

# Epilexo: uno strumento per la creazione, modifica e interlinking di lessici antichi

## Rapporto Tecnico v.1.0

<b>Riassunto:</b>  Il documento descrive le caratteristiche tecniche e lo sviluppo dell'interfaccia web della piattaforma EpiLexo, con annesse informazioni relative al data-model e alle tecnologie utilizzate per la realizzazione della piattaforma.
<b>Autori:</b> Michele Mallia, Istituto di Linguistica Computazionale "A. Zampolli" - CNR, Pisa  <b>Supervisione:</b> Valeria Quochi, Istituto di Linguistica Computazionale "A. Zampolli" - CNR, Pisa

### Storia del documento

Versione	Autori	Data	Commenti
0.1	Michele Mallia	05/02/2021	Prima bozza
0.2	Michele Mallia	14/03/2022	Architettura interfaccia
0.3	Michele Mallia	03/05/2022	Editor di lessici e dei testi
0.4	Valeria Quochi	29/12/2022	Revisione sezioni 1 e 2;
0.5	Valeria Quochi	4-5/1/2023, 10/1	Revisione sezione 3;
0.6	Michele Mallia	24-25/01/2023	Correzione sezioni 1 e 2
0.7	Valeria Quochi	06/03/2023	Revisione complessiva, suggerimenti per miglioramenti: presentare come lavoro presente, sintetizzare parti introduttive non funzionali al documento presente
0.8	Michele Mallia	07/03/2023	Modifiche parti introduttive, aggiunta descrizione funzionalità tecniche sezioni 2 - 3

0.9	Valeria Quochi & Michele Mallia	10/03/2023	Riorganizzazione delle e tagli alle parti introduttive
1.0	Michele Mallia	10/04/2025	Cambio reference bibliografica LexO

# INDICE

## Contents

<b>1. Introduzione</b>	<b>7</b>
1.1 Scopo	7
<b>2. Quadro generale: il progetto ItAnt*</b>	<b>7</b>
2.1 Obiettivi del progetto	7
2.2 Architettura funzionale del sistema EpiLexO	8
2.2.1 Schema di base dello stack applicativo	8
<b>3. Background e stato dell'arte</b>	<b>9</b>
3.1. Epigrafia Digitale	9
3.1 Papyri.info	10
3.1.1 Papyrological Navigator (PN)	11
3.1.2 Papyrological Editor (PE)	12
3.2 iSicity	13
<b>4 L'applicazione web EpiLexo</b>	<b>14</b>
4.1 Interfaccia di editing	14
4.2 Requisiti dell'interfaccia di editing	15
4.3 Layout dell'interfaccia	16
4.3.1 Text-Tree, Lexical Entries Tree, Metadata Panel e Language Manager	17
4.3.2 Linker	19
4.3.3 Editing delle risorse lessicali	21
4.3.4 Editing risorse esterne	22
4.4 Gestione Utenti	23
4.5 Servizi del front-end	24
4.5.1 Servizi del lessico (LexO-Server)	24
4.5.1.1 Servizi per la creazione del lessico	24
Create Lexical Entry	25
Create Form	25
Create Sense	26
Create Etymology	26
Create EtyLink	26
Create Bibliography	27
Create Language	27
4.5.1.2 Servizi per il recupero dei dati lessicali	28
Get Languages	28
Get Lexical Entries	28
Get Lexical Entry	29
Get Elements	29
Get Filtered Forms	30
Get Forms	31

Get Form	31
Get Filtered Senses	31
Get Senses	32
Get Sense	33
Get Etimologies	33
Get Etimology	33
Get Linguistic Relation	34
Get Generic Relation	34
4.5.1.3 Servizi per la modifica dei dati lessicali	34
Update Lexical Entry	35
Update Form	35
Update Sense	36
Update Language	36
Update Etymology	37
Update EtyLink	37
Update Linguistic Relation	38
Update Generic Relation	38
Synchronize Bibliography	39
4.5.1.4 Servizi per l'eliminazione dei dati lessicali	39
Delete Lexical Entry	40
Delete Form	40
Delete Sense	40
Delete Etymology	41
Delete Etymological Link	41
Delete Language	41
Delete Bibliography	42
4.5.1.5 Servizi per il recupero di dati statistici e di vocabolario	42
Get Lexical Entry Types	42
Get Namespaces	43
Get Authors	43
Get Languages	43
Get Pos	44
Get Status	44
Get Morphology	45
Get Form Type	45
4.5.1.6 Altri servizi	45
Import Lexicon	45
Export Lexicon	46
Query Federation	47
4.5.2 Servizi dei testi (CASH)	47
4.5.2.1 Servizi per la gestione delle attestazioni	47
Create annotation	47
Update annotation	48

Get Unstructured Data	48
Create Token	49
Get Tokens	49
Get Text	50
Get Content	50
Get Annotations	50
Delete Annotation by value	51
Delete Annotation	51
4.5.2.2 Servizi per la ricerca all'interno della collezione di testi	51
Test Search	52
Search Files	52
4.5.2.3 Servizi per la gestione dei testi	52
Download File	52
Upload File	53
Update File Metadata	53
Rename Folder, Rename File	54
Move Folder, Move File	54
Delete Metadata	55
Create File	55
Copy File	55
Add Folder	56
4.5.2.4 Servizi accessori per il file-system e lo status dei servizi	56
Ping	56
Get Document System	57
Get Users	57
4.5.3 Servizi di autenticazione (Keycloak)	57
Search Users	58
Get User Roles	58
Get User Info	58
Get Users By Role	59
Create User	59
Delete User	59
Update User	60
Get Client Roles	60
Get Client Info	60
Set Roles to User	61
Delete Roles to User	61
4.5.4 Modelli dei dati	62
4.5.4.1 LexO Data Model	62
Lexical Entry	62
Form	63
Sense	64
EtymologySet, EtyLink ed Etymology	65

BibliographyItem	66
Language	67
LexicalFilter	68
LexicalEntryElements	69
Morphology	69
Link	70
Property	70
MorphologyValue	71
PropertyValue	71
LinguisticRelation, Generic Relation	72
UpdateResponse	72
StatisticElement	72
LiLaResponse	73
4.5.4.2 CASH Data Model	73
Annotation	73
UnstructuredBody	74
TextResponse	74
AnnotationResponse	74
TokenResponse	75
Token	75
AnnotationAttributes	76
AnnotationBibliography	76
BibliographyTags	78
BibliographyCreator	78
AnnotationSpan	78
TestSearchResponse	79
MetadataModel	79
SearchFileRequest	79
SearchFileResponse	80
File	80
DownloadRequest, RemoveRequest, DeleteMetadataRequest, AddFolderRequest	81
CashResponse	81
UploadResponse, CreateFileResponse, AddFolderResponse	81
Node	81
UpdateMetadataRequest	82
RenameFolderRequest	82
MoveFileRequest, MoveFolderRequest	83
CreateFileRequest	83
GetDocumentSystemResponse	83
GetUsersResponse	84
User	84
4.5.4.3 Keycloak Data Model	84
UserModel	85

RolesModel	85
CreateUserRequestParam	86
Credential	86
UpdateUserRequest	86
4.5 Problemi aperti e sviluppi futuri	87
<b>5. Bibliografia</b>	<b>88</b>

# 1. Introduzione

## 1.1 Scopo

Il presente documento descrive lo sviluppo di “EpiLexo”, una piattaforma web per la creazione, l’aggiornamento e la revisione di lessici elettronici collegati a materiali testuali e bibliografici, nonché a risorse lessicografiche esterne. Il rapporto si concentra sulla progettazione e lo sviluppo del front-end di una piattaforma web, a partire da una revisione delle tecnologie utilizzate nel campo dell’epigrafia digitale e della rappresentazione della conoscenza linguistica e lessicale. L’obiettivo è quello di creare uno strumento che possa migliorare la modifica dei lessici e il collegamento tra questi ultimi e le risorse epigrafiche. Il rapporto definisce l’architettura funzionale generale della soluzione che viene implementata all’interno del progetto.

Il sistema di creazione, di modifica e di ricerca delle risorse linguistiche, sviluppato in Angular, opera su dati formalizzati secondo standard dell’epigrafia digitale – in questo caso di EpiDoc; della lessicografia computazionale e dei Linguistic Linked Open Data, al fine di creare nuove risorse digitali per le lingue dell’Italia antica, che siano allo stesso tempo fruibili facilmente dagli studiosi, e accessibili a tutti, e *machine-understandable/ machine actionable*.

Attualmente la piattaforma prevede, oltre ai servizi già menzionati, un sistema per la gestione delle utenze e delle autorizzazioni per modificare i materiali caricati sulla piattaforma. Nel presente documento vengono discusse le varie fasi della progettazione, le specifiche tecniche di ciascun componente e il modello generale dell’organizzazione dei dati.

## 2. Quadro generale: il progetto ItAnt\*

L’applicazione web EpiLexo è dedicata in particolare alla gestione di materiali epigrafici risalenti all’Italia antica, ed è sviluppata nel contesto del progetto ItAnt ovvero “Lingue e culture dell’Italia antica: linguistica storica e modelli digitali” (\*finanziato dal MUR - PRIN 2017 - 2017XJLE8J) che ha tra i suoi obiettivi primari quello di realizzare una infrastruttura informatica modulare per la creazione, gestione e fruizione di materiali linguistici per lingue e arcaiche in un ambiente collaborativo multi-utente a supporto degli epigrafisti e degli studiosi della linguistica storica.

### 2.1 Obiettivi del progetto

L’obiettivo principale del progetto ItAnt è quello di offrire ai linguisti storici e studiosi di culture antiche uno strumento elettronico per la descrizione e/o lo studio del lessico di lingue arcaiche anche di frammentaria attestazione a partire da e in collegamento con i testi che ne costituiscono le attestazioni /i testimoni e la bibliografia della letteratura secondaria di riferimento. L’idea di base è quella di creare uno strumento che consenta agli studiosi di realizzare dei lessici elettronici

di lingue antiche e arcaiche basati e collegati ai relativi testimoni al fine di facilitare la condivisione e la collaborazione (tra studenti, professori e centri di ricerca) anche a scopi didattici e di divulgazione. Da un punto di vista prettamente tecnico al momento non sembrano esistere soluzioni software web disponibili che permettano una facile creazione di risorse lessicali integrate con testi (annotati) e riferimenti bibliografici, né tantomeno che permettano export e pubblicazione dei dati in formati compatibili con il paradigma della scienza aperta e del web semantico.

Un obiettivo tecnico di questa piattaforma consiste dunque nella possibilità di realizzare in modalità user-friendly lessici compatibili e pubblicabili come Linguistic Linked Open Data, ovvero basati su ontologie, modelli e vocabolari condivisi nella comunità del web semantico di modo che questi stessi dati possano essere riutilizzati all'interno della comunità scientifica, magari con scopi diversi

## 2.2 Architettura funzionale del sistema EpiLexO

In questa sezione si illustra l'architettura generale del sistema EpiLexO di cui in questo documento si discute l'interfaccia. Si parte dallo spiegare i vari livelli che costituiscono il progetto e dell'architettura realizzata in Docker per la comunicazione tra i vari "container" che rappresentano i vari servizi back-end che comunicheranno con la parte di front-end. Di quest'ultima, vengono illustrate anche le varie interfacce e viste, come ad esempio quella della fruizione delle risorse e delle altre relative all'editing dei lessici per passare poi alla gestione degli utenti.

### 2.2.1 Schema di base dello stack applicativo

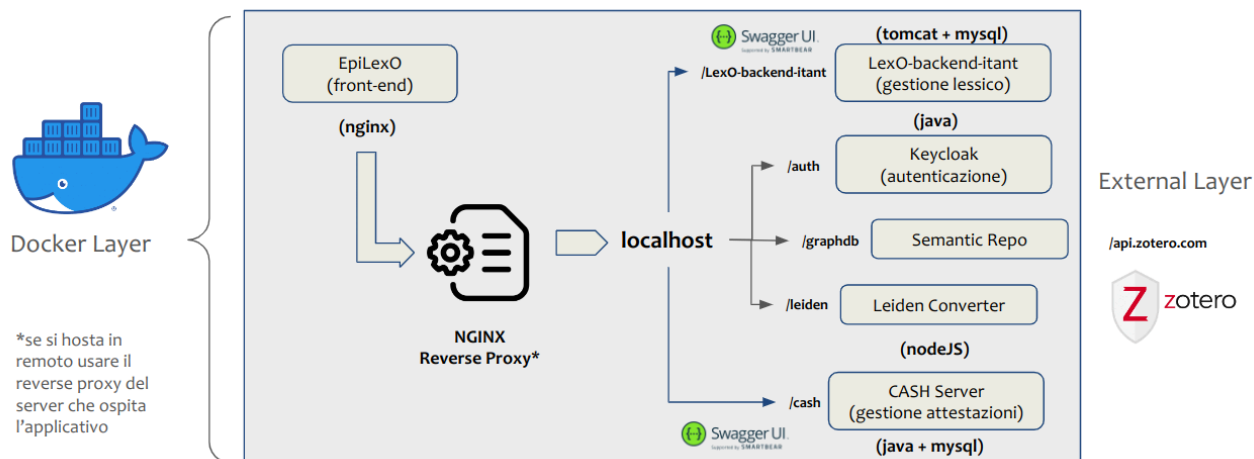
L'utilizzo di tecnologie come Docker<sup>1</sup>, che consentono di isolare e "containerizzare" applicazioni facendo in modo di tenere ben distinte le varie strutture di una piattaforma, permettono di avere una rappresentazione più chiara e ordinata dell'architettura di un progetto.

Le macchine virtuali che operano all'interno di questo contesto sono le seguenti:

- nginx: si occuperà di ospitare la build del progetto in Angular
- tomcat: contiene l'applicativo web per la gestione dei servizi del lessico, da cui dipende un'altra macchina virtuale che contiene un database mySql per la memorizzazione delle operazioni
- open-jdk: queste due macchine contengono i servizi relativi all'autenticazione (keycloak) e la gestione dei materiali epigrafici (belexo);

---

<sup>1</sup> <https://www.docker.com/>



Architettura generale di EpiLexO

Questa piccola rete di comunicazione permette di realizzare un “cluster” applicativo che consente di poter gestire separatamente i vari aspetti dell’applicazione e di poter avere quindi un maggior controllo sui vari servizi; è possibile infatti mantenere una persistenza dei dati delle applicazioni e di poter effettuare deploy in maniera più semplice, adottando anche policy di riavvio delle macchine in caso di problemi relativi a eventuali bug o casi che potrebbero appesantire la virtual machine coinvolta.

### 3. Background e stato dell’arte

#### 3.1. Epigrafia Digitale

Il compito dell’epigrafia digitale è quello di codificare le informazioni presenti nei materiali epigrafici attraverso l’uso di meta-dati scritti tramite linguaggi formali che possono essere elaborati da un calcolatore. Purtroppo, la maggior parte dei progetti presenti appaiono molto vari e diseguali, sia per quanto riguarda il materiale preso in considerazione sia per il modo di strutturare l’informazione, utilizzando standard e tecniche di codifica diverse.

A partire dai primi anni del 2000 si iniziano a intravedere i primi progetti legati alla conservazione e alla consultazione di iscrizioni antiche; tra i maggiori portali di epigrafia digitale tenuti sotto osservazione per lo sviluppo di questo progetto e che costituiscono punti di riferimento importanti di questa disciplina ritroviamo il Database Epigrafico di Heidelberg, i database epigrafici di Roma e

Bari e il portale del progetto Papyri<sup>2</sup>. In questi progetti è possibile consultare / esplorare corpus di risorse attraverso una vasta gamma di filtri e metodi di ricerca che consentono anche ad un utente non esperto di poter ricercare la risorsa desiderata, anche attraverso l'ausilio di mappe ed altri elementi interattivi. Oltre alla consultazione, in progetti come Papyri, è possibile anche effettuare delle modifiche attraverso un'interfaccia grafica che permette l'editing della risorsa, accessibile soltanto da chi dispone delle autorizzazioni per poter modificare le informazioni dell'iscrizione.

Oltre ad avvalersi in maniera sempre più consistente delle tecnologie digitali per poter affrontare lo studio delle iscrizioni in maniera innovativa, potendo condividere meglio i risultati all'interno della comunità accademica e al di fuori di essa, il mondo dell'epigrafia digitale ha visto anche nascere un'integrazione tra queste tecnologie e quelle relative alla gestione del sapere e della conoscenza, orientandosi in un paradigma del web affermatosi nell'ultimo decennio come uno standard: quello dei *linked open data* (LOD) e del *web-semantic*<sup>3</sup>.

Di seguito vengono illustrati due dei progetti più salienti di epigrafia digitale che stabiliscono lo stato dell'arte della disciplina e/o costituiscono un riferimento nel settore dell'epigrafia digitale, e che hanno ispirato lo sviluppo della piattaforma EpiLexo, descrivendone le caratteristiche peculiari.

### 3.1 Papyri.info

Papyri.info è una piattaforma lanciata nel 2010 contenente una vasta raccolta di testi digitali di papiri documentari greci e latini. I documenti raccolti riguardano conti, petizioni, ricevute fiscali e lettere databili dal IV secolo a.C all'VIII secolo d.C, per lo più scritti su materiali papiracei e scoperti prevalentemente negli scavi archeologici in Egitto[5].

Il progetto è composto da due elementi principali: il Papyrological Navigator (PN), ovvero uno strumento per cercare e sfogliare i documenti e il Papyrological Editor (PE), che consente agli utenti autorizzati di contribuire alla raccolta dei testi e alla loro edizione elettronica. Il contenuto, ovvero il testo dei papiri e i relativi metadati, è ottenuto tramite l'aggregazione di database preesistenti e anche al lavoro collaborativo degli iscritti alla piattaforma, prevalentemente rappresentato da linguisti e storici delle culture antiche.

---

<sup>2</sup> I siti elencati sono consultabili ai seguenti indirizzi:

- <http://www.uni-heidelberg.de/institute/sonst/adw/edh/index.html>
- <http://www.edr-edr.it/index.php>
- <http://www.edb.uniba.it/>
- <https://papyri.info/>

<sup>3</sup> I linked data sono un metodo per codificare/rappresentare e pubblicare i dati strutturati usando vocabolari come schema.org che possono essere collegati tra loro e interpretati dalle macchine. I Linked Data si basano su tecnologie standard del web come il protocollo HTTP e gli URI, ma anziché usarle per permettere ai lettori di accedere alle pagine web, le estende in modo tale da condividere le informazioni anche con i computer che possono così leggere automaticamente le informazioni ivi contenute. In questo modo, diventa possibile collegare i dati provenienti da fonti diverse e ricavarne ulteriori informazioni.

Questa piattaforma costituisce un punto di riferimento per la papirologia documentaria, in quanto aggrega tre principali banche di dati di documenti papirologici:

- **Dike Databank of Documentary Papyri (DDbDP)**, è una raccolta molto ampia di testi di papiri documentari realizzati dalla Duke University di Heidelberg;
- **Heidelberger Gesamtverzeichnis der griechischen Papyruskunden Ägyptens (HGV)**, fornisce i metadati di tutti i papiri documentari;
- **Advanced Papyrological Information System (APIS)**, creata dall'Università del Michigan, è una banca dati che si concentra sulla immagini e sui metadati di papiri provenienti dalle istituzioni nordamericane e da alcune raccolte federate in tutto il mondo

Di grande importanza è anche l'integrazione di uno strumento essenziale per la ricerca papirologica, ovvero il principale database bibliografico, la **Bibliographie Papyrologique (BP)**.

Questa raccolta di dati, ottenuti tramite la codifica dei papiri, offre due vantaggi principali. In primo luogo, fornisce una maggiore accessibilità per coloro che sono interessati allo studio della papirologia. In secondo luogo, offre l'opportunità di contribuire in modo significativo a questa disciplina, poiché è possibile migliorare la qualità delle risorse (come testi, bibliografie e citazioni) presenti nella piattaforma, aggiungendo o modificando informazioni che non sono mai state inserite prima.

### 3.1.1 Papyrological Navigator (PN)

Il Papyrological Navigator è formato dall'aggregazione di queste tre banche dati (DDbDP, HGV, APIS) all'interno delle quali possiamo trovare testi, metadati, immagini, riferimenti bibliografici e risorse attraverso l'integrazione del database *Trismegistos*<sup>4</sup>. Questo progetto è stato creato con l'obiettivo di fornire in modo facile e gratuito a informazioni sui testi egizi antichi, inclusi i testi funerari, le iscrizioni monumentali sui testi, le loro provenienze geografiche e temporali e sulle persone coinvolte nella loro creazione e conservazione.

---

<sup>4</sup> <https://www.trismegistos.org/>

**Papyri.info** sign in

Browse: DDBDP HGV APIS DCLP Authors TM Number or Search: **Data** Bibliography

Refine Search New Search Search

within  chars

and or not then near lex clear

regex abbr start-not end-not

Convert from betacode as you type  
 ignore capitalization  
 ignore diacritics/accents

Text  Metadata  Translations

Series or Collection Author Work

DDBDP: c.pap.gr (49) Vol. ID # unknown (49) unknown (49)

DDBDP SERIES: c.pap.gr X  
 COLLECTION: ddbdp X  
 VOLUME: 1 X

49 hits. Records per page 15 Go

Identifier	Title	Location	Date	Languages	Translations	Images
<a href="#">c.pap.gr 1 1</a>	keiner	Tebtynis (Arsinoites)	232 BCE	grc, grc	None	Img (ext.), Print
<a href="#">c.pap.gr 1 2</a>	Aufhebung ...	Alexandria	15 BCE	grc, grc	None	Img (ext.), Print
<a href="#">c.pap.gr 1 3</a>	Rückzahlung ...	Alexandria Alexandria	14 BCE	grc, grc	None	Img (ext.), Print

Papyrological Navigator - <https://papyri.info/search>

Oltre a questi dati, il PN include il più importante database di informazioni bibliografiche per i papiri, il *Bibliographie Papyrologique* (BP), all'interno del quale sono presenti ricche corrispondenze tra papiri e bibliografie. Infine, il PN include una *checklist* delle edizioni, ovvero un elenco di edizioni papirologiche e strumenti utilizzati come riferimento per le abbreviazioni standard.

### 3.1.2 Papyrological Editor (PE)

Il secondo componente, l'Editor papirologico (PE), è stato sviluppato per facilitare l'inserimento e la modifica di materiali epigrafici caricati nel database. All'interno dell'editor, oltre alla struttura TEI EpiDoc, è possibile effettuare una modifica dei metadati che rappresentano il materiale, apportando, ove possibile, eventuali correzioni. La piattaforma invita attivamente gli utenti a collaborare con gli studiosi al fine di creare un corpus di informazioni sui papiri quanto più esteso e corretto possibile, al fine, anche, di raffinare le interrogazioni al database.

home | MicheleMalia | sign out | help

Browse: DDbDP HGV APIS DCLP Authors TM Number or Search: Data Bibliography | Advanced Create

Overview Text Meta APIS (beta)

Edit History Comment

Leiden+ XML Commentary Preview

Editing C.Pap.Gr. 1 1 from publication [C.Pap.Gr. 1 1](#)  
Identifier [papyri.info/ddbpc.pap.gr;1;1](#) ([View in PN](#))

Markup Symbols Leiden+ Help UnderDot Try It Find Error

Edit summary (Briefly describe the changes you have made):

Leiden+ [Copy to Clipboard](#) [Save](#)

<S=>g<D=.r<=>

1. ((Ἔτους)) <#ις=16#> (Φαμενῶ(θ)) <#κ=20#>. πέπτωκεν εἰς κιβωτὸν τὸ συνάλλαγμα
2. ἐν Τεβτύνει τοῦ Ἀρσινοίτου νομοῦ δι' Ἀντικράτους τοῦ
3. παρὰ Ἀρμοδίου τροφοῦ εἰς ἔτη τρία <#=3#>, ((δραχμῶν)) <#σν=250#>. <:ἑποίησεν|regίεπωσεν:>
4. Σποννήσις Ὀρου Φανήσει Νεχθύριος.

=>=D><D=.v<=>

1. (Lang: Demotic 1 lines) =>=D>

Vista del Papyrological Editor - [https://papyri.info/editor/publications/92980/ddb\\_identifiers/212204/edit](https://papyri.info/editor/publications/92980/ddb_identifiers/212204/edit)

Per facilitare il lavoro anche ai non addetti ai lavori, gli sviluppatori di Papyri.info hanno implementato un'interfaccia (nominata *Leiden+*) che assiste gli utenti nell'interpretare i segni e nello scrivere correttamente una sintassi nel formato di *Leida* rappresentante un materiale epigrafico. Le modifiche apportate ai materiali deve passare attraverso la revisione di un utente autorizzato ad approvare le modifiche effettuate a un testo e, soltanto dopo la sua approvazione e *chiusura*, il testo viene pubblicato e reso visibile agli altri utenti. Degno di nota è anche lo spazio per le edizioni "born digital", soggette a revisione editoriale, in cui le consuete discussioni introduttive, commenti e immagini possono essere integrate con relativa facilità nelle registrazioni testuali[6].

### 3.2 iSicily

Il progetto I.Sicily[7] mira a creare un corpus completo delle iscrizioni in tutte le lingue parlate nella Sicilia antica, datate dal VII secolo a.C. fino al V secolo d.C. Al momento, la copertura primaria sono i testi incisi su pietra. Il progetto è stato motivato dalla frammentarietà, dal difficile accesso e dalla mancata digitalizzazione dei documenti cartacei. La piattaforma si concentra sulla gestione dei testi, la creazione di un elenco di riferimenti basato su citazioni bibliografiche e una quantità limitata di metadati. I dati sono stati convertiti in formato XML appartenente allo standard EpiDoc e implementati con tecnologie come eXist e SOLR per la gestione dei dati. La conversione dei dati in EpiDoc XML ha fornito l'opportunità di arricchire il dataset esistente in conformità ai Linked Open Data, implementando, ad esempio, dei riferimenti ad entità relative a istituzioni o luoghi utilizzando riferimenti geografici e con l'incorporazione di Pleiades. Inoltre, sono state inserite

informazioni relative alla bibliografia, ai musei e il testo epigrafico stesso, utilizzando un processo automatizzato.

## 4 L'applicazione web EpiLexo

In questa sezione si espongono le caratteristiche tecniche relative all'applicazione *EpiLexO*; vengono chiariti gli obiettivi dell'applicazione stessa, i requisiti funzionali e l'architettura funzionale di essa, al fine di offrire una panoramica quanto più completa al lettore di questa relazione tecnica.

La realizzazione di EpiLexO prende le mosse da un'esperienza precedente (Bellandi 2021) di applicazione full-stack per la creazione di lessici multilingue secondo il modello Ontolex-lemon<sup>5</sup>.

Poiché l'utilizzo di editor di ontologie come *Protégé*<sup>6</sup> può rivelarsi abbastanza complicato, la realizzazione di un sistema ad interfaccia grafica più intuitiva e interattiva per l'editing di risorse lessicali per il web semantico, può facilitare l'attività anche per quegli utenti che non hanno competenze tecniche per gestire le varie complessità del modello Ontolex *lemon*.

Per questo progetto è stato utilizzato Angular 11 per la realizzazione della *single page application* in sé e, per quanto riguarda la parte grafica, è stato utilizzato Bootstrap<sup>7</sup>, una delle librerie più famose per la realizzazione di interfacce grafiche per applicazioni web. Il fatto di usare un framework come Angular è stato dettato dal fatto di voler realizzare un'applicazione che tenesse conto degli standard moderni per la "costruzione" di uno strumento accessibile a tutti particolarmente versatile e "fluid" rispetto alle tipologie classiche di applicativi web, sia in termini di sviluppo che in termini di usabilità.

### 4.1 Interfaccia di editing

Una delle parti fondamentali del progetto è quella che riguarda la gestione e la modifica dei lessici e dei testi epigrafici caricati sulla piattaforma. Si tratta di un'interfaccia particolarmente complessa, in quanto deve tenere conto di diversi aspetti, uno tra i quali è il seguente: è necessario che si debba poter lavorare, contemporaneamente, sia sui testi e sia sul corpus di lessici. La motivazione principale è che l'utente deve poter avere sotto mano tutti gli elementi per poter effettuare le operazioni che gli serviranno per costruire l'apposito dizionario di lessici o gestire la collezione di testi e le relative annotazioni.

---

<sup>5</sup> Nel campo della lessicografia computazionale OntoLex-Lemon è lo standard de-facto per la pubblicazione di risorse lessicali nel web semantico. Il modello Ontolex-lemon un modello e un vocabolario per rappresentare informazioni linguistiche collegate alle ontologie e ad altri vocabolari comuni.

<sup>6</sup> <https://protege.stanford.edu/>

<sup>7</sup> <https://getbootstrap.com/>

## 4.2 Requisiti dell'interfaccia di editing

Qui vengono elencati i requisiti dell'interfaccia di editing, che si suddividono nelle seguenti sezioni:

- Fetching e modifica dei dati per le alberature: la piattaforma dovrà recuperare i dati relativi ai testi XML caricati su CASH e sul lessico caricato/creato nel repository semantico di GraphDB; sinteticamente, vengono qui riportate le operazioni consentite dalla piattaforma:
  - Visualizzare in una struttura *tree-like* i testi e i lessici
  - Operazioni sui testi come in un *file-system* (crea/elimina/rinomina cartelle, sottocartelle, files, files vuoti)
  - Operazioni sul lessico (aggiungere lexical entries al nodo *root* principale, forme, sensi ed etimologie come sotto-nodi delle lexical entries)
  
- Sistema per le attestazioni: è possibile, una volta caricato un documento TEI-XML su CASH, effettuare le seguenti operazioni:
  - Visualizzare il testo sotto-forma di *tokens*
  - Visualizzare il testo codificato nella *convenzione di Leida*
  - Visualizzare le traduzioni del testo
  - Creare attestazioni sui tokens legandoli a forme presenti nel lessico (GraphDB)
    - è possibile includere bibliografie e informazioni su altri riferimenti per ciascuna forma
  - Visualizzare dei metadati presenti nel file EpiDoc
  - Filtrare i risultati per la ricerca di un testo (ricerca full-text, per identificativo, data, etc...)
  - Scaricare i files XML caricati su CASH
  
- Sistema per la gestione del lessico: comprende una vasta gamma di funzionalità, tra le quali:
  - Creazione di Lexical Entries, forme, sensi, etimologie e costituenti per le *multi-word*
  - Modifica di informazioni come label, se è un etymon, il tipo di L.E., il grado di certezza, la lingua, gli aspetti morfologici, stem-type
  - Riferimenti ad altre risorse (cognates, denotes, sameAs, seeAlso) attraverso ricerche interne sul lessico e la possibilità di assegnare risorse esterne (tramite ricerca su SPARQL endpoint esterni o applicare URI esterni senza passare da filtri di ricerca)
  - Aggiunta di riferimenti bibliografici grazie alle API fornite da Zotero

## 4.3 Layout dell'interfaccia

Come si può notare nella figura 1, il layout dell'interfaccia si sviluppa in tre colonne, al cui interno ci sono determinati componenti che determinano alcune aree di lavoro. Si può infatti dividere l'interfaccia nelle seguenti aree:

- area (a) - Alberature: in questa parte sono presenti i due componenti relativi alle alberature, da una parte, dei materiali epigrafici codificati in xml e caricati nell'interfaccia con un approccio *file-system* like e, dall'altra, delle risorse lessicali create da questi ultimi
- area (b) - Area Editing: la parte centrale è dedicata alla modifica degli elementi lessicali e alla gestione dei testi, da dove è possibile effettuare le operazioni di annotazione; questa è la parte a cui è stato riservato più spazio per poter consentire una migliore fruizione degli strumenti messi a disposizione per i lessicografi
- area (c) - Riferimenti esterni: in questa parte più piccola, ma non meno importante, sono presenti tutti i vari pannelli contenenti tutte le informazioni e i riferimenti esterni che rappresentano l'entità lessicale o il materiale epigrafico selezionato;

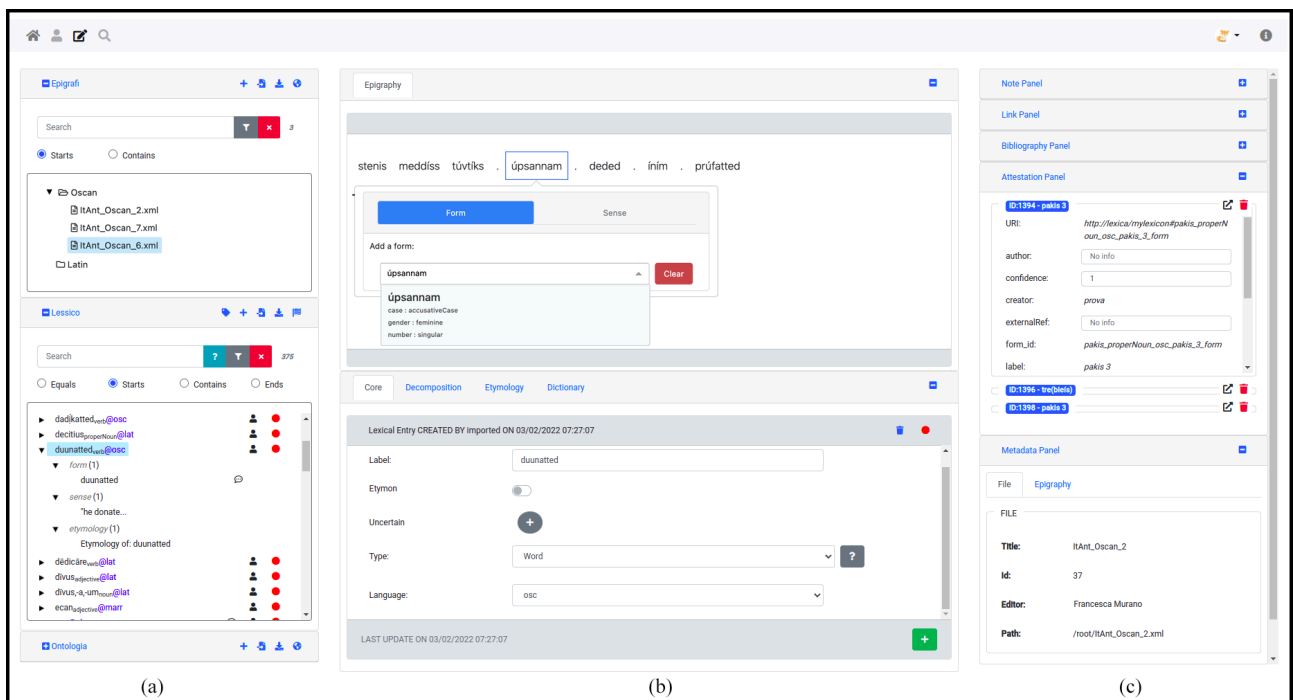


Fig. 1 : Interfaccia di editing

Ogni componente dell'interfaccia è istanziato in maniera autonoma, nel senso che ha delle proprie caratteristiche e proprietà ed è teoricamente non collegato agli altri componenti, ma esistono dei servizi che consentono lo scambio di informazioni tra le varie parti della vista in modo da poter condividere eventuali modifiche apportate dagli utenti ed eventuali aggiornamenti lato back-end.

Si passeranno adesso in rassegna le varie componenti dell'interfaccia, partendo dai componenti relativi alle alberature per passare alla parte centrale e concludendo fino alla parte a destra dell'interfaccia.

#### 4.3.1 Text-Tree, Lexical Entries Tree, Metadata Panel e Language Manager

Nella parte sinistra dell'interfaccia sono presenti due componenti fondamentali, ovvero gli alberi relativi alle entrate lessicali e quello relativo ai materiali epigrafici. Questi componenti rappresentano la parte iniziale dell'interfaccia poiché tramite questi ultimi è possibile iniziare a effettuare operazioni sul lessico e sui testi.

I due componenti fanno riferimento a due servizi differenti separati, uno dedicato alla gestione dei testi e alla loro permanenza, l'altro dedicato alla gestione delle triple e di "ontologie" che tendono a rappresentare il corpus di entrate lessicali, andando a ricalcare i vari collegamenti "semantici" e logici degli elementi lessicali e le loro risorse esterne.

Nella fase di avvio dell'applicazione, questi componenti si occupano di effettuare un bootstrap dei dati a partire dalle rispettive fonti di dati, popolando gli alberi con i propri dati dando così la possibilità all'utente di iniziare a effettuare le operazioni sul lessico o sui testi.

L'utente infatti può interagire con questi componenti cliccando sui vari componenti attivando così dei meccanismi per "attivare" le altre componenti dell'interfaccia. Se, infatti, il lessicografo cliccherà su un'entrata lessicale si attiverà il componente di editing dell'entrata lessicale stessa, previo recupero delle informazioni sull'entrata/testo selezionato; meccanismo simile anche per l'altro componente, con la differenza che si attiverà il componente per la gestione delle attestazioni.

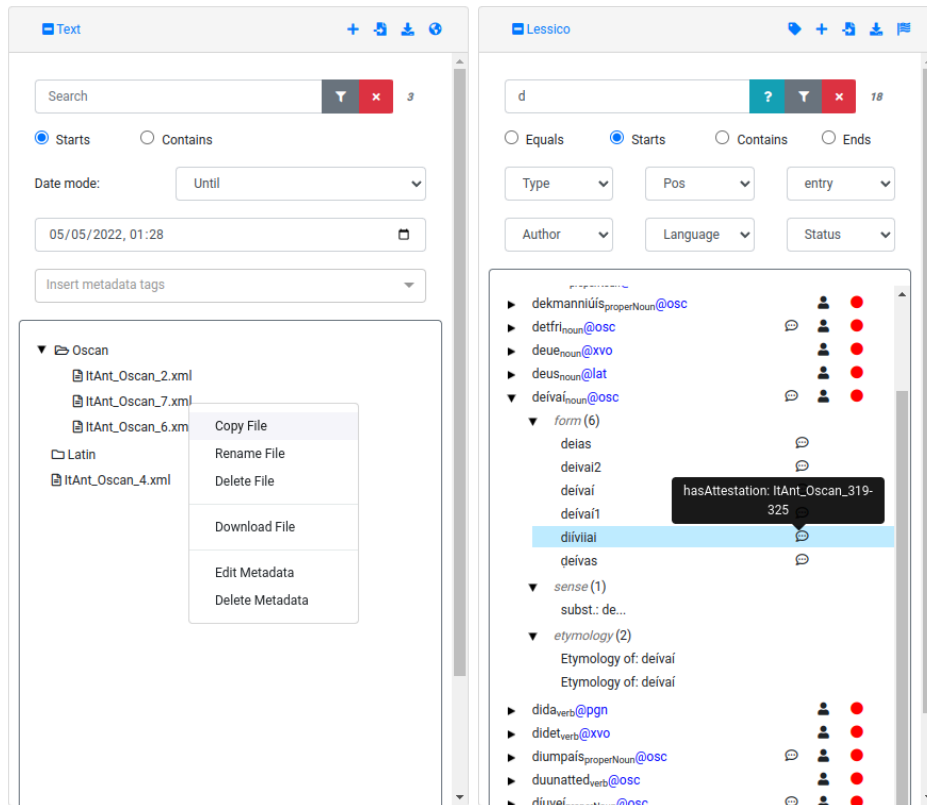


Fig. 2: A sinistra, l'albero contenente i materiali epigrafici, a destra invece l'albero contenente le entrate lessicali

All'interno di entrambi i componenti sono presenti dei filtri per la ricerca e degli elementi grafici per segnalare la presenza di eventuali commenti, stati delle entrate lessicali con eventuali autori / reviewers, tipologie di files o cartelle. Nel caso del Text-Tree, sarà possibile visualizzare  $n$  livelli in base all'organizzazione del file-system dei materiali epigrafici, nel Lexical Entries Tree invece si possono visualizzare soltanto 3 livelli, uno relativo alle entrate lessicali e i restanti due saranno suddivisi in base alla tipologia di elementi lessicali legati ad stesse (forme, sensi, etimologie, constituents e subterms<sup>8</sup>). A partire da questi componenti, inoltre, sarà possibile effettuare operazioni CRUD sugli elementi attraverso un menù contestuale che apparirà cliccando col tasto destro su uno degli items dell'albero.

Negli header dei componenti sono inoltre presenti due ulteriori componenti: il *Metadata Panel*<sup>9</sup> e il *Language Manager*<sup>10</sup>. Il primo (fig. 3) si occupa di gestire e rappresentare i metadati<sup>11</sup> di ciascun file, mentre il secondo (fig. 4) si occupa di gestire le lingue presenti nel corpus lessicale, al fine di poter aggiungere o modificare le informazioni della lingua memorizzate nel database ontologico.

<sup>8</sup> Per gli elementi *constituents* e *subterms* si deve fare riferimento alle lexical entries di tipo *multiword*.

<sup>9</sup> Accessibile attraverso tasto destro su file e cliccando su *Edit Metadata*

<sup>10</sup> Accessibile attraverso l'icona a forma di bandiera nell'header del Lexical Entries Tree

<sup>11</sup> Nella gestione dei testi, è previsto un meccanismo per la conservazione e la modifica di metadati che servono a descrivere l'oggetto in questione, testo o cartella che sia. I metadati possono essere utilizzati anche a fini di ricerca e filtraggio delle informazioni richieste dall'utente (dal Text Tree infatti è possibile fare una ricerca per "metadati" attraverso l'apposito uso di tag)

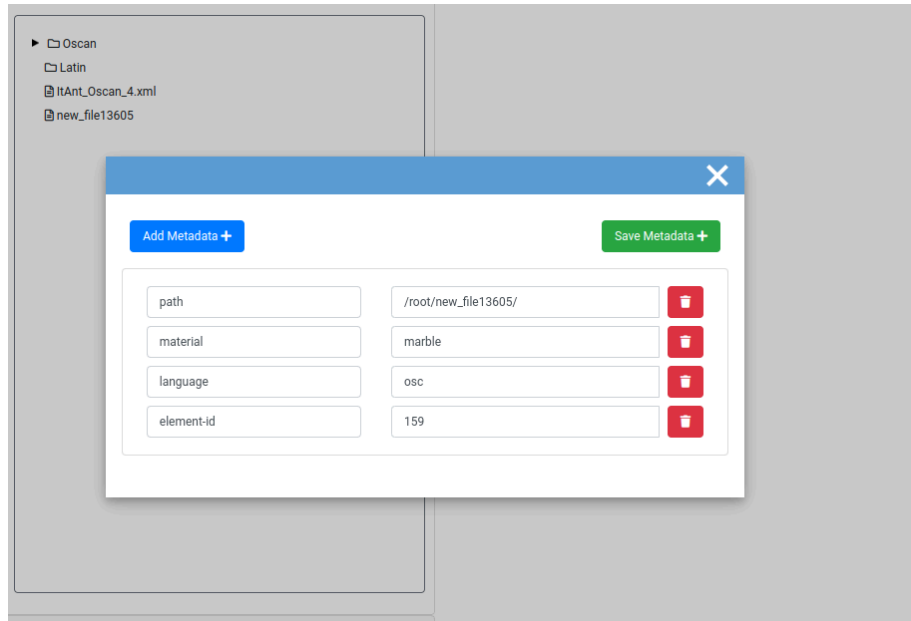


Fig. 3: Metadata Panel

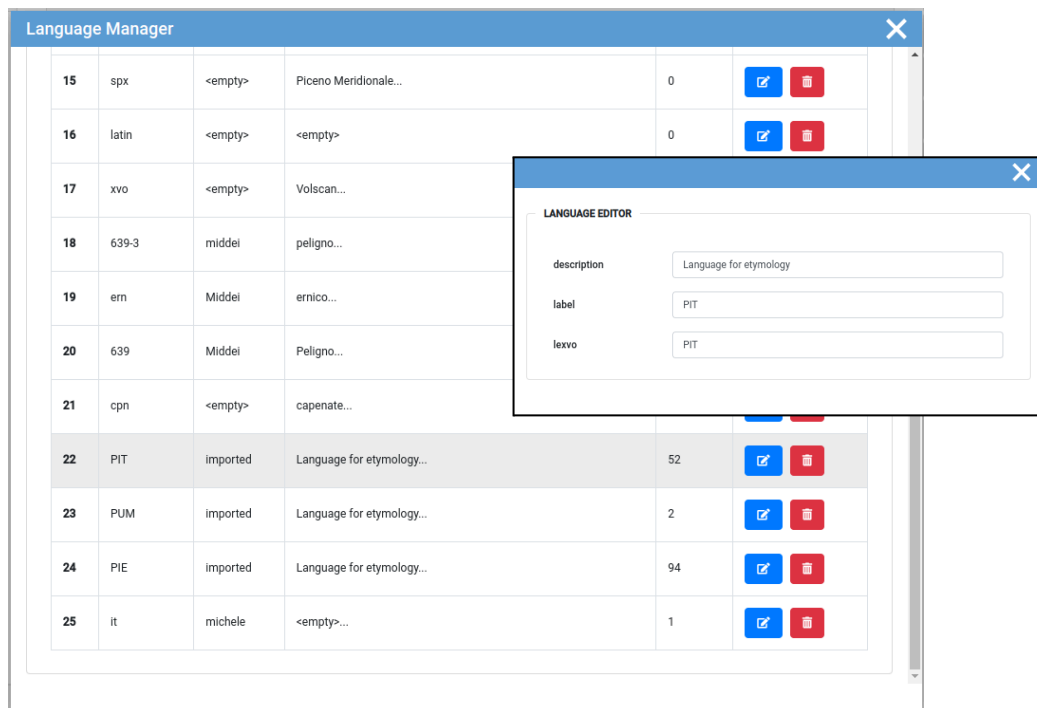


Fig. 4: Language manager

#### 4.3.2 Linker

Il primo componente della parte centrale è quello relativo alla gestione dei testi e delle varie parti del testo che lo compongono. L'utilità principale di questo componente, come si evince dal nome stesso, consiste nel collegare elementi lessicali a parti del testo presenti nel materiale epigrafico. Oltre a questo, all'interno di esso sarà possibile visualizzare tutte le parti del testo divise in token e sarà possibile inoltre vedere una rappresentazione delle trascrizioni del testo codificato secondo le convenzioni di Leiden.

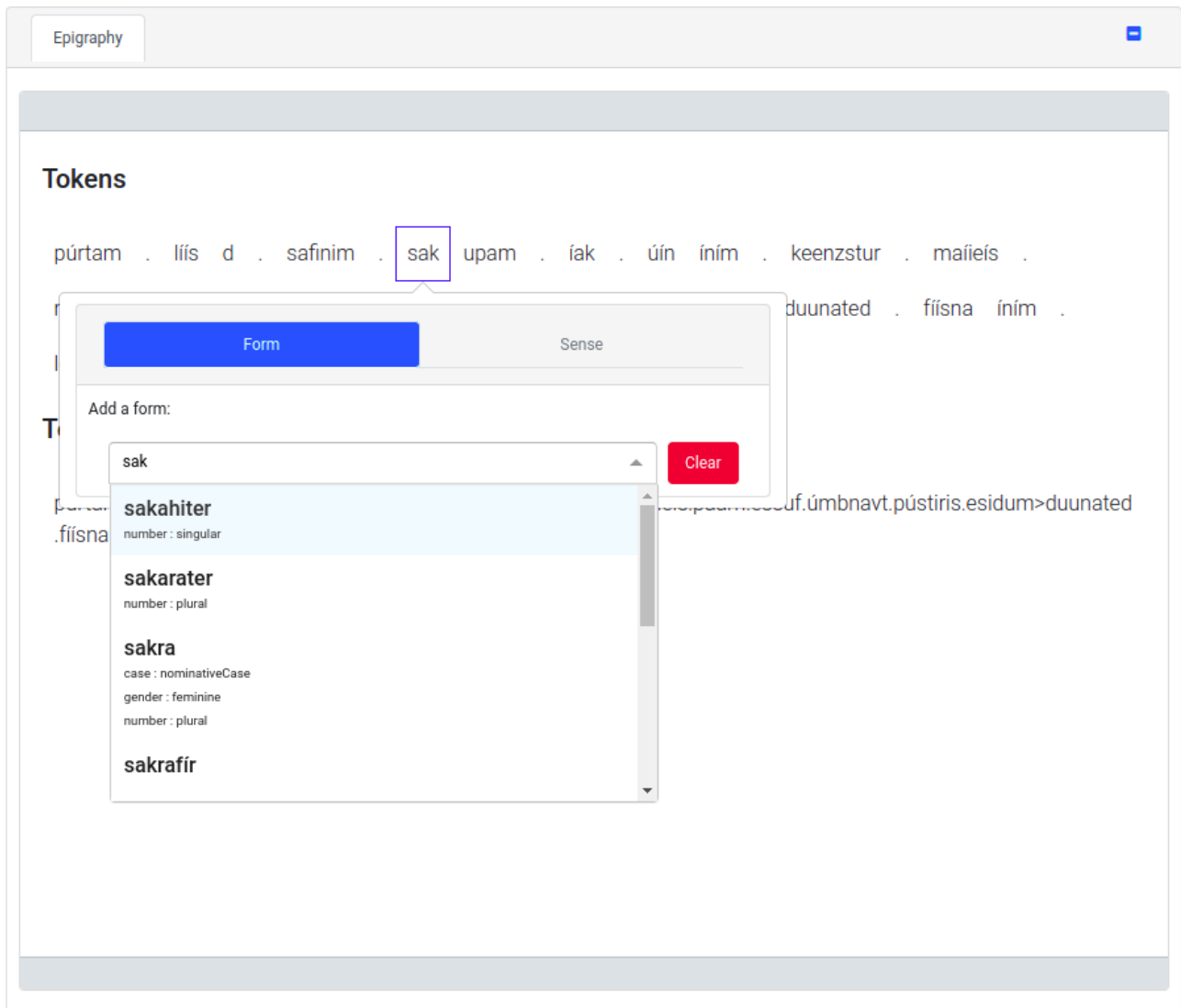


Fig. 5: vista del componente Linker nel momento in cui si sta facendo un collegamento tra una forma e un token del testo

Il testo diviso in token può essere selezionato per intero oppure può essere selezionata una piccola parte; in seguito, sarà possibile anche selezionare più token al fine di creare una *multiword*, ovvero una parola composta da più token. Per effettuare l'attestazione basta semplicemente cliccare col mouse su un token presente nell'interfaccia, attivando così il *popover* contenente il componente per la ricerca e la selezione di forme. Il processo di collegamento tra il testo e il lessico tiene in considerazione alcuni aspetti, tra cui lo *span*<sup>12</sup> di selezione dei caratteri, la posizione e l'identificativo del token selezionato. Tutte le attestazioni fatte dagli utenti saranno presenti all'interno dell'*Attestation Panel*.

<sup>12</sup> Per *span* di caratteri si intende l'intervallo corrispondente alla sottostringa o alla parola intera selezionata. Se si considera la parola *castello* e si decide di selezionare soltanto 'tel' lo *span*, che inizia con 0 e finisce con  $n-1$ , sarà di 3-6, ovvero quell'intervallo che rappresenta la sottostringa selezionata.

### 4.3.3 Editing delle risorse lessicali

Uno dei componenti principali di questa piattaforma è quello relativo alla modifica delle risorse lessicali presenti nel corpus. Tramite questo strumento è possibile modificare le informazioni relative all'elemento lessicale selezionato e aggiungere informazioni relative a forme, sensi, etimologie ed elementi bibliografici.

Lexical Entry CREATED BY imported ON 03/02/2022 07:27:07

Label: alltrei

Etymon

Uncertain

Type: Word

Language: osc

**MORPHOLOGY**

Part of speech: adjective

Stem type:

**DENOTES**

**COGNATES**

[http://lexica/mylexicon#alter\\_adjective Jat\\_entry](http://lexica/mylexicon#alter_adjective Jat_entry)

- Add Form
- Add Sense
- Add Etymology
- Add Bibliography

LAST UPDATE ON 03/02/2022 07:27:07

Fig. 6: editor di un'entrata lessicale

Esistono diversi moduli in base all'elemento lessicale selezionato: esistono infatti moduli dedicati alla modifica o alla preparazione di *entrate lessicali*, *forme*, *sensi* ed *etimologie*, in quanto questi diversi tipi di elementi possiedono informazioni da modificare diverse tra loro. Oltre a questo, si è prevista anche una suddivisione in base ai moduli presenti in *OntoLex-Lemon*; sono presenti infatti il modulo *Core*, che contiene tutti aspetti principali di un'entrata lessicale, forma o senso, e *Decomposition*, un modulo che riguarda gli aspetti delle *multiword*.

I moduli presenti nel componente sono collegati direttamente ai servizi back-end, per cui basta effettuare una singola modifica in un campo per richiamare i servizi e salvare le informazioni

aggiornate. Per la gestione delle informazioni lato front-end si è deciso di usare la tecnologia interna ad Angular dei *ReactiveForm*, strumenti in grado di gestire in maniera semplice e dinamica tutti gli aspetti relativi alla “modulistica” di una *single page application*, tramite i quali è possibile tenere traccia di tutte le modifiche apportate sul momento dall’utente.

#### 4.3.4 Editing risorse esterne

In questa parte dell’editor dei testi/lessici sono presenti vari componenti nella parte destra dell’interfaccia: sono presenti infatti cinque componenti diversi, ognuno con una propria funzione.

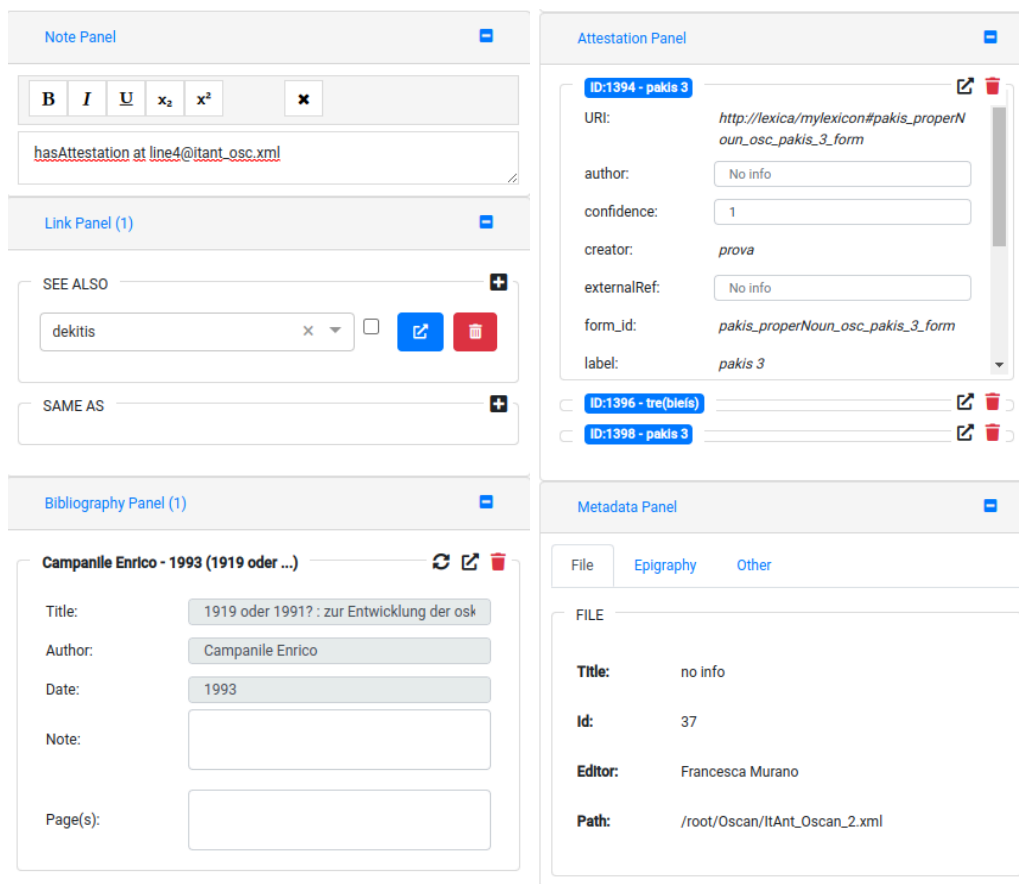


Fig.7: vista nel dettaglio dei vari componenti di editing delle risorse esterne

I componenti presenti in questa sezione sono i seguenti:

- **Note panel:** componente per la memorizzazione di note testuali contenenti informazioni aggiuntive da parte dell’autore
- **Link Panel:** componente per la gestione delle proprietà *seeAlso* e *sameAs* di ontolex-lemon per il riferimento ad altre entrate lessicali
- **Bibliography Panel:** componente per la gestione delle risorse bibliografiche legate a un’entità del lessico (forma, lexical entry, senso, etimologia...) e le proprie informazioni

- **Attestation Panel:** in questo componente sono presenti tutte le attestazioni che riguardano i token di un testo selezionato; ogni attestazione rappresenta una legatura tra un token (o una porzione di esso) a una forma presente nel dizionario dei lessici (o, eventualmente, esterna ad esso); è inoltre possibile modificare alcune informazioni di un'attestazione come ad esempio le note, riferimenti esterni e la bibliografia che documenta questo collegamento
- **Metadata Panel:** contiene tutti i metadati relativi a un testo selezionato; all'interno di esso saranno presenti tre sezioni in base alla tipologia di metadati, ovvero:
  - informazioni strutturali a livello di *file-system*
  - informazioni principali relative al documento epigrafico stesso (Trismegistos, Other ID, Luogo di origine, materiale e tipo di oggetto)
  - altre informazioni accessorie legate al materiale epigrafico

I componenti presenti in quest'interfaccia sono completamente autonomi tra di loro e comunicano con il resto dei componenti esterni a questa parte di interfaccia, come l'editor dei lessici e l'editor dei testi.

#### 4.4 Gestione Utenti

All'interno della piattaforma è possibile anche gestire le varie utenze iscritte attraverso il sistema di autenticazione di Keycloak<sup>13</sup>, un servizio open-source per l'autenticazione e la memorizzazione di informazioni relative agli accessi. Il servizio, come tutte le altre componentistiche, è montato su una macchina virtuale con Docker.

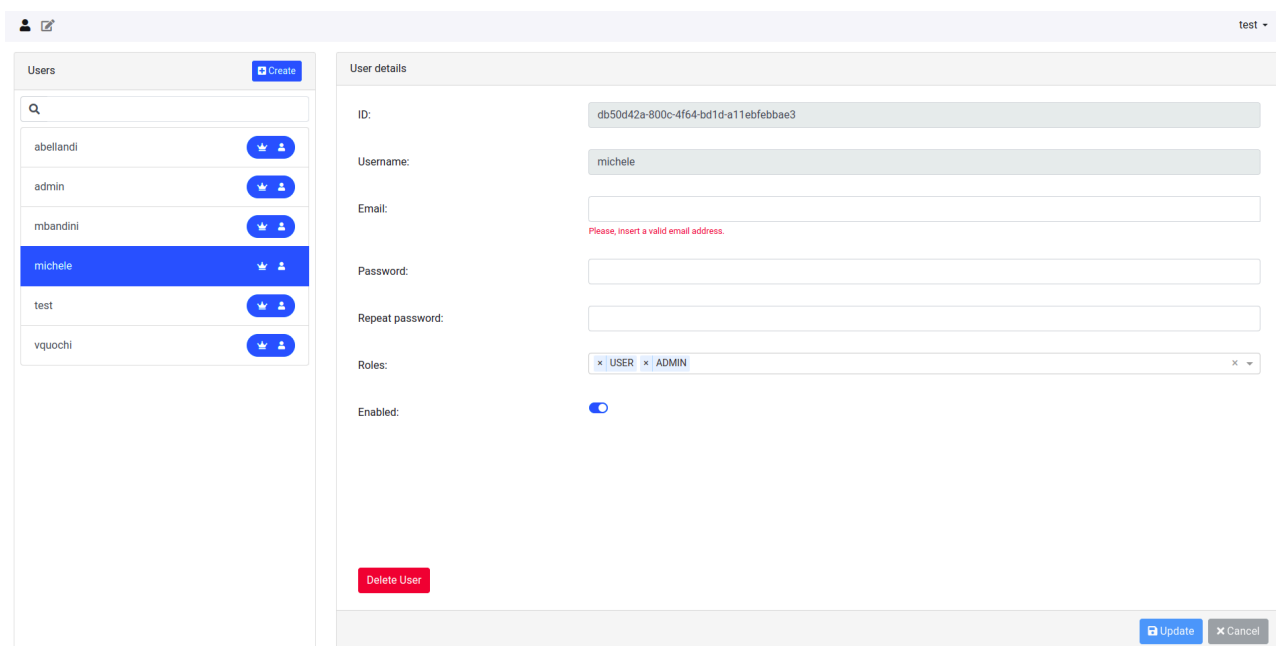
La parte relativa all'interfaccia consiste semplicemente nell'elencare tutte le utenze presenti nel *realm*<sup>14</sup> e di poter effettuare le seguenti azioni: creare un utente, modificare dati anagrafici e di profilo, eliminare un utenza.

In questo modo è possibile gestire le utenze in maniera semplice senza dover passare necessariamente dal pannello di controllo di Keycloak, potendo così consentire ai meno esperti di poter gestire questo aspetto direttamente da EpiLexO.

---

<sup>13</sup> <https://www.keycloak.org/>

<sup>14</sup> Lett: "reame", un insieme di regole e utenze con regole specifiche e a sé stanti rispetto ad altri realm presenti nel sistema di keycloak.



*Interfaccia di gestione utente*

## 4.5 Servizi del front-end

In questa sezione vengono elencati i servizi utilizzati dalla piattaforma per performare le operazioni di editing sui testi / lessici e per la gestione delle utenze. Il paragrafo è diviso in ulteriori sotto-sezioni relative agli ambiti di competenza dei servizi, i quali sono elencati in forma tabellare con annessi dati e modelli di chiamata.

### 4.5.1 Servizi del lessico (LexO-Server)

I servizi relativi al lessico sono erogati dal servizio LexO-Server, il quale comunica con GraphDB per la gestione del lessico computazionale sotto-forma di collezione di triple nel formato RDF. Esistono diversi servizi, divisi in base al tipo di operazione da effettuare su un determinato tipo di dato, che si possono riassumere in:

- Servizi per la creazione del lessico
- Servizi per il recupero di informazioni sul lessico
- Servizi per l'eliminazione di tratti morfologici / caratteristiche relative al lessico
- Servizi per l'aggiornamento di informazioni lessicali
- Servizi per la gestione delle informazioni SKOS

#### 4.5.1.1 Servizi per la creazione del lessico

Qui vengono riportati i servizi per la creazione di elementi lessicali come *entrate lessicali*, *forme*, *sensi* e le restanti componenti appartenenti al contesto OntoLex-Lemon. I servizi sono divisi in API contenente il path per eseguire le funzioni, il payload (o parametri) da inserire come queryParams<sup>15</sup>, gli eventuali BODY per le chiamate di tipo POST e le RESPONSE da parte del server. Nella parte finale di questa sezione verranno elencate anche le interfacce che rappresentano i tipi di dati restituiti dal back-end.

## Create Lexical Entry

Questa chiamata serve per creare una lexical entry nuova.

- API: GET /lexicon/creation/lexicalEntry
- PAYLOAD:
  - key : string
  - author: string
  - prefix : string
  - baseIRI : string
- RESPONSE: [LexicalEntry](#)

## Create Form

Creazione di una nuova forma a partire da una lexical entry.

- API: GET /lexicon/creation/form
- PAYLOAD:
  - lexicalEntryID : string
  - key : string
  - author: string
  - prefix : string
  - baseIRI : string
- RESPONSE: [Form](#)

---

<sup>15</sup> <https://guides.emberjs.com/release/routing/query-params/>

## Create Sense

Creazione di un nuovo senso a partire da una lexical entry.

- API: GET /lexicon/creation/lexicalSense
- PAYLOAD:
  - lexicalEntryID : string
  - key : string
  - author: string
  - prefix : string
  - baseIRI : string
- RESPONSE: [Sense](#)

## Create Etymology

- API: GET /lexicon/creation/etymology
- PAYLOAD:
  - lexicalEntryID : string
  - key : string
  - author: string
  - prefix : string
  - baseIRI : string
- RESPONSE: [EtymologySet](#)

## Create EtymLink

Creazione di un nuovo etymLink a partire da una lexical entry e da un'etimologia di riferimento.

- API: /lexicon/creation/etymologicalLink
- PAYLOAD:
  - lexicalEntryID : string
  - etymologyID : string
  - key : string

- author: string
- prefix : string
- baseIRI : string

- RESPONSE: [EtyLink](#)

## Create Bibliography

Creazione di un nuovo elemento bibliografico a partire da una lexical entry, forma, senso o etimologia.

- API: POST /lexicon/creation/bibliography

- PAYLOAD:
  - id : string
  - key : string
  - author: string
  - prefix : string
  - baseIRI : string

- BODY:
  - id : string
  - title : string
  - author : string
  - date : string
  - url : string
  - seeAlsoLink : string

- RESPONSE: [BibliographyItem](#)

## Create Language

Creazione di una nuova lingua nel lessico.

- API: GET /lexicon/creation/language

- PAYLOAD:
  - key : string

- author: string
- prefix : string
- baseIRI : string

- RESPONSE: [Language](#)

#### 4.5.1.2 Servizi per il recupero dei dati lessicali

In questa sezione vengono riportati i servizi per l'ottenimento di dati lessicali, come ad esempio informazioni su un'entrata lessicale, forma, senso, etimologia ed altri elementi che fanno riferimento al modello OntoLex-Lemon.

##### Get Languages

Chiamata per il recupero della lista delle lingue legate a un lessico di riferimento.

- API: GET /lexicon/data/languages
- PAYLOAD:
  - key : string
- RESPONSE: Array<[Language](#)>

##### Get Lexical Entries

Questa chiamata serve per recuperare l'alberatura delle entrate lessicali e per effettuare ricerche attraverso i filtri dell'albero per recuperare le entrate lessicali in base ai parametri impostati dall'utente

- API: POST /lexicon/data/lexicalEntries
- PAYLOAD:
  - key : string

- BODY:

```
text : string
searchMode : string ("startsWith" | "contains" | "end" | "equals")
type : string
pos : string
formType : string ("flexed" | "entry")
author : string
lang : string
status : string
offset : number
limit : number
```

- RESPONSE: [LexicalFilter](#)

## Get Lexical Entry

Questa chiamata serve per recuperare i dati relativi a una lexical entry, si attiva quando un utente clicca sull'albero delle entrate lessicali, i dati in risposta vengono inviati al componente dell'editor lessicale.

- API: GET /lexicon/data/lexicalEntry

- PAYLOAD:

```
○ id : string
○ key : string
○ aspect: string ("core" | "decomposition" | "vartrans" | "synsem")
```

- RESPONSE: [LexicalEntry](#)

## Get Elements

Questa chiamata serve per recuperare i dati relativi agli elementi "figli" di una lexical entry, ovvero sensi, forme, etimologie ed altri elementi che fanno riferimento all'entrata lessicale selezionata;

questa chiamata viene effettuata quando un utente espande un nodo (lexical entry) dell'albero delle lexical entries.

- API: GET /lexicon/data/elements
- PAYLOAD:
  - id : string
  - key : string
- RESPONSE: [LexicalEntryElements](#)

### Get Filtered Forms

Chiamata per ottenere una lista di forme in base ai parametri di filtering impostati dall'utente

- API: POST /lexicon/data/filteredForms
- PAYLOAD:
  - key : string
- BODY:
  - text : string
  - searchMode : string ("startsWith" | "contains" | "end" | "equals")
  - author : string
  - representationType : string
    - writtenRep
    - phoneticRep
    - pronunciation
    - romanization
    - segmentation
    - transliteration
  - offset : number
  - limit : number
- RESPONSE: [LexicalFilter](#)

## Get Forms

Chiamata utilizzata per ottenere la lista completa delle forme di una Lexical Entry.

- API: GET /lexicon/data/forms
- PAYLOAD:
  - id : string
  - key : string
- RESPONSE: Array<[Form](#)>
- 

## Get Form

Chiamata utilizzata per ottenere i dati di una forma selezionata nell'albero delle entrate lessicali.

- API: GET /lexicon/data/form
- PAYLOAD:
  - id : string
  - aspect : string ("core" | "decomposition" | "vartrans" | "synsem")
  - key : string
- RESPONSE: [Form](#)

## Get Filtered Senses

Chiamata per ottenere una lista di forme in base ai parametri di filtering impostati dall'utente

- API: POST /lexicon/data/filteredSense

- PAYLOAD:
  - key : string
- BODY:
  - text : string
  - searchMode : string ("startsWith" | "contains" | "end" | "equals")
  - author : string
  - type : string
  - field : string
    - definition
    - description
    - gloss
    - senseExample
    - senseTranslation
  - pos : string
  - formType : string
  - lang : string,
  - status : string
  - offset : number
  - limit : number
- RESPONSE: [LexicalFilter](#)

## Get Senses

Chiamata utilizzata per ottenere la lista completa dei sensi di una Lexical Entry.

- API: GET /lexicon/data/senses
- PAYLOAD:
  - id : string
  - key : string
- RESPONSE: Array<[Sense](#)>

## Get Sense

Chiamata utilizzata per ottenere i dati di un senso selezionato nell'albero delle entrate lessicali.

- API: GET /lexicon/data/lexicalSense
- PAYLOAD:
  - id : string
  - aspect : string ("core" | "decomposition" | "vartrans" | "synsem")
  - key : string
- RESPONSE: [Sense](#)

## Get Etymologies

Chiamata utilizzata per ottenere la lista completa delle etimologie di una Lexical Entry

- API: GET /lexicon/data/etimologies
- PAYLOAD:
  - id : string
  - key : string
- RESPONSE: Array<[Etymology](#)>

## Get Etymology

Chiamata utilizzata per ottenere i dati di un'etimologia selezionata nell'albero delle entrate lessicali.

- API: GET /lexicon/data/etymology
- PAYLOAD:
  - id : string
  - key : string

- RESPONSE: [Etymology](#)

### Get Linguistic Relation

Chiamata utilizzata per restituire relazioni linguistiche legate con una lexical entry; in questo caso si usa per ottenere i dati relativi alle *denotes* o ai *cognates* presenti in una lexical entry.

- API: GET /lexicon/data/linguisticRelation
- PAYLOAD:
  - id : string
  - property : string ("denotes", "cognate")
  - key : string
- RESPONSE: Array<[LinguisticRelation](#)>

### Get Generic Relation

Chiamata utilizzata per restituire relazioni linguistiche generiche legate con una lexical entry; in questo caso si usa per ottenere i dati relativi ai *sameAs* o ai *seeAlso* presenti in una lexical entry.

- API: GET /lexicon/data/genericRelation
- PAYLOAD:
  - id : string
  - property : string ("seeAlso", "sameAs")
  - key : string
- RESPONSE: Array<[GenericRelation](#)>

#### 4.5.1.3 Servizi per la modifica dei dati lessicali

In questa sezione sono presenti i servizi per la modifica dei dati lessicali; vengono qui elencati quindi tutti i servizi che portano la firma "update" nelle API di riferimento.

## Update Lexical Entry

Chiamata per modificare un'entrata lessicale

- API: POST /lexicon/update/lexicalEntry
- PAYLOAD:
  - key : string
  - user : string
  - id : string
- BODY:
  - relation : string ("label", "type", "language", "status", "note")
  - value : string
- RESPONSE: [UpdateResponse](#)

## Update Form

Chiamata per modificare una forma di un'entrata lessicale.

- API: POST /lexicon/update/form
- PAYLOAD:
  - key : string
  - user : string
  - id : string
- BODY:
  - relation : string ("type", "note", "writtenRep", "phoneticRep",  
"pronunciation", "romanization", "segmentation",  
"transliteration")
  - value : string

- RESPONSE: [UpdateResponse](#)

## Update Sense

Chiamata per modificare un senso di un'entrata lessicale.

- API: POST /lexicon/update/lexicalSense
- PAYLOAD:
  - key : string
  - user : string
  - id : string
- BODY:
  - relation : string ("note", "usage", "reference", "subject",  
"definition", "description", "explanation",  
"gloss", "senseExample", "senseTranslation")
  - value : string
- RESPONSE: [UpdateResponse](#)

## Update Language

Chiamata per modificare una lingua del lessico.

- API: POST /lexicon/update/language
- PAYLOAD:
  - key : string
  - user : string
  - id : string
- BODY:

```
relation : string ("language", "lexvo", "description")
value : string
```

- RESPONSE: [UpdateResponse](#)

## Update Etymology

Chiamata per modificare un'etimologia del lessico.

- API: POST /lexicon/update/etymology

- PAYLOAD:

```
key : string
user : string
id : string
```

- BODY:

```
relation : string ("note", "hypothesisOf", "label")
value : string
```

- RESPONSE: [UpdateResponse](#)

## Update EtyLink

Chiamata per modificare un link etimologico riferito a un'etimologia.

- API: POST /lexicon/update/etymologicalLink

- PAYLOAD:

```
key : string
user : string
id : string
```

- BODY:  
relation : string ("note", "etyLinkType", "label")  
value : string
- RESPONSE: [UpdateResponse](#)

### Update Linguistic Relation

Chiamata per modificare una relazione di tipo linguistico di un'entrata lessicale, forma, senso, etimologia.

- API: POST /lexicon/update/linguisticRelation
- PAYLOAD:  
key : string  
id : string
- BODY:  
relation : string ("morphology", "decomp", "conceptRef",  
"lexicalRel", "senseRel", "conceptRel",  
"etyLink")  
value : string
- RESPONSE: [UpdateResponse](#)

### Update Generic Relation

Chiamata per modificare una relazione di tipo generico di un'entrata lessicale, forma, senso o etimologia; il servizio è utilizzato quando vengono modificati i riferimenti esterni / informazioni accessorie di un elemento lessicale.

- API: POST /lexicon/update/genericRelation
- PAYLOAD:
  - key : string
  - id : string
- BODY:
  - relation : string ("reference", "bibliography", "attestation", "multimedia")
  - value : string
- RESPONSE: [UpdateResponse](#)

### Synchronize Bibliography

Servizio per sincronizzare eventuali modifiche nella bibliografia caricata su Zotero.

- API: POST /lexicon/update/genericRelation
- PAYLOAD:
  - key : string
  - id : string
  - author : string
  - itemKey : string
- BODY: {}
- RESPONSE: [UpdateResponse](#)

#### 4.5.1.4 Servizi per l'eliminazione dei dati lessicali

In questa sezione sono elencati i servizi per l'eliminazione di dati lessicali.

## Delete Lexical Entry

Chiamata per eliminare un'entrata lessicale

- API: GET /lexicon/delete/lexicalEntry
- PAYLOAD:
  - key : string
  - id : string
- RESPONSE: null

## Delete Form

Chiamata per eliminare una forma a partire da un'entrata lessicale.

- API: GET /lexicon/delete/form
- PAYLOAD:
  - key : string
  - id : string
- RESPONSE: null

## Delete Sense

Chiamata per eliminare un senso a partire da un'entrata lessicale.

- API: GET /lexicon/delete/lexicalSense
- PAYLOAD:
  - key : string
  - id : string

- RESPONSE: null

### **Delete Etymology**

Chiamata per eliminare un'etimologia a partire da un'entrata lessicale.

- API: GET /lexicon/delete/etymology

- PAYLOAD:

key : string

id : string

- RESPONSE: null

### **Delete Etymological Link**

Chiamata per eliminare un link etimologico a partire da un'etimologia di riferimento.

- API: GET /lexicon/delete/etymologicalLink

- PAYLOAD:

key : string

id : string

- RESPONSE: null

### **Delete Language**

Chiamata per eliminare una lingua dal lessico di riferimento.

- API: GET /lexicon/delete/language

- PAYLOAD:  
key : string  
id : string
- RESPONSE: null

### **Delete Bibliography**

Chiamata per eliminare un'entrata bibliografica dall'elemento lessicale di riferimento.

- API: GET /lexicon/delete/bibliography
- PAYLOAD:  
key : string  
id : string
- RESPONSE: null

#### *4.5.1.5 Servizi per il recupero di dati statistici e di vocabolario*

In questa sezione vengono elencati i servizi per il recupero di dati relativi alle statistiche sul lessico, come ad esempio le lingue, tipi di entrate lessicali presenti e le part of speech delle unità lessicali con le relative occorrenze.

### **Get Lexical Entry Types**

Chiamata per ottenere i tipi di entrate lessicali in un lessico di riferimento.

- API: GET /lexicon/statistics/type

- PAYLOAD:  
key : string
- RESPONSE: Array<[StatisticElement](#)>

### Get Namespaces

Chiamata per ottenere gli IRI dei namespace nel lessico (o ontologia) di riferimento.

- API: GET /lexicon/statistics/namespaces
- PAYLOAD:  
key : string
- RESPONSE: Array<[StatisticElement](#)>

### Get Authors

Chiamata per ottenere l'elenco degli autori delle entrate lessicali, con annesso il numero delle occorrenze delle entrate lessicali su cui hanno lavorato.

- API: GET /lexicon/statistics/authors
- PAYLOAD:  
key : string
- RESPONSE: Array<[StatisticElement](#)>

### Get Languages

Chiamata per ottenere l'elenco delle lingue presenti nel lessico, con annesso il numero delle occorrenze delle entrate lessicali a cui fanno riferimento.

- API: GET /lexicon/statistics/languages
- PAYLOAD:  
key : string
- RESPONSE: Array<[StatisticElement](#)>

### Get Pos

Chiamata per ottenere l'elenco delle *part of speech* utilizzate per modificare le entrate lessicali, con annesso il numero relativo delle occorrenze di tali PoS distribuite nel lessico.

- API: GET /lexicon/statistics/languages
- PAYLOAD:  
key : string
- RESPONSE: Array<[StatisticElement](#)>

### Get Status

Chiamata per ottenere l'elenco degli stati delle entrate lessicali, con allegato il numero delle occorrenze di entrate lessicali legate per ogni stato.

- API: GET /lexicon/statistics/status
- PAYLOAD:  
key : string
- RESPONSE: Array<[StatisticElement](#)>

## Get Morphology

Chiamata per ottenere l'elenco dei tratti morfologici da usare nel cambio di PoS o di altri tratti degli elementi lessicali.

- API: GET /lexinfo/data/morphology
- PAYLOAD:  
key : string
- RESPONSE: Array<[MorphologyValue](#)>

## Get Form Type

Chiamata per ottenere l'elenco dei tipi di forme.

- API: GET /ontolex/data/formType
- PAYLOAD:  
key : string
- RESPONSE: Array<[PropertyValue](#)>

### 4.5.1.6 Altri servizi

## Import Lexicon

Chiamata per importare un lessico computazionale in formato *turtle-RDF*.

- API: POST /import/lexicon
- PAYLOAD:
  - key : string
- BODY:
  - InputStream (.ttl file)
- RESPONSE: null

## Export Lexicon

Chiamata per fare un export del lessico.

- API: POST /export/lexicon
- PAYLOAD:
  - key : string
- BODY:
  - fileName : string
  - format : string ("xml", "jsonld", "n3", "ntriples", "nquads", "turtle")
  - subject : string | null
  - predicate : string | null
  - object : string | null
  - contest : string
  - inferred : boolean
- RESPONSE: InputStream

## Query Federation

Chiamata per restituire i risultati di una ricerca effettuata su repository semantici esterni attraverso una SPARQL query.

- API: `POST /fedex/search`
- PAYLOAD:  
`sparqlQuery : string`  
`endpoint : string`
- RESPONSE: [FedexResponse](#)

### 4.5.2 Servizi dei testi (CASH)

I servizi relativi ai testi sono erogati dal servizio di CASH, il quale comunica con un database MySQL per la gestione dei documenti XML caricati sulla piattaforma e delle associazioni tra testo e lessico. I servizi sono divisi in base al tipo di operazioni, si possono riassumere in:

- Servizi per la gestione delle attestazioni
- Servizi per la ricerca all'interno della collezione di testi
- Servizi per la gestione dei testi
- Servizi accessori per il *file-system* e lo status dei servizi

Le API normalmente funzionano previa autorizzazione tramite token, alcune invece (sotto la firma `/public/`) sono senza autorizzazione (utilizzate per la fruizione).

#### 4.5.2.1 Servizi per la gestione delle attestazioni

In questo paragrafo vengono elencati i servizi che si occupano di gestire le attestazioni.

#### Create annotation

Questo servizio serve per creare un'annotazione a partire da un token visualizzato nell'editor dei testi

- API: POST /api/annotation
- PAYLOAD:

```
requestUUID : string
nodeid : number
```
- BODY: [Annotation](#)
- RESPONSE: null

### Update annotation

Questo servizio serve per aggiornare un'annotazione.

- API: PUT /api/annotation
- PAYLOAD:

```
requestUUID : string
```
- BODY: [CreateAnnotationBody](#)
- RESPONSE: null

### Get Unstructured Data

Questo servizio serve per verificare il contenuto non strutturato di un file (plain-text).

- API: POST /api/unstructured
- PAYLOAD:

requestUUID : string

- BODY: [CreateAnnotationBody](#)
- RESPONSE: null

## Create Token

Questo servizio serve per creare un token a partire da una stringa di testo.

- API: POST /api/token
- PAYLOAD:  
requestUUID : string  
nodeid : number
- BODY: [Token](#)
- RESPONSE: null

## Get Tokens

Questo servizio serve per ottenere tutti i token di un testo a partire da un suo identificativo.

- API: GET /api/public/token
- PAYLOAD:  
requestUUID : string  
nodeid : number
- RESPONSE: [TokenResponse](#)

## Get Text

Questo servizio serve per ottenere il contenuto testuale di un file a partire da un suo identificativo.

- API: GET /api/public/gettext
- PAYLOAD:  
    requestUUID : string  
    nodeid : number
- RESPONSE: [TextResponse](#)

## Get Content

Questo servizio serve per ottenere il contenuto XML di un file a partire da un suo identificativo.

- API: GET /api/public/getcontent
- PAYLOAD:  
    requestUUID : string  
    nodeid : number
- RESPONSE: [TextResponse](#)

## Get Annotations

Questo servizio serve per ottenere la lista di attestazioni presenti in un file a partire da un suo identificativo.

- API: GET /api/public/annotation
- PAYLOAD:  
    requestUUID : string

nodeid : number

- RESPONSE: [AnnotationResponse](#)

### Delete Annotation by value

Questo servizio serve per eliminare un'attestazione a partire da un valore relativo a un'entità lessicale.

- API: GET /api/annotationbyvalue
- PAYLOAD:  
requestUUID : string  
value : string
- RESPONSE: null

### Delete Annotation

Questo servizio serve per eliminare un'attestazione a partire dal suo identificativo.

- API: GET /api/annotate
- PAYLOAD:  
requestUUID : string  
annotationID : string
- RESPONSE: null

#### 4.5.2.2 Servizi per la ricerca all'interno della collezione di testi

##### Test Search

Questo servizio serve per effettuare un semplice test per la ricerca, per verificare lo status del servizio.

- API: POST /api/public/testSearch
- PAYLOAD:  
    query : string
- RESPONSE: [TestSearchResponse](#)

##### Search Files

Questo servizio serve per effettuare una ricerca all'interno della collezione di testi presenti in CASH.

- API: POST /api/public/searchFiles
- PAYLOAD: {}
- BODY: [SearchFileRequest](#)
- RESPONSE: [SearchFileResponse](#)

#### 4.5.2.3 Servizi per la gestione dei testi

In questa sezione vengono elencate le API utilizzate da EpiLexO per la gestione dei testi.

## Download File

Questo servizio serve per effettuare un download di una risorsa.

- API: POST /api/public/crud/downloadFile
- PAYLOAD: {}
- BODY: [DownloadRequest](#)
- RESPONSE: [CashResponse](#)

## Upload File

Questo servizio serve per effettuare un upload di una risorsa.

- API: POST /api/crud/uploadFile
- PAYLOAD:
  - requestUUID : string
  - element-id : number
- BODY: FileBinary
- RESPONSE: [UploadResponse](#)

## Update File Metadata

Questo servizio serve per effettuare l'aggiornamento dei metadati di una risorsa testuale.

- API: POST /api/crud/updateMetadata

- PAYLOAD: {}
- BODY: [UpdateMetadataRequest](#)
- RESPONSE: [CashResponse](#)

### **Rename Folder, Rename File**

Servizio per cambiare il nome di una cartella/file all'interno del *file-system* di CASH.

- API: POST /api/crud/renameFolder  
POST /api/crud/renameFile
- PAYLOAD: {}
- BODY: [RenameFolderRequest](#)
- RESPONSE: [CashResponse](#)

### **Move Folder, Move File**

Servizio per eliminare una cartella o file all'interno del *file-system* di CASH.

- API: POST /api/crud/moveFolder  
POST /api/crud/moveFileTo
- PAYLOAD: {}
- BODY: [MoveFileRequest](#) | [MoveFolderRequest](#)

- RESPONSE: [CashResponse](#)

## Delete Metadata

Servizio per eliminare i metadati di un file all'interno di CASH.

- API: POST /api/crud/deleteMetadata
- PAYLOAD: {}
- BODY: [DeleteMetadataRequest](#)
- RESPONSE: [CashResponse](#)

## Create File

Servizio per creare un file vuoto su CASH.

- API: POST /api/crud/createFile
- PAYLOAD: {}
- BODY: [CreateFileRequest](#)
- RESPONSE: [CreateFileResponse](#)

## Copy File

Servizio per copiare un file all'interno del *file-system* di CASH.

- API: POST /api/crud/copyFileTo
- PAYLOAD: {}
- BODY: [CreateFileRequest](#)
- RESPONSE: [CreateFileResponse](#)

### Add Folder

Servizio per copiare un file all'interno del *file-system* di CASH.

- API: POST /api/crud/addFolder
- PAYLOAD: {}
- BODY: [AddFolderRequest](#)
- RESPONSE: [AddFolderResponse](#)

#### 4.5.2.4 Servizi accessori per il *file-system* e lo status dei servizi

In questa sezione vengono elencati i vari servizi accessori che riguardano il controllo sullo stato dei servizi e l'ottenimento del *file-system* con l'alberatura completa dei files presenti in CASH.

### Ping

Servizio per testare la funzionalità dei servizi.

- API: GET /rest/ping
- PAYLOAD: {}

- RESPONSE: string

### Get Document System

Servizio per ottenere l'alberatura completa dei files e delle cartelle presenti in CASH.

- API: GET /api/public/getDocumentSystem
- PAYLOAD:
  - requestUUID : string
- RESPONSE: [GetDocumentSystemResponse](#)

### Get Users

Servizio per ottenere la lista degli utenti che hanno interagito con CASH.

- API: GET /api/getUsers
- PAYLOAD:
  - requestUUID : string
- RESPONSE: GetUsersResponse

### 4.5.3 Servizi di autenticazione (Keycloak)

In questa sezione vengono elencati i servizi relativi all'autenticazione tra i vari servizi, in questo documento verranno riportati precisamente i servizi legano EpiLexO all'infrastruttura di autorizzazioni di Keycloak.

### Search Users

Servizio per ottenere una lista di utenti a partire da una stringa di ricerca.

- API: GET /admin/realms/{idRealm}/users
- PAYLOAD:
  - search : string
- RESPONSE: Array<[UserModel](#)>

### Get User Roles

Servizio per ottenere una lista di ruoli che appartengono a un utente.

- API: GET /admin/realms/{idRealm}/users/{userId}/role-mappings/clients/{clientId}
- PAYLOAD: {}
- RESPONSE: Array<[RolesModel](#)>

### Get User Info

Servizio per ottenere una lista di informazioni relative a un'utenza.

- API: GET /admin/realms/{idRealm}/users/{usersId}/
- PAYLOAD: {}

- RESPONSE: Array<[UserModel](#)>

### Get Users By Role

Servizio per ottenere una lista di utenti a partire da un ruolo.

- API: GET /admin/realms/{idRealm}/clients/{clientId}/roles/{roleName}/users/
- PAYLOAD: {}
- RESPONSE: Array<[UserModel](#)>

### Create User

Servizio per ottenere una lista di utenti a partire da un ruolo.

- API: POST /admin/realms/{idRealm}/users/
- PAYLOAD: {}
- BODY: CreateUserRequestParams
- RESPONSE: null

### Delete User

Servizio per eliminare un'utenza dal client.

- API: DELETE /admin/realms/{idRealm}/users/{idUser}

- PAYLOAD: {}
- RESPONSE: null

### **Update User**

Servizio per eliminare un'utenza dal client.

- API: PUT /admin/realms/{idRealm}/users/{idUser}
- PAYLOAD: {}
- BODY: [UpdateUserRequest](#)
- RESPONSE: null

### **Get Client Roles**

Servizio per ottenere una lista di ruoli all'interno di un client.

- API: GET /admin/realms/{idRealm}/clients/{clientId}/roles/
- PAYLOAD: {}
- RESPONSE: Array<[RolesModel](#)>

### **Get Client Info**

Servizio per ottenere una lista di informazioni relative a un client.

- API: GET /admin/realms/{idRealm}/clients/{clientId}/
- PAYLOAD: {}
- RESPONSE: Array<[ClientRepresentation](#)>

### Set Roles to User

Servizio per ottenere una lista di informazioni relative a un client.

- API: POST  
/admin/realms/{idRealm}/users/{userId}/role-mappings/clients/{clientId}/
- PAYLOAD: {}
- BODY: [RolesModel](#)
- RESPONSE: null

### Delete Roles to User

Servizio per ottenere una lista di informazioni relative a un client.

- API: POST  
/admin/realms/{idRealm}/users/{userId}/role-mappings/clients/{clientId}/
- PAYLOAD: {}
- BODY: [RolesModel](#)

- RESPONSE: null

#### 4.5.4 Modelli dei dati

In questa sezione vengono elencati tutti i tipi di dati in risposta alle richieste elaborate dalla piattaforma. Ogni modello viene elencato con tutti i tipi primitivi che lo costituiscono e la descrizione di ogni campo.

##### 4.5.4.1 LexO Data Model

In questa sezione vengono elencate le interfacce/modelli dei dati relativi agli elementi del lessico computazionale, ovvero tutti i dati provenienti da LexO-Server, che, come evidenziato nella sezione 2, sono in larga misura conformi alle specifiche del modello Ontolex-lemon e sue estensioni

#### Lexical Entry

Una *lexical entry* è un oggetto JSON composto da vari campi che contengono le informazioni relative a un'entrata lessicale. Essa costituisce l'unità base del lessico computazionale.

CAMPO	TIPO	DESCRIZIONE
creator	string   null	Nome del creatore della lexical entry
lastUpdate	string	Timestamp dell'ultima modifica
creationData	string	Timestamp della creazione
confidence	number	Affidabilità delle informazioni sulla lexical entry. Può avere due valori (-1 e 0)
lexicalEