: upload

From the ELG catalogue, click the “Upload” link as shown below:

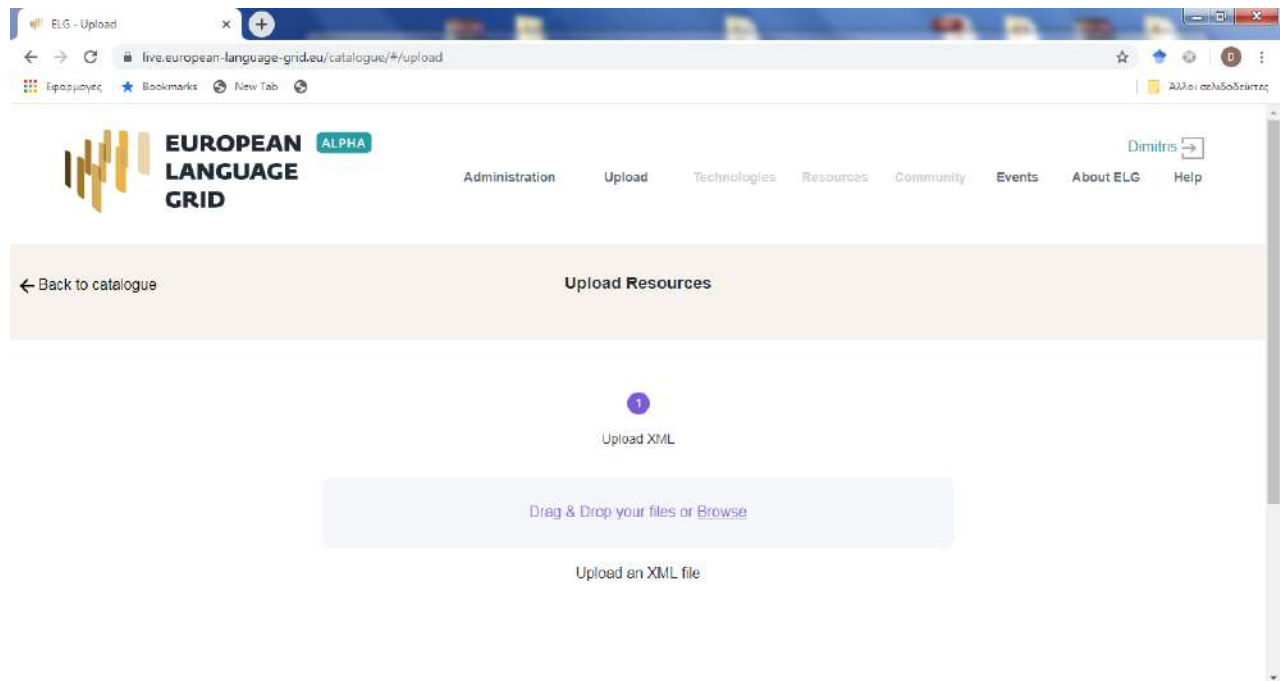


Now upload the file you created in Step 1:

If there are any errors in your XML file, these will be shown to you. Fix them and try the upload again. Eventually, a success message will be shown to you and the metadata will be imported into the database.

11.4 Step 3: wait for approval

At this stage, the software is only visible to you and to us, the ELG platform administrators. We will check your contribution and integrate it into the ELG catalogue if everything is in order, and contact you otherwise.



Contribute a corpus/dataset

This page describes how to contribute a *corpus* to the European Language Grid.

12.1 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*.
- For this release, you can provide the data files of the corpus
 - at a remote URL and include this information in the relevant metadata element (accessLocation), or
 - if you want us to upload it at the ELG cloud area, contact us through the [ELG contact form](#).

12.2 Step 1: create metadata

The first step is to describe your corpus using ELG’s metadata format, ELG-SHARE. Future releases of ELG will include an interactive editor for this. However, for now, you must create an XML file. Refer to the examples below for how to do this.

The elements you need are documented on the following pages:

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

For more information about ELG-SHARE, see:

- *Metadata schema*

At the ELG GitLab, you will find [templates](#) (that you can use to create new metadata records) and [examples](#) in XML format.

12.2.1 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="http://w3id.org/meta-share/meta-share/" xmlns:datacite=
↳ "http://purl.org/spar/datacite/" xmlns:dc="http://www.w3.org/ns/dc/" xmlns:ms=
↳ "http://w3id.org/meta-share/meta-share/" xmlns:omtd="http://w3id.org/meta-share/
↳ omtd-share/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
↳ xsi:schemaLocation="http://w3id.org/meta-share/meta-share/ ../Schema/ELG-SHARE.
↳ xsd">
<ms:MetadataRecordIdentifier ms:MetadataRecordIdentifierScheme="http://w3id.org/meta-
↳ share/meta-share/elg">value automatically assigned - leave as is</
↳ ms:MetadataRecordIdentifier>
  <ms:metadataCreationDate>2020-10-03</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:email>username@someDomain.com</ms:email>
  </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">Smith</ms:surname>
    <ms:givenName xml:lang="en">John</ms:givenName>
    <ms:email>username@someDomain.com</ms:email>
  </ms:metadataCreator>
  <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.
      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) de-
↳ duplication 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 l1/l2 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:description>
```

(continues on next page)

(continued from previous page)

```

<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">Rusinoval</
↪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>
    <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>

```

(continues on next page)

(continued from previous page)

```

        </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>Cyr</
↪ 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>
              <ms:creationMode>http://w3id.
↪ org/meta-share/meta-share/mixed</ms:creationMode>
              <ms:hasOriginalSource>
                <ms:resourceName>
↪ xml:lang="en">ELRC-447</ms:resourceName>
              <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>

```

(continues on next page)

(continued from previous page)

```

<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: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>

```

12.2.2 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="http://w3id.org/meta-share/meta-share/" xmlns:datacite=
↪"http://purl.org/spar/datacite/" xmlns:dc="http://www.w3.org/ns/dc#" xmlns:ms=
↪"http://w3id.org/meta-share/meta-share/" xmlns:omtd="http://w3id.org/meta-share/
↪omtd-share/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
↪xsi:schemaLocation="http://w3id.org/meta-share/meta-share/ ../Schema/ELG-SHARE.
↪xsd">
  <ms:MetadataRecordIdentifier ms:MetadataRecordIdentifierScheme="http://w3id.
↪org/meta-share/meta-share/elg">value automatically assigned - leave as is</
↪ms:MetadataRecordIdentifier>
  <ms:metadataCreationDate>2020-02-02</ms:metadataCreationDate>
  <ms:metadataCurator>

```

(continues on next page)

(continued from previous page)

```

        <ms:actorType>Person</ms:actorType>
        <ms:surname xml:lang="en">Smith</ms:surname>
        <ms:givenName xml:lang="en">John</ms:givenName>
        <ms:email>curator@somedomain.com</ms:email>
    </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">Smith</ms:surname>
        <ms:givenName xml:lang="en">John</ms:givenName>
        <ms:email>curator@somedomain.com</ms:email>
    </ms:metadataCreator>
    <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 consits 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>

```

(continues on next page)

(continued from previous page)

```

<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: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>

```

(continues on next page)

(continued from previous page)

```

↪org/meta-share/meta-share/mixed</ms:creationMode>      <ms:creationMode>http://w3id.
↪xml:lang="en">web news</ms:originalSourceDescription>  <ms:originalSourceDescription
↪xml:lang="en">EU texts</ms:originalSourceDescription>  <ms:originalSourceDescription
</ms:CorpusTextPart>
</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: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: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>
      <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>

```

(continues on next page)

(continued from previous page)

```

<ms:tagset>
  <ms:resourceName xml:lang="en
↪ ">ILSP/PAROLE tagset</ms:resourceName>
</ms:tagset>
<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:Corpus>
</ms:LRSubclass>
</ms:LanguageResource>
</ms:DescribedEntity>
</ms:MetadataRecord>

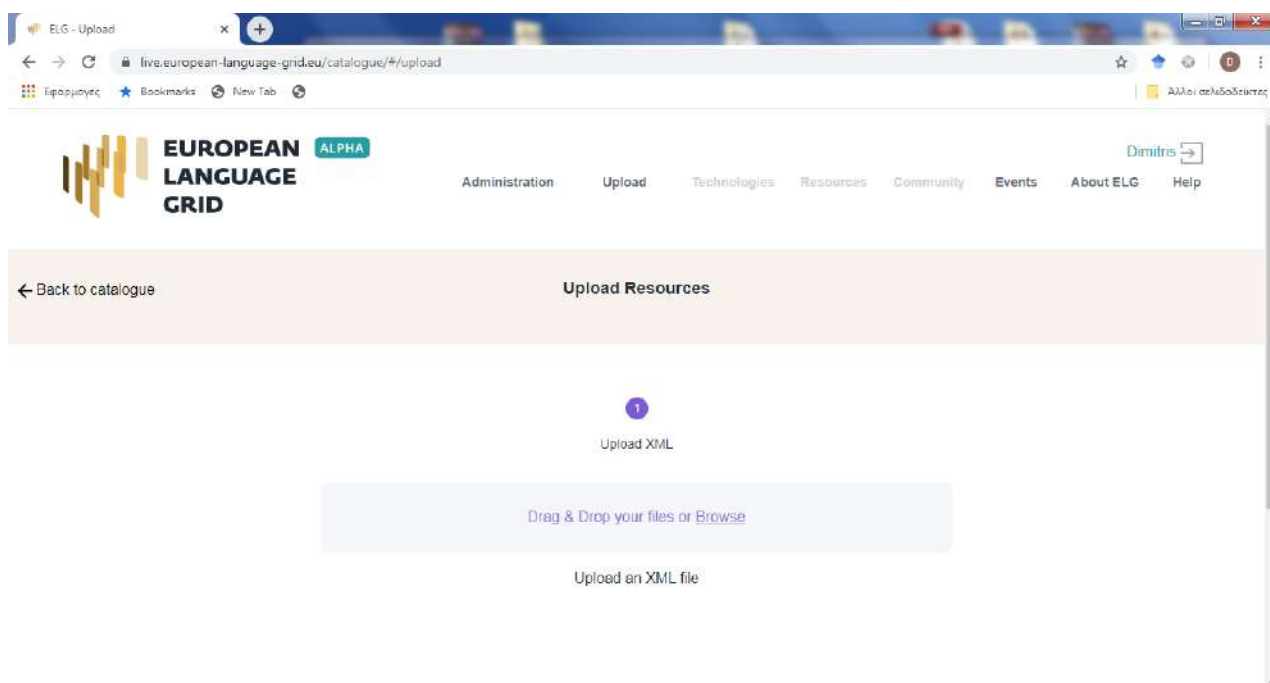
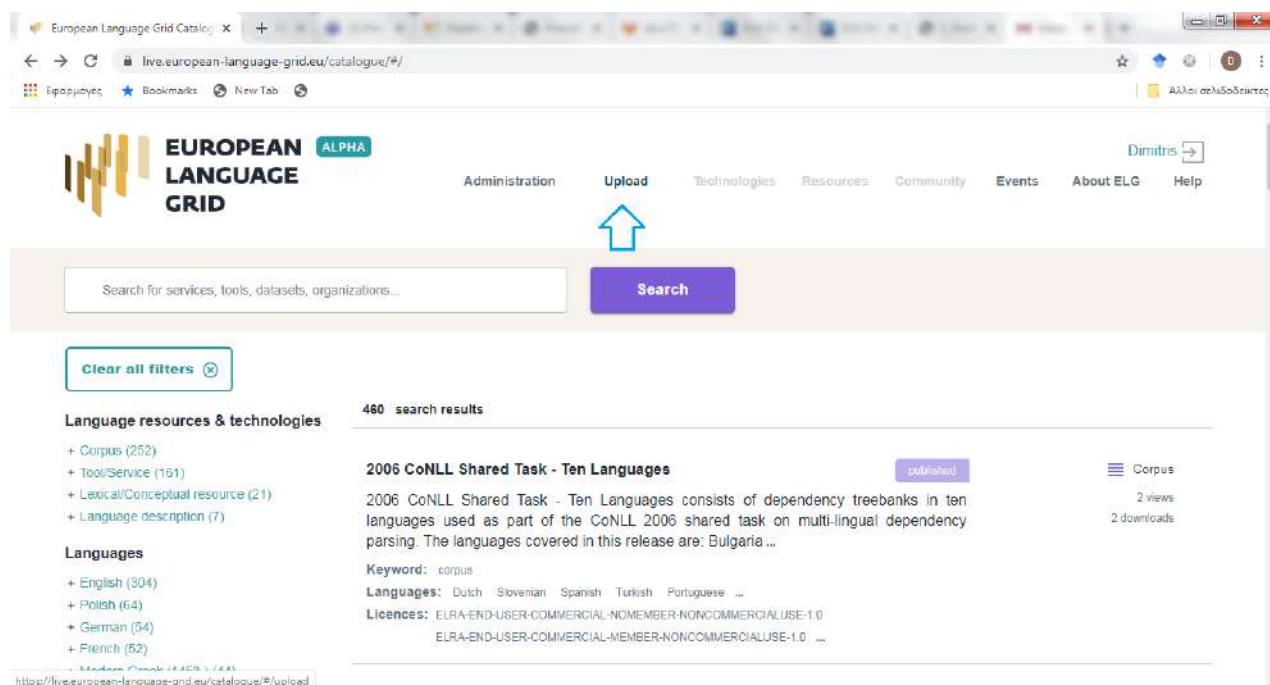
```

12.3 Step 2: upload

From the ELG catalogue, click the “Upload” link as shown below:

Now upload the file you created in Step 1:

If there are any errors in your XML file, these will be shown to you. Fix them and try the upload again. Eventually, a success message will be shown to you and the metadata will be imported into the database.



12.4 Step 3: wait for approval

At this stage, the metadata record is only visible to you and to us, the ELG platform administrators. We will check your contribution and integrate it into the ELG catalogue if everything is in order, and contact you otherwise.

This page describes how to contribute a *model* to the European Language Grid.

13.1 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*.
- For this release, you can provide the data files of the model
 - at a remote URL and include this information in the relevant metadata element (accessLocation), or
 - if you want us to upload it at the ELG cloud area, contact us through the [ELG contact form](#).

13.2 Step 1: create metadata

The first step is to describe your model using ELG’s metadata format, ELG-SHARE. Future releases of ELG will include an interactive editor for this. However, for now, you must create an XML file. Refer to the example below for how to do this.

The elements you need are documented on the following pages:

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

For more information about ELG-SHARE, see:

- *Metadata schema*

At the ELG GitLab, you will find [templates](#) (that you can use to create new metadata records) and [examples](#) in XML format.

13.2.1 Example: N-gram model

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

```
<?xml version="1.0" encoding="UTF-8"?>
<ms:MetadataRecord xmlns="http://w3id.org/meta-share/meta-share/" xmlns:datacite=
  ↪ "http://purl.org/spar/datacite/" xmlns:dc="http://www.w3.org/ns/dcat#" xmlns:ms=
  ↪ "http://w3id.org/meta-share/meta-share/" xmlns:omtd="http://w3id.org/meta-share/
  ↪ omtd-share/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  ↪ xsi:schemaLocation="http://w3id.org/meta-share/meta-share/ ../Schema/ELG-SHARE.
  ↪ xsd">
  <ms:MetadataRecordIdentifier ms:MetadataRecordIdentifierScheme="http://w3id.
  ↪ org/meta-share/meta-share/elg">value automatically assigned - leave as is</
  ↪ ms:MetadataRecordIdentifier>
  <ms:metadataCreationDate>2020-10-03</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:email>username@someDomain.com</ms:email>
  </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">Smith</ms:surname>
    <ms:givenName xml:lang="en">John</ms:givenName>
    <ms:email>username@someDomain.com</ms:email>
  </ms:metadataCreator>
  <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>v1.0</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>
```

(continues on next page)

(continued from previous page)

```

        <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>
        <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:LRSubclass>
      <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>

```

(continues on next page)

(continued from previous page)

```

<ms:languageId>el</
ms:languageId>
</ms:language>
<ms:metalanguage>
  <ms:languageTag>und</
ms:languageTag>
  <ms:languageId>und</
ms:languageId>
</ms:metalanguage>
<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>

```

(continues on next page)

(continued from previous page)

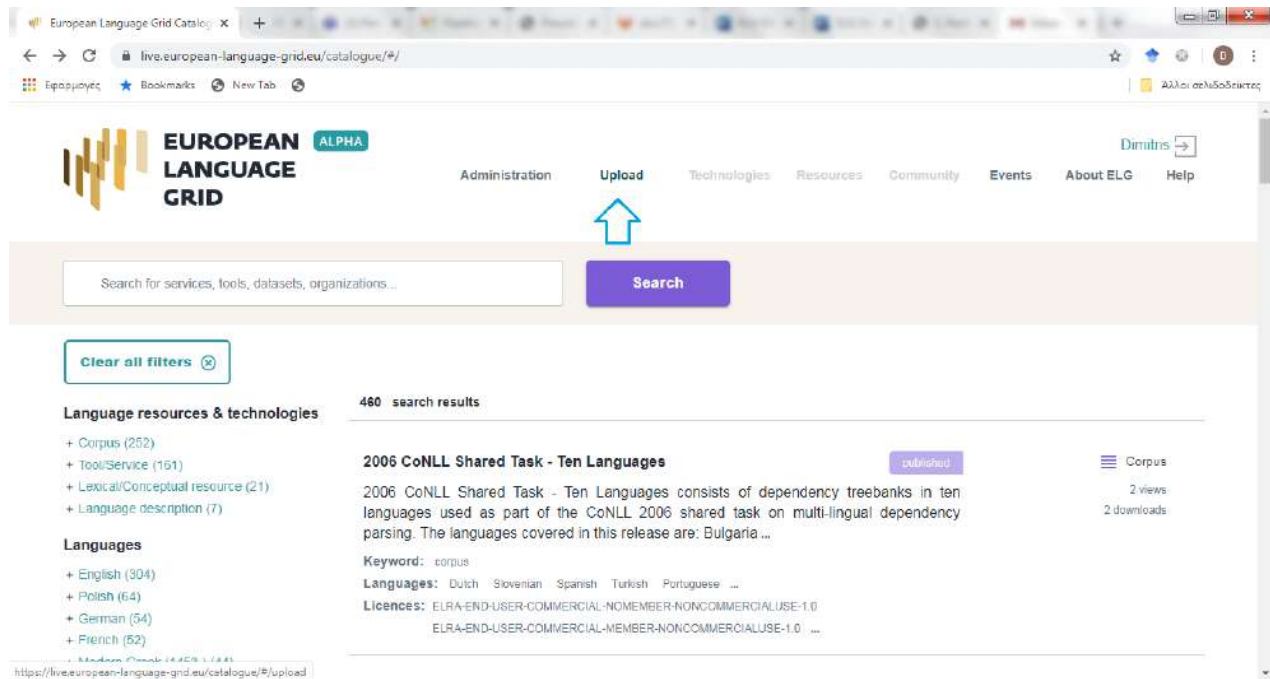
```

<ms:licenceTermsURL>https://
→spx.org/licenses/CC-BY-SA-4.0.html</ms:licenceTermsURL>
</ms:licenceTerms>
<ms:attributionText xml:lang="en">
→This LR has been created by Athena R.C./ILSP (www.ilsp.gr) and is licensed under a
→CC-BY-SA licence</ms:attributionText>
</ms:DatasetDistribution>
<ms:personalDataIncluded>>false</
→ms:personalDataIncluded>
<ms:sensitiveDataIncluded>false</
→ms:sensitiveDataIncluded>
</ms:LanguageDescription>
</ms:LRSubclass>
</ms:LanguageResource>
</ms:DescribedEntity>
</ms:MetadataRecord>

```

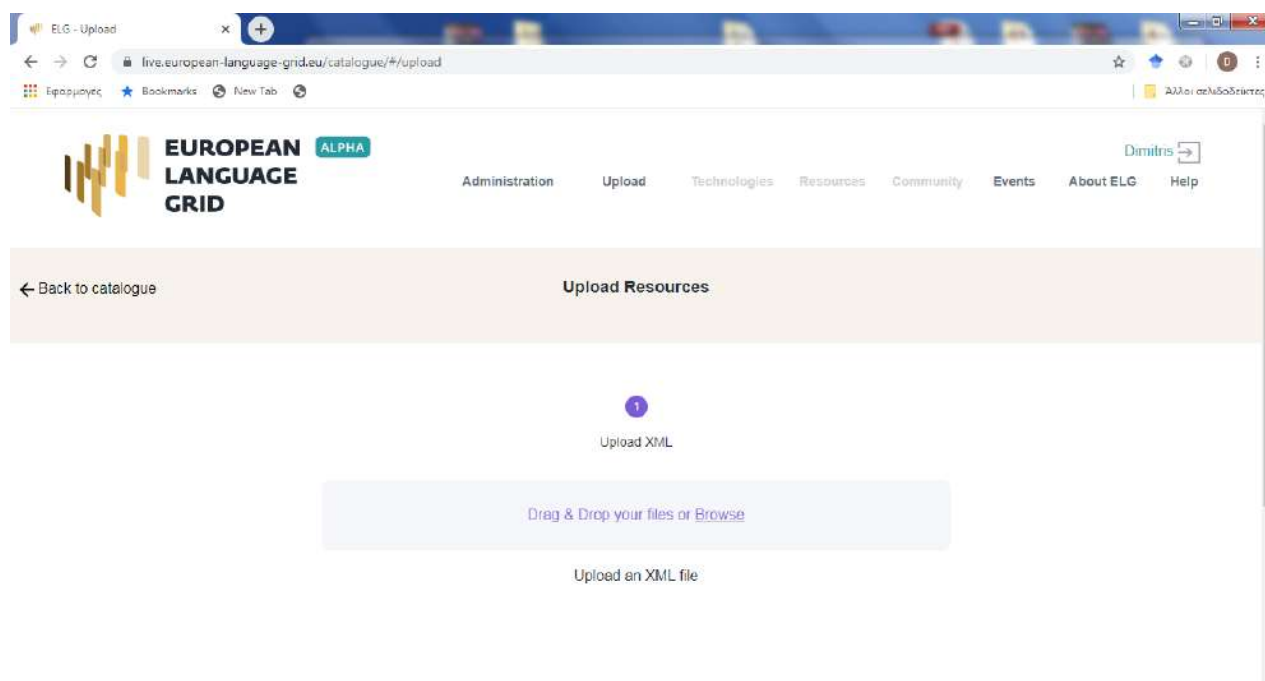
13.3 Step 2: upload

From the ELG catalogue, click the “Upload” link as shown below:



Now upload the file you created in Step 1:

If there are any errors in your XML file, these will be shown to you. Fix them and try the upload again. Eventually, a success message will be shown to you and the metadata will be imported into the database.



13.4 Step 3: wait for approval

At this stage, the metadata record is only visible to you and to us, the ELG platform administrators. We will check your contribution and integrate it into the ELG catalogue if everything is in order, and contact you otherwise.

Contribute a grammar

This page describes how to contribute a *grammar* to the European Language Grid.

14.1 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*.
- For this release, you can provide the data files of the grammar
 - at a remote URL and include this information in the relevant metadata element (accessLocation), or
 - if you want us to upload it at the ELG cloud area, contact us through the [ELG contact form](#).

14.2 Step 1: create metadata

The first step is to describe your grammar using ELG's metadata format, ELG-SHARE. Future releases of ELG will include an interactive editor for this. However, for now, you must create an XML file. Refer to the examples below for how to do this.

The elements you need are documented on the following pages:

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

For more information about ELG-SHARE, see:

- *Metadata schema*

At the ELG GitLab, you will find [templates](#) (that you can use to create new metadata records) and [examples](#) in XML format.

14.2.1 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="http://w3id.org/meta-share/meta-share/" xmlns:datacite=
↳ "http://purl.org/spar/datacite/" xmlns:dc="http://www.w3.org/ns/dcat#" xmlns:ms=
↳ "http://w3id.org/meta-share/meta-share/" xmlns:omtd="http://w3id.org/meta-share/
↳ omtd-share/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
↳ xsi:schemaLocation="http://w3id.org/meta-share/meta-share/ ../../Schema/ELG-SHARE.
↳ xsd">
    <ms:MetadataRecordIdentifier ms:MetadataRecordIdentifierScheme="http://w3id.
↳ org/meta-share/meta-share/elg">value automatically assigned - leave as is</
↳ ms:MetadataRecordIdentifier>
    <ms:metadataCreationDate>2020-10-03</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:email>username@someDomain.com</ms:email>
    </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">Smith</ms:surname>
        <ms:givenName xml:lang="en">John</ms:givenName>
        <ms:email>username@someDomain.com</ms:email>
    </ms:metadataCreator>
    <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>v1.0.0 (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>
                    <ms:email>contact@someDomain.com</ms:email>
                </ms:Person>
            </ms:contact>
```

(continues on next page)

(continued from previous page)

```

<ms:keyword xml:lang="en">languageDescription</ms:keyword>
<ms:fundingProject>
  <ms:projectName xml:lang="en">Portdial</
↪ms:projectName>
  </ms:fundingProject>
<ms:LRSubclass>
  <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:licenceTerms>
      </ms:DatasetDistribution>
      <ms:personalDataIncluded>>false</
↪ms:personalDataIncluded>
      <ms:sensitiveDataIncluded>>false</
↪ms:sensitiveDataIncluded>
    </ms:LanguageDescription>
  </ms:LRSubclass>
</ms:LanguageResource>

```

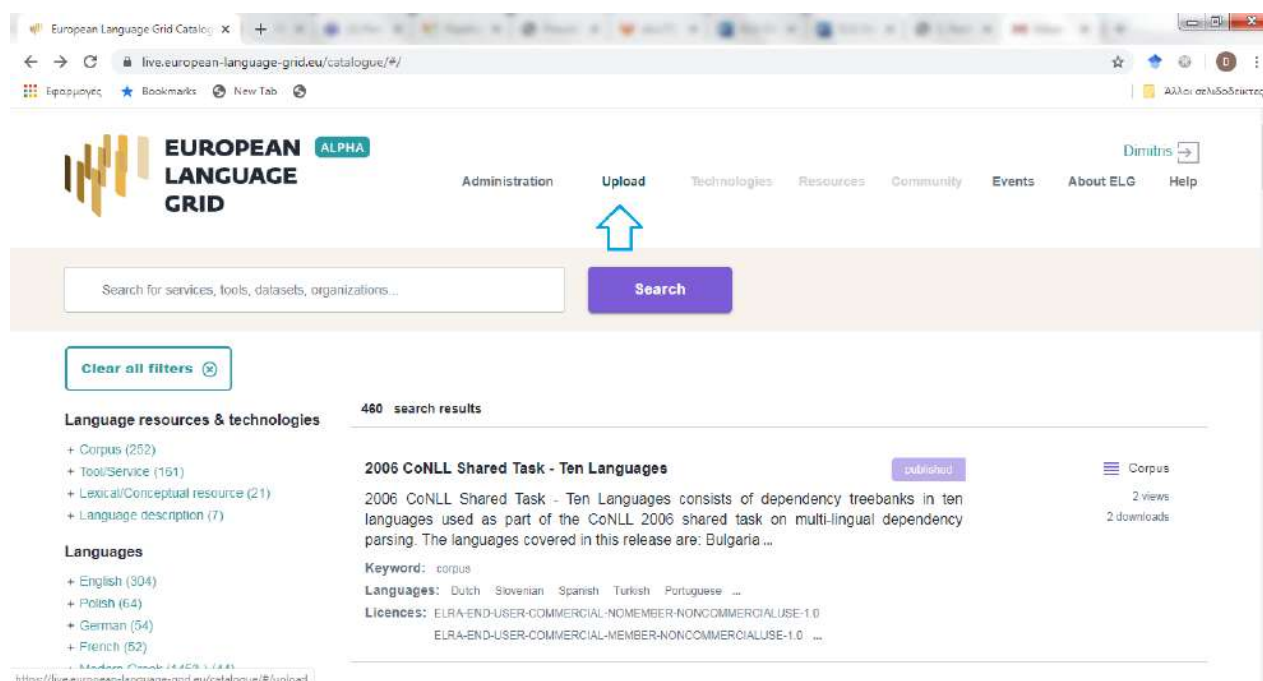
(continues on next page)

(continued from previous page)

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

14.3 Step 2: upload

From the ELG catalogue, click the “Upload” link as shown below:

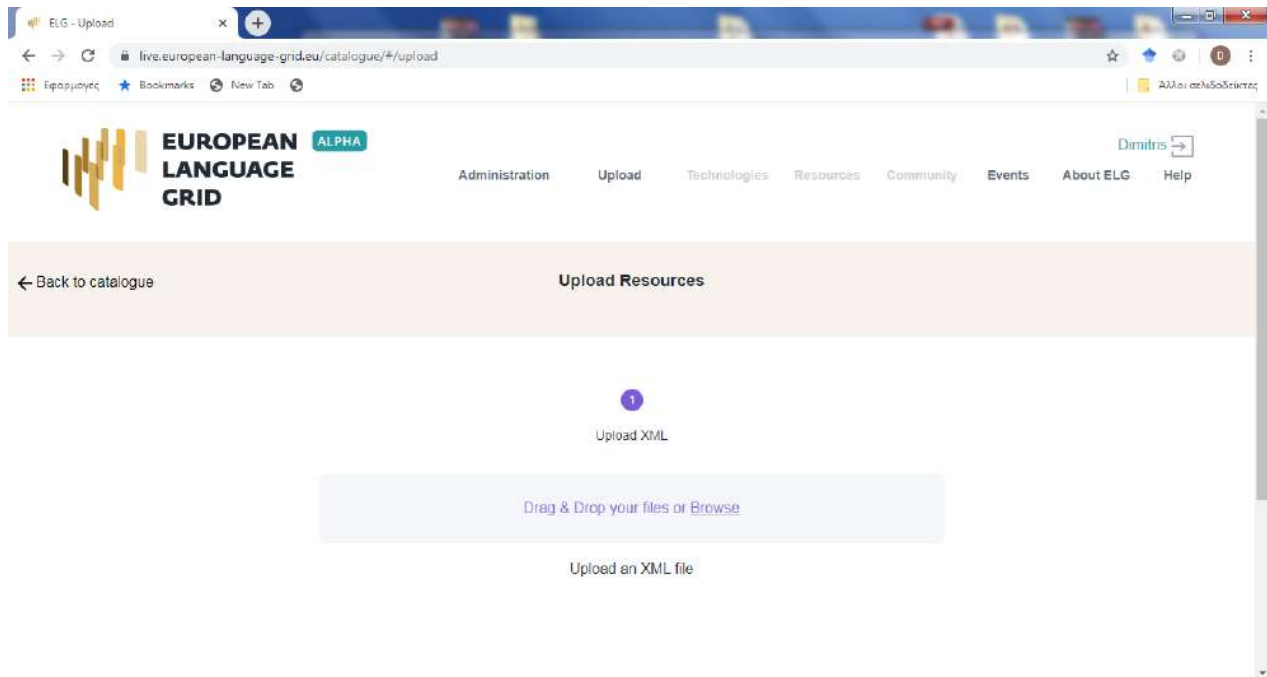


Now upload the file you created in Step 1:

If there are any errors in your XML file, these will be shown to you. Fix them and try the upload again. Eventually, a success message will be shown to you and the metadata will be imported into the database.

14.4 Step 3: wait for approval

At this stage, the metadata record is only visible to you and to us, the ELG platform administrators. We will check your contribution and integrate it into the ELG catalogue if everything is in order, and contact you otherwise.



Contribute a lexical/conceptual resource

This page describes how to contribute a *lexical/conceptual resource* to the European Language Grid.

15.1 Before you start

- Please make sure that the lexical/conceptual resources you want to contribute complies with our *terms of use*.
- Please make sure you have *registered* and been assigned the *provider role*.
- For this release, you can provide the data files of the lexical/conceptual resource
 - at a remote URL and include this information in the relevant metadata element (`accessLocation`), or
 - if you want us to upload it at the ELG cloud area, contact us through the [ELG contact form](#).

15.2 Step 1: create metadata

The first step is to describe your lexical/conceptual resource using ELG's metadata format, ELG-SHARE. Future releases of ELG will include an interactive editor for this. However, for now, you must create an XML file. Refer to the examples below for how to do this.

The elements you need are documented on the following pages:

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

For more information about ELG-SHARE, see:

- *Metadata schema*

At the ELG GitLab, you will find [templates](#) (that you can use to create new metadata records) and [examples](#) in XML format.

15.2.1 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="http://w3id.org/meta-share/meta-share/" xmlns:datacite=
↳ "http://purl.org/spar/datacite/" xmlns:dc="http://www.w3.org/ns/dcat#" xmlns:ms=
↳ "http://w3id.org/meta-share/meta-share/" xmlns:omtd="http://w3id.org/meta-share/
↳ omtd-share/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
↳ xsi:schemaLocation="http://w3id.org/meta-share/meta-share/ ../../Schema/ELG-SHARE.
↳ xsd">
    <ms:MetadataRecordIdentifier ms:MetadataRecordIdentifierScheme="http://w3id.
↳ org/meta-share/meta-share/elm">value automatically assigned - leave as is</
↳ ms:MetadataRecordIdentifier>
    <ms:metadataCreationDate>2020-02-02</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:email>curator@somedomain.com</ms:email>
    </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">Smith</ms:surname>
        <ms:givenName xml:lang="en">John</ms:givenName>
        <ms:email>curator@somedomain.com</ms:email>
    </ms:metadataCreator>
    <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>v1.0.0 (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">Gavrillidou</
↳ 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>
            <ms:domain>
                <ms:categoryLabel xml:lang="en">education</
↳ ms:categoryLabel>
```

(continues on next page)

(continued from previous page)

```

</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>
      <ms:encodingLevel>http://w3id.org/meta-share/
↪ meta-share/morphology</ms:encodingLevel>

```

(continues on next page)

(continued from previous page)

```

        <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:LicenceIdentifier
↳ ms:LicenceIdentifierScheme="http://w3id.org/meta-share/meta-share/elg">ELG-ENT-LIC-
↳ 270220-00000072</ms:LicenceIdentifier>

```

(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>

```

15.2.2 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:MetadataRecordIdentifier ms:MetadataRecordIdentifierScheme="http://w3id.
↪org/meta-share/meta-share/elg">value automatically assigned - leave as is</
↪ms:MetadataRecordIdentifier>
  <ms:metadataCreationDate>2005-05-12</ms:metadataCreationDate>
  <ms:metadataLastDateUpdated>2020-02-24</ms:metadataLastDateUpdated>
  <ms:metadataCurator>
    <ms:actorType>Person</ms:actorType>
    <ms:surname xml:lang="en">Smith</ms:surname>
    <ms:givenName xml:lang="en">John</ms:givenName>
    <ms:email>curator@somedomain.com</ms:email>
  </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">Smith</ms:surname>
    <ms:givenName xml:lang="en">John</ms:givenName>
    <ms:email>curator@somedomain.com</ms:email>
  </ms:metadataCreator>
  <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 morphosyn ↪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>

```

(continues on next page)

(continued from previous page)

```

    <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>
    <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>

```

(continues on next page)

(continued from previous page)

```

</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: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: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>

```

(continues on next page)

(continued from previous page)

```

        </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-MEMBER-COMMERCIALUSE-1.0</ms:licenceTermsName>
        <ms:licenceTermsURL>http://
↪www.elra.info/licenses/ELRA-VAR-COMMERCIAL-MEMBER-COMMERCIALUSE-1.0.html</
↪ms:licenceTermsURL>
      </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-NONCOMMERCIALUS
↪ms:licenceTermsURL>

```

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

  <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>

```

(continues on next page)

(continued from previous page)

```

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

```

(continues on next page)

(continued from previous page)

```

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

```

15.3 Step 2: upload

From the ELG catalogue, click the “Upload” link as shown below:

Now upload the file you created in Step 1:

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

The screenshot shows the European Language Grid Catalogue search results page. The header includes the logo, 'ALPHA' badge, and navigation links: Administration, Upload, Technologies, Resources, Community, Events, About ELG, and Help. A search bar with the placeholder 'Search for services, tools, datasets, organizations...' and a 'Search' button is present. Below the search bar, there are filters for 'Language resources & technologies' and 'Languages'. The search results section shows '460 search results' and a list of resources, including '2006 CoNLL Shared Task - Ten Languages'. A blue arrow points to the 'Upload' link in the header.

European Language Grid Catalogue

live.european-language-grid.eu/catalogue/#/

European Language GRID ALPHA

Administration Upload Technologies Resources Community Events About ELG Help

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

Search

Clear all filters

Language resources & technologies

460 search results

2006 CoNLL Shared Task - Ten Languages

published

Corpus

2 views

2 downloads

Keyword: corpus

Languages: Dutch Slovenian Spanish Turkish Portuguese ...

Licences: ELRA-END-USER-COMMERCIAL-NOMEMBER-NONCOMMERCIALUSE-1.0 ELRA-END-USER-COMMERCIAL-MEMBER-NONCOMMERCIALUSE-1.0 ...

https://live.european-language-grid.eu/catalogue/#/upload

The screenshot shows the European Language Grid Catalogue Upload Resources page. The header is the same as the previous screenshot. The main content area has a 'Back to catalogue' link and a 'Upload Resources' section. There is a button labeled 'Upload XML' and a large box with the text 'Drag & Drop your files or Browse'. Below this box, it says 'Uploaded an XML file'.

ELG - Upload

live.european-language-grid.eu/catalogue/#/upload

European Language GRID ALPHA

Administration Upload Technologies Resources Community Events About ELG Help

Back to catalogue

Upload Resources

1

Upload XML

Drag & Drop your files or Browse

Uploaded an XML file

success message will be shown to you and the metadata will be imported into the database.

15.4 Step 3: wait for approval

At this stage, the metadata record is only visible to you and to us, the ELG platform administrators. We will check your contribution and integrate it into the ELG catalogue if everything is in order, and contact you otherwise.

Contribute a project

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

16.1 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*.

16.2 Step 1: create metadata

The first step is to describe your project using ELG's metadata format, ELG-SHARE. Future releases of ELG will include an interactive editor for this. However, for now, you must create an XML file. Refer to the examples below for how to do this.

The elements you need are documented on the following pages:

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

For more information about ELG-SHARE, see:

- *Metadata schema*

At the ELG GitLab, you will find *templates* (that you can use to create new metadata records) and *examples* in XML format.

16.2.1 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:MetadataRecordIdentifier ms:MetadataRecordIdentifierScheme="http://w3id.org/meta-share/meta-share/elg">value automatically assigned - leave as is</ms:MetadataRecordIdentifier>
  <ms:metadataCreationDate>2020-01-07</ms:metadataCreationDate>
  <ms:metadataLastDateUpdated>2020-01-07</ms:metadataLastDateUpdated>
  <!-- the metadataCurator is the person responsible for editing/updating the metadata record in the ELG system and maybe different from metadata creator (for metadata records harvested from other repos, there will be no metadata creator) -->
  <ms:metadataCurator>
    <ms:actorType>Person</ms:actorType>
    <ms:surname xml:lang="en">Smith</ms:surname>
    <ms:givenName xml:lang="en">John</ms:givenName>
    <!-- please add an identifier (preferably ORCID in the format below) and/or email -->
    <ms:PersonalIdentifier ms:PersonalIdentifierScheme="http://purl.org/spar/datacite/orcid">0000-0000-0000-0000</ms:PersonalIdentifier>
    <ms:email>smith@example.com</ms:email>
  </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">Smith</ms:surname>
    <ms:givenName xml:lang="en">John</ms:givenName>
    <!-- please add an identifier (preferably ORCID in the format below) and/or email -->
    <ms:PersonalIdentifier ms:PersonalIdentifierScheme="http://purl.org/spar/datacite/orcid">0000-0000-0000-0000</ms:PersonalIdentifier>
    <ms:email>smith@example.com</ms:email>
  </ms:metadataCreator>
  <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:fundingCountry>EU</ms:fundingCountry>
      <ms:projectStartDate>2019-01-01</ms:projectStartDate>
      <ms:projectEndDate>2021-12-31</ms:projectEndDate>
    </ms:Project>
  </ms:DescribedEntity>
</ms:MetadataRecord>

```

(continues on next page)

(continued from previous page)

```

<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:LTCClassRecommended>http://w3id.org/meta-share/
↳omtd-share/MachineTranslation</ms:LTCClassRecommended>
</ms:LTArea>
<ms:LTArea>
  <ms:LTCClassOther>Browser-based Machine Translation</
↳ms:LTCClassOther>
</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>
  <ms:status>SIGNED</ms:status>
  <ms:relatedCall>H2020-ICT-2018-2</ms:relatedCall>
  <ms:relatedProgramme>H2020</ms:relatedProgramme>

```

(continues on next page)

(continued from previous page)

```

        <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>

```

16.2.2 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:MetadataRecordIdentifier ms:MetadataRecordIdentifierScheme="http://w3id.
↪org/meta-share/meta-share/elg">value automatically assigned - leave as is</
↪ms:MetadataRecordIdentifier>
    <ms:metadataCreationDate>2020-01-07</ms:metadataCreationDate>
    <ms:metadataLastDateUpdated>2020-01-07</ms:metadataLastDateUpdated>
    <!-- the metadataCurator is the person responsible for editing/updating the
↪metadata record in the ELG system and maybe different from metadata creator (for
↪metadata records harvested from other repos, there will be no metadata creator) -->
    <ms:metadataCurator>
        <ms:actorType>Person</ms:actorType>
        <ms:surname xml:lang="en">Smith</ms:surname>
        <ms:givenName xml:lang="en">John</ms:givenName>
        <!-- please add an identifier (preferably ORCID in the format below)
↪and/or email -->

```

(continues on next page)

(continued from previous page)

```

<ms:PersonalIdentifier ms:PersonalIdentifierScheme="http://purl.org/
→spar/datacite/orcid">0000-0000-0000-0000</ms:PersonalIdentifier>
<ms:email>smith@example.com</ms:email>
</ms:metadataCreator>
<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">Smith</ms:surname>
  <ms:givenName xml:lang="en">John</ms:givenName>
  <!-- please add an identifier (preferably ORCID in the format below)
→and/or email -->
  <ms:PersonalIdentifier ms:PersonalIdentifierScheme="http://purl.org/
→spar/datacite/orcid">0000-0000-0000-0000</ms:PersonalIdentifier>
  <ms:email>smith@example.com</ms:email>
</ms:metadataCreator>
<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>
    <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 βC" by nation states, languages, verticals
→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.

```

(continues on next page)

(continued from previous page)

```

        <ms:cost>
          <ms:amount>7460206.25</ms:amount>
          <ms:currency>http://w3id.org/meta-share/meta-share/
↪euro</ms:currency>
        </ms:cost>
        <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: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: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>

```

(continues on next page)

(continued from previous page)

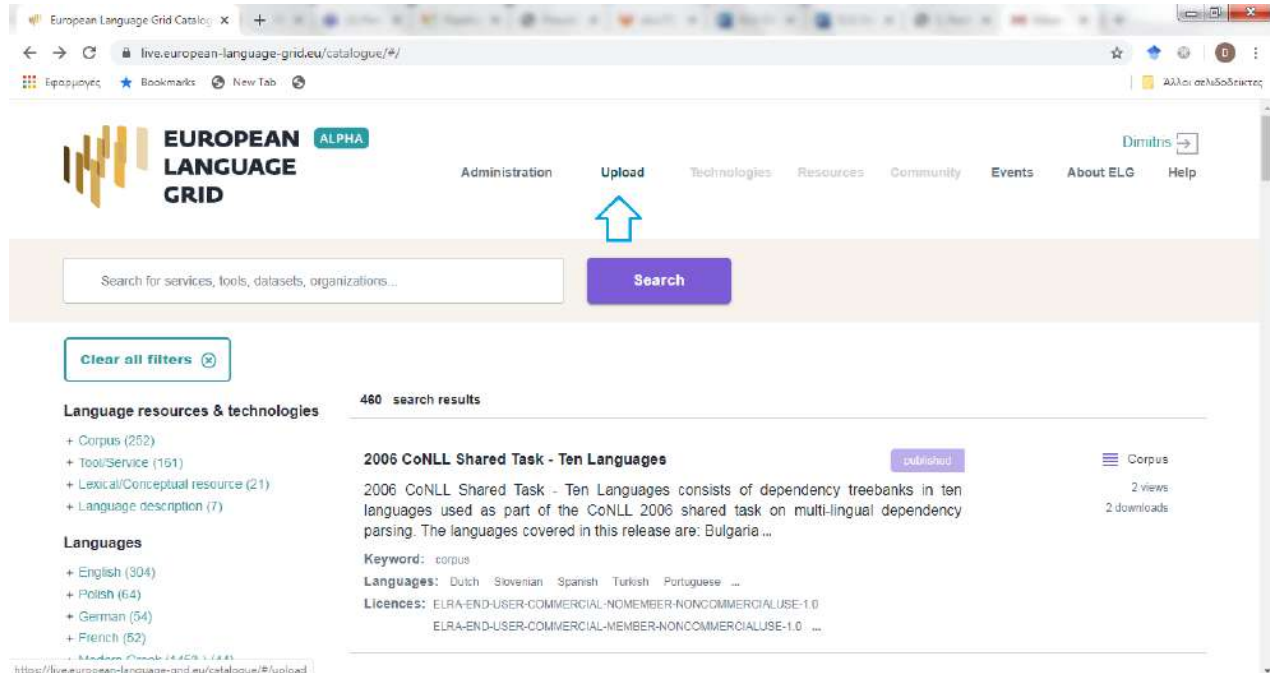
```

        <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>
        <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>

```

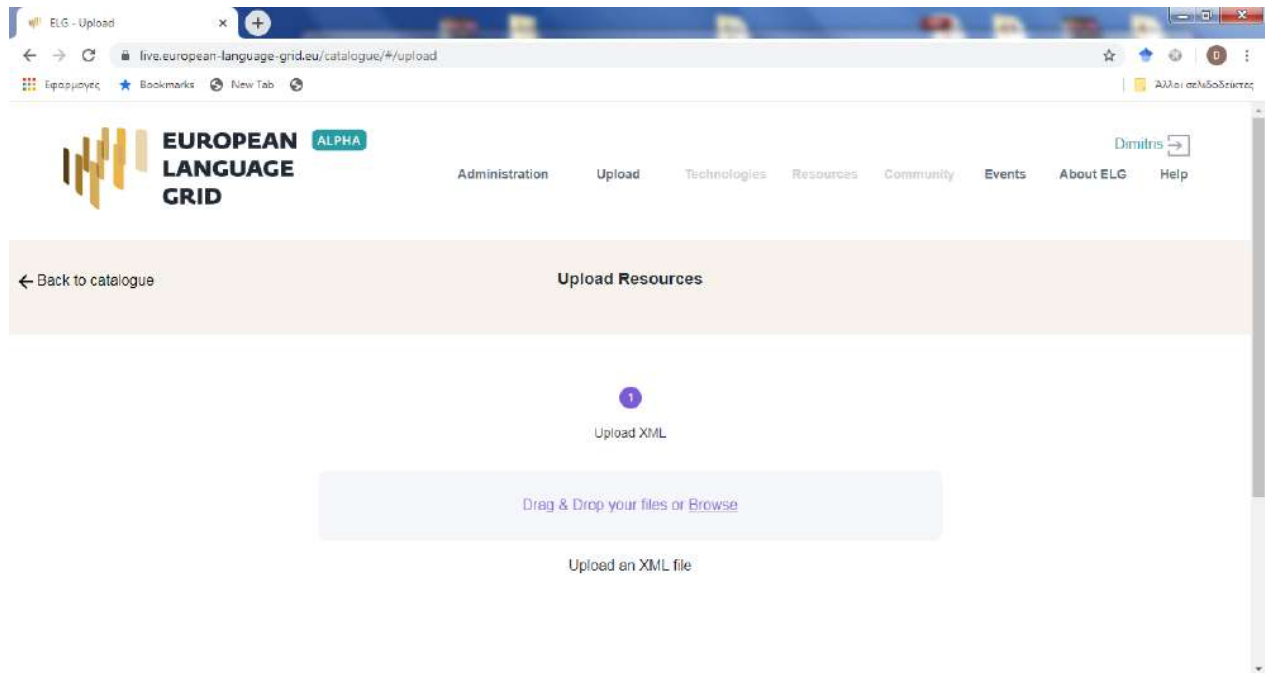
16.3 Step 2: upload

From the ELG catalogue, click the “Upload” link as shown below:



Now upload the file you created in Step 1:

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



success message will be shown to you and the metadata will be imported into the database.

16.4 Step 3: wait for approval

At this stage, the metadata record is only visible to you and to us, the ELG platform administrators. We will check your contribution and integrate it into the ELG catalogue if everything is in order, and contact you otherwise.

Contribute an organisation

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

17.1 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*.

17.2 Step 1: create metadata

The first step is to describe your organisation using ELG's metadata format, ELG-SHARE. Future releases of ELG will include an interactive editor for this. However, for now, you must create an XML file. Refer to the examples below for how to do this.

The elements you need are documented on the following pages:

- *Minimal elements for all entities*
- *Minimal elements for organisations*

For more information about ELG-SHARE, see:

- *Metadata schema*

At the ELG GitLab, you will find [templates](#) (that you can use to create new metadata records) and [examples](#) in XML format.

17.2.1 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:MetadataRecordIdentifier ms:MetadataRecordIdentifierScheme="http://w3id.org/meta-share/meta-share/elg">value automatically assigned - leave as is</ms:MetadataRecordIdentifier>
  <ms:metadataCreationDate>2020-01-07</ms:metadataCreationDate>
  <ms:metadataLastDateUpdated>2020-01-07</ms:metadataLastDateUpdated>
  <ms:metadataCurator>
    <ms:actorType>Person</ms:actorType>
    <ms:surname xml:lang="en">Smith</ms:surname>
    <ms:givenName xml:lang="en">John</ms:givenName>
    <ms:email>smith@example.com</ms:email>
  </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">Smith</ms:surname>
    <ms:givenName xml:lang="en">John</ms:givenName>
    <ms:email>smith@example.com</ms:email>
  </ms:metadataCreator>
  <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 (14 in Prague, 2 in Hradec Králové and 1 in Plzeň), 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:Organization>
</ms:DescribedEntity>
</ms:MetadataRecord>

```

(continues on next page)

(continued from previous page)

```

        <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>
                                </ms:LTArea>
                                <ms:LTArea>
                                        <ms:LTClassOther>language resources creation</ms:LTClassOther>
                                        </ms:LTArea>
                                        <ms:LTArea>
                                                <ms:LTClassOther>Dialog systems</ms:LTClassOther>
                                                </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:hasDivision>
                                                        <ms:divisionName xml:lang="en">Institute of Formal and Applied
↪Linguistics</ms:divisionName>
                                                        <ms:divisionShortName xml:lang="en">UFAL</ms:divisionShortName>
                                                        <ms:divisionCategory>http://w3id.org/meta-share/meta-share/institute</
↪ms:divisionCategory>
                                                        <ms:organizationBio xml:lang="en">'Institute of Formal and Applied
↪Linguistics (UFAL) 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
↪both for the Master's degree (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>
                                                        <ms:logo>https://ufal.mff.cuni.cz/sites/all/themes/drufal/css/logo/
↪logo_ufal_110u.png</ms:logo>

```

(continues on next page)

(continued from previous page)

```

    <ms:LTArea>
      <ms:LTCClassRecommended>http://w3id.org/meta-share/omtd-share/
↪MachineTranslation</ms:LTCClassRecommended>
    </ms:LTArea>
    <ms:LTArea>
      <ms:LTCClassRecommended>http://w3id.org/meta-share/omtd-share/
↪SpeechRecognition</ms:LTCClassRecommended>
    </ms:LTArea>
    <ms:LTArea>
      <ms:LTCClassRecommended>http://w3id.org/meta-share/omtd-share/
↪Annotation</ms:LTCClassRecommended>
    </ms:LTArea>
    <ms:LTArea>
      <ms:LTCClassRecommended>http://w3id.org/meta-share/omtd-share/
↪LexiconCreation</ms:LTCClassRecommended>
    </ms:LTArea>
    <ms:LTArea>
      <ms:LTCClassOther>Lexical Resources</ms:LTCClassOther>
    </ms:LTArea>
    <ms:LTArea>
      <ms:LTCClassOther>Dialog systems</ms:LTCClassOther>
    </ms:LTArea>
    <ms:LTArea>
      <ms:LTCClassOther>Corpus Creation</ms:LTCClassOther>
    </ms:LTArea>
    <ms:LTArea>
      <ms:LTCClassOther>Research Infrastructure</ms:LTCClassOther>
    </ms:LTArea>
    <ms:LTArea>
      <ms:LTCClassOther>LT services</ms:LTCClassOther>
    </ms:LTArea>
    <ms:LTArea>
      <ms:LTCClassOther>NLP Support</ms:LTCClassOther>
    </ms:LTArea>
    <ms:LTArea>
      <ms:LTCClassOther>Digital Humanities</ms:LTCClassOther>
    </ms:LTArea>
    <ms:keyword xml:lang="en">Computational Linguistics</ms:keyword>
    <ms:headOfficeAddress>
      <ms:address xml:lang="en">Malostranská n. 25</ms:address>
      <ms:zipCode>11800</ms:zipCode>
      <ms:city xml:lang="en">Praha 1</ms:city>
      <ms:country>CZ</ms:country>
    </ms:headOfficeAddress>
  </ms:hasDivision>
</ms:Organization>
</ms:DescribedEntity>
</ms:MetadataRecord>

```

17.2.2 Example 2: SME

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:MetadataRecordIdentifier ms:MetadataRecordIdentifierScheme="http://w3id.
↳ org/meta-share/meta-share/elg">value automatically assigned - leave as is</
↳ ms:MetadataRecordIdentifier>
    <ms:metadataCreationDate>2020-01-07</ms:metadataCreationDate>
    <ms:metadataLastDateUpdated>2020-01-07</ms:metadataLastDateUpdated>
    <ms:metadataCurator>
      <ms:actorType>Person</ms:actorType>
      <ms:surname xml:lang="en">Smith</ms:surname>
      <ms:givenName xml:lang="en">John</ms:givenName>
      <ms:email>smith@example.com</ms:email>
    </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">Smith</ms:surname>
      <ms:givenName xml:lang="en">John</ms:givenName>
      <ms:email>smith@example.com</ms:email>
    </ms:metadataCreator>
    <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 constit
↳ 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>

```

(continues on next page)

(continued from previous page)

```

<ms:logo>https://www.european-language-grid.eu/wp-content/uploads/2019/03/
↪logo__consortium-elda.svg</ms:logo>
<ms:LTArea>
  <ms:LTCClassRecommended>http://w3id.org/meta-share/omtd-share/
↪LanguageTechnology</ms:LTCClassRecommended>
</ms:LTArea>
<ms:LTArea>
  <ms:LTCClassRecommended>http://w3id.org/meta-share/omtd-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>

```

17.3 Step 2: upload

From the ELG catalogue, click the “Upload” link as shown below:

Now upload the file you created in Step 1:

If there are any errors in your XML file, these will be shown to you. Fix them and try the upload again. Eventually, a success message will be shown to you and the metadata will be imported into the database.

17.4 Step 3: wait for approval

At this stage, the metadata record is only visible to you and to us, the ELG platform administrators. We will check your contribution and integrate it into the ELG catalogue if everything is in order, and contact you otherwise.

European Language Grid Catalogue

live.european-language-grid.eu/catalogue/#/

Ευρωπαϊκή Γρίδα

Bookmarks New Tab

Αλλάει σελίδα/θέμα

EUROPEAN LANGUAGE GRID ALPHA

Administration Upload Technologies Resources Community Events About ELG Help

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

Search

Clear all filters

460 search results

Language resources & technologies

- + Corpus (252)
- + Tool/Service (151)
- + Lexical/Conceptual resource (21)
- + Language description (7)

Languages

- + English (304)
- + Polish (64)
- + German (54)
- + French (52)

2006 CoNLL Shared Task - Ten Languages

published

Corpus

2 views

2 downloads

2006 CoNLL Shared Task - Ten Languages consists of dependency treebanks in ten languages used as part of the CoNLL 2006 shared task on multi-lingual dependency parsing. The languages covered in this release are: Bulgaria ...

Keyword: corpus

Languages: Dutch Slovenian Spanish Turkish Portuguese ...

Licences: ELRA-END-USER-COMMERCIAL-NOMEMBER-NONCOMMERCIALUSE-1.0 ELRA-END-USER-COMMERCIAL-MEMBER-NONCOMMERCIALUSE-1.0 ...

https://live.european-language-grid.eu/catalogue/#/upload

ELG - Upload

live.european-language-grid.eu/catalogue/#/upload

Ευρωπαϊκή Γρίδα

Bookmarks New Tab

Αλλάει σελίδα/θέμα

EUROPEAN LANGUAGE GRID ALPHA

Administration Upload Technologies Resources Community Events About ELG Help

← Back to catalogue

Upload Resources

1

Upload XML

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

Upload an XML file

CHAPTER 18

Update / Delete a catalogue entry

Note: Instructions on how to update or delete entries in the ELG catalogue will be provided shortly.

Contribute via an external repository

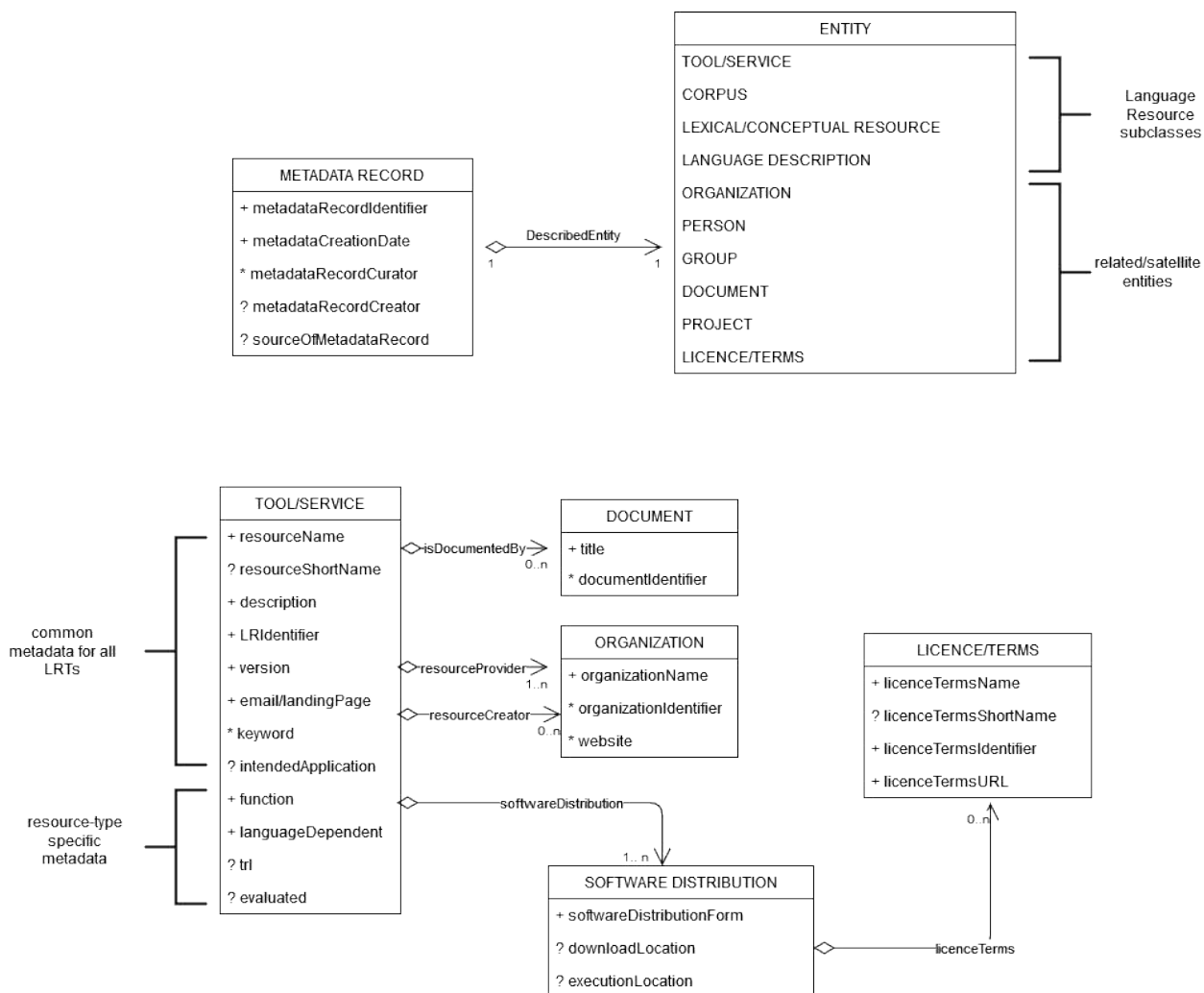
19.1 Metadata harvesting

Note: Instructions on harvesting metadata from external repositories will be provided shortly.

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.

20.1 Basic concepts

The following figure shows the main notions upon which the ELG schema builds.



These include:

- **MetadataRecord**: It corresponds to the catalogue entry, and records information concerning the registration process, such as who created the entry 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 entries*).
- **LanguageResource**, which is further classified 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, 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.

20.2 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](#)

20.3 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, prerequisites for tools and services, etc.

For each metadata element we present 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:**
 - **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
- *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.

Minimal elements for all entities

This page describes the minimal metadata elements common to all types of entities.

21.1 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 includes a set of administrative data, of which the main elements (**automatically assigned by the ELG software**) for metadata records registered by individuals are:

- metadataCreator: the person that has created the metadata record
- 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
- metadataCreationDate: the date when the metadata record was created
- compliesWith: for ELG metadata records, this is by default the ELG-SHARE metadata schema

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>
```

(continues on next page)

(continued from previous page)

```
<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>
</ms:metadataRecord>
```

Minimal elements for all language resources and technologies

This page describes the minimal metadata elements common to **all language resources and technologies**.

22.1 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>
```

22.2 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>
```

22.3 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>
```

22.4 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 resource 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>
```

22.5 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.

Example

```
<logo>https://gate.ac.uk/plugins/gau-0.1/images/logo-gate.png</logo>
```

22.6 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>
```

22.7 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>
```

22.8 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>
```

22.9 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>

```

22.10 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 organizations (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>

```

22.11 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>
```

22.12 resourceCreator

Path `MetadataRecord.DescribedEntity.LanguageResource.resourceCreator`

Data type `component`

Optionality Recommended

Explanation & Instructions

Links a resource to the person, group or organisation that has created the resource

The element is important for citation and acknowledgement purposes.

For organizations you must add the name of the organizations (`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>
```

22.13 fundingProject (RA)

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).

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:fundingProject>
```

22.14 intendedApplication

Path MetadataRecord.DescribedEntity.LanguageResource.

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:LTClassRecommended>http://w3id.org/meta-share/omtd-share/
↪NamedEntityRecognition</ms:LTClassRecommended>
</ms:intendedApplication>

<ms:intendedApplication>
  <ms:LTClassRecommended>http://w3id.org/meta-share/omtd-share/
↪SentimentAnalysis</ms:LTClassRecommended>
```

(continues on next page)

(continued from previous page)

```
</ms:intendedApplication>

<ms:intendedApplication>
  <ms:LTCClassOther>face recognition</ms:LTCClassRecommended>
</ms:intendedApplication>
```

22.15 isDocumentedBy

Path `MetadataRecord.DescribedEntity.LanguageResource.`

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>
```

Minimal elements for tools/services

This page describes the minimal metadata elements specific to **tools/services**.

23.1 ToolService

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.ToolService

Data type component

Optionality Mandatory

Explanation & Instructions

Introduces the set of elements that is specific to tools/services

Example

```
<ms:LRSubclass>
  <ms:ToolService>
    <ms:lrType>toolService</ms:lrType>
    ...
  </ms:ToolService>
</ms:LRSubclass>
```

23.2 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>
```

23.3 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`.
- `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): The URL where the docker image of the service adapter can be downloaded from. Required only for ELG functional 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'
- `licenceTerms` (Mandatory): See *licenceTerms*

23.4 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.

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:licenceTerms>

<ms:licenceTerms>
  <ms:licenceTermsName xml:lang="en">publicDomain</ms:licenceTermsName>
  <ms:licenceTermsURL>https://elrc-share.eu/terms/publicDomain.html</
↪ms:licenceTermsURL>
</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:licenceTerms>
```

23.5 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>
```

23.6 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>

```

23.7 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>

```

(continues on next page)

(continued from previous page)

```

        <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>

```

23.8 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>

```

(continues on next page)

(continued from previous page)

```
<ms:language>
  <ms:languageTag>en-US</ms:languageTag>
  <ms:languageId>en</ms:languageId>
  <ms:regionId>US</ms:regionId>
</ms:language>
```

23.9 framework

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.ToolService.framework

Data type CV (framework)

Optionality Recommended

Explanation & Instructions

Specifies the implementation framework used for developing and running a tool/service

Example

```
<ms:framework>http://w3id.org/meta-share/meta-share/TensorFlow</ms:framework>
```

23.10 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

Example

```
<ms:implementationLanguage>Java v8.1</ms:implementationLanguage>
```

23.11 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>
```

23.12 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>
```

23.13 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>
```

Minimal elements for corpora

This page describes the minimal metadata elements specific to **corpora**.

24.1 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>
```

24.2 corpusSubclass

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corpus.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>
```

24.3 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](#).

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>
  <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>

```

24.4 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 text 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 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.
- `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](#)

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>
```

24.5 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 text 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 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.
- `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.

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>
```

(continues on next page)

(continued from previous page)

```

    <ms:modalityType>http://w3id.org/meta-share/meta-share/spokenLanguage</
↪ms:modalityType>
    <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>

```

24.6 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 video 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 text 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 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.

- 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.

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>
```

24.7 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>
```

24.8 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>
```

24.9 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>
```

24.10 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>
```

24.11 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>
```

24.12 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](#)

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>
```

(continues on next page)

(continued from previous page)

```

        <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>
        <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>

```

24.13 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>
  <ms:characterEncoding>http://w3id.org/meta-share/meta-share/UTF-8</
↪ms:characterEncoding>
</ms:distributionTextFeature>

```

24.14 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**).
- **audioFormat** (Mandatory): 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>

```

(continues on next page)

(continued from previous page)

```

    </ms:durationOfAudio>
    <ms:audioFormat>
      <ms:dataFormat>http://w3id.org/meta-share/omtd-share/wav</
↪ms:dataFormat>
      <ms:compressed>true</ms:compressed>
    </ms:audioFormat>
  </ms:distributionAudioFeature>

```

24.15 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**).
- **videoFormat** (Mandatory): 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:videoFormat>
    <ms:dataFormat>http://w3id.org/meta-share/omtd-share/wav</
↪ms:dataFormat>
    <ms:compressed>true</ms:compressed>
  </ms:videoFormat>

```

24.16 distributionImageFeature

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.
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`).
- `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:imageFormat>
    <ms:dataFormat>http://w3id.org/meta-share/omtd-share/Pdf</
↪ms:dataFormat>
    <ms:compressed>true</ms:compressed>
  </ms:imageFormat>
```

24.17 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 (optional) `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>
```

24.18 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 (optional) sensitiveDataDetails to provide more information.

Example

```
<ms:sensitiveDataIncluded>true</ms:sensitiveDataIncluded>
<ms:sensitiveDataDetails>The corpus contains medical data for persons with_
↪disabilities</ms:sensitiveDataDetails>
```

24.19 anonymized

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.Corporus.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>
```

24.20 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>
```

Minimal elements for language descriptions

This page describes the minimal metadata elements specific to **language descriptions**, a type of language resource under which we subsume both models and grammars.

25.1 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>
```

25.2 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>
```

25.3 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>

```

25.4 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>

```

25.5 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>
```

Minimal elements for lexical/conceptual resources

This page describes the minimal metadata elements specific to **lexical/conceptual resources**.

26.1 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>
```

26.2 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>
```

26.3 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>
```

26.4 ContentType

Path `MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.LexicalConceptualResource.ContentType`

Data type CV (`ContentType`)

Optionality Mandatory

Explanation & Instructions

A more detailed account of the linguistic information contained in the lexical/conceptual resource

You can repeat the element for multiple encoding levels.

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>
```

26.5 compliesWith

Path MetadataRecord.DescribedEntity.LanguageResource.LRSubclass.LexicalConceptualResource.ContentType

Data type CV (compliesWith)

Optionality Mandatory

Explanation & Instructions

Specifies the vocabulary/standard/best practice to which a resource is compliant with

You can repeat the element for multiple encoding levels.

Example

`<ms:compliesWith>http://w3id.org/meta-share/meta-share/LMF</ms:compliesWith>`

26.6 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` (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.
- `metalanguage` (Mandatory): Specifies the language that is used in the resource part, 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>
```

Minimal elements for projects

This page describes the minimal metadata elements specific to **projects**.

27.1 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>
```

27.2 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>
```

27.3 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>
```

27.4 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>
```

27.5 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>
```

27.6 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>
```

27.7 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>
```

27.8 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>
```

27.9 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>
```

27.10 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

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>
```

27.11 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>
```

27.12 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>
```

27.13 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>
```

27.14 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

Note: TODO: add example

27.15 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 βC" 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
↳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.
```

27.15. projectSummary

Based on these, 30\ national competence centres and the European LT Board will be
 set up for European coordination. The ELG will foster β language technologies for
 Europe built in Europeβ, tailored to our languages and cultures and to our
 societal and economical demands, benefitting the European citizen, society,

(continued from previous page)

--

Minimal elements for organisations

This page describes the minimal metadata elements specific to **organisations**.

28.1 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>
```

28.2 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.

Example

```
<ms:OrganizationIdentifier ms:OrganizationIdentifierScheme="http://w3id.org/meta-
↪share/meta-share/elg">automatically assigned by ELG - please don't change</
↪ms:OrganizationIdentifier>
```

28.3 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>
```

28.4 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>
```

28.5 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>
```

28.6 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 (14 in Prague, 2 in Hradec Králové and 1 in Plzeň), 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 catalogues of
available data sets, embracing all aspects of Language Technologies. ELDA was also
involved in evaluation initiatives, in several FPs' projects involving HLT (continues on next page)
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
```

28.5. organizationAlternativeName

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

28.7 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

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>
```

28.8 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>
```

28.9 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>
```

28.10 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>
```

28.11 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">Computational Linguistics</ms:keyword>
<ms:keyword xml:lang="en">Natural Language Processing</ms:keyword>
<ms:keyword xml:lang="en">Language Resources</ms:keyword>
```

(continues on next page)

(continued from previous page)

```
<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>
```

28.12 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@elda.org</ms:email>
```

28.13 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>
```

28.14 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>
```

28.15 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/release1.1.0/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>
```

28.16 hasDivision

Path MetadataRecord.DescribedEntity.Organization.hasDivision

Data type component

Optionality Recommended

Explanation & Instructions

Links an organization to the division(s) it consists of

Example

```

<ms:hasDivision>
  <ms:divisionName xml:lang="en">Institute of Formal and Applied Linguistics</
↪ms:divisionName>
  <ms:divisionShortName xml:lang="en">UFAL</ms:divisionShortName>
  <ms:divisionCategory>http://w3id.org/meta-share/meta-share/institute</
↪ms:divisionCategory>
  <ms:organizationBio xml:lang="en">'Institute of Formal and Applied
↪Linguistics (IFAL) 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
↪both for the Master's degree (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>
  <ms:logo>https://ufal.mff.cuni.cz/sites/all/themes/drufal/css/logo/logo_ufal_
↪110u.png</ms:logo>
<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>
</ms:LTArea>
<ms:LTArea>
  <ms:LTClassOther>Lexical Resources</ms:LTClassOther>
</ms:LTArea>
<ms:LTArea>
  <ms:LTClassOther>Dialog systems</ms:LTClassOther>
</ms:LTArea>
<ms:LTArea>
  <ms:LTClassOther>Corpus Creation</ms:LTClassOther>
</ms:LTArea>
<ms:LTArea>
  <ms:LTClassOther>Research Infrastructure</ms:LTClassOther>
</ms:LTArea>
<ms:LTArea>
  <ms:LTClassOther>LT services</ms:LTClassOther>
</ms:LTArea>
<ms:LTArea>
  <ms:LTClassOther>NLP Support</ms:LTClassOther>
</ms:LTArea>

```

(continues on next page)

(continued from previous page)

```
<ms:LTArea>
  <ms:LTCClassOther>Digital Humanities</ms:LTCClassOther>
</ms:LTArea>
<ms:keyword xml:lang="en">Computational Linguistics</ms:keyword>
  <ms:addressSet>
    <ms:address xml:lang="en">Malostransk  n m. 25</ms:address>
    <ms:zipCode>11800</ms:zipCode>
    <ms:city xml:lang="en">Praha 1</ms:city>
    <ms:country>CZ</ms:country>
  </ms:addressSet>
</ms:hasDivision>
```


CHAPTER 29

API specifications

This annex contains the specifications of ELG's internal and public application programming interfaces (APIs).

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*
 - *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.

30.1 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`.

30.2 Utility datatypes

The following JSON structures are used in several places in this specification, they are documented here to avoid duplication.

30.2.1 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.

30.2.2 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.

30.3 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*.

30.3.1 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.

30.3.2 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": [
    {
      "texts": [
        { "content": "The" }, { "content": "European" }, { "content": "Language" }, { "content":
↪ "Grid" }
```

(continues on next page)

(continued from previous page)

```

    ]
  },
  {
    "texts": [
      { "content": "An" }, { "content": "API" }, { "content": "example" }
    ]
  }
]
}

```

30.3.3 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.

30.4 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.

30.4.1 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.

30.4.2 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.

30.4.3 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 */
  }
}
```

(continues on next page)

(continued from previous page)

```

"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.

30.4.4 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.

30.4.5 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 */

```

(continues on next page)

(continued from previous page)

```

"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.

30.4.6 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.

30.5 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`.

30.6 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}
```

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 programatically. The token is passed via the HTTP `Authorization` header in the usual way:
`Authorization: Bearer <tokenValue>`

31.1 Input formats

31.1.1 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`

31.1.2 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.

31.1.3 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.

31.2 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.

31.3 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.

CHAPTER 32

Terms of use

Use of the European Language Grid is subject to its [terms of use](#).

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.

33.1 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 Lagzdīns, 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 Gavrilidou, 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 Lagzdīns. [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 Hajic, 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]

33.2 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 Odijk, 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](#). Dordrecht, 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 Prys 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 Hajič, 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 Odijk, 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 Hajič, Inma Hernáez, John Judge, Svetla Koeva, Simon Krek, Cvetana Krstev, Krister Lindén, Bernardo Magnini, Joseph Mariani, John McNaught, Maite Melero, Monica Monachini, Asunción Moreno, Jan Odijk, Maciej Ogrodniczuk, Piotr Pęzik, Stelios Piperidis, Adam Przepiórkowski, Eiríkur Rögnvaldsson, Michael Rosner, Bolette Pedersen, Inguna Skadiņa, 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 Odijk, 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.

33.3 Deliverables

Note: These publications will be provided shortly.

CHAPTER 34

Indices and tables

- `genindex`
- `modindex`
- `search`

C

Code sample, [21](#)

D

Download, [21](#)

I

Internal LT API, [175](#)

P

Public API, [188](#)

T

Try Out, [21](#)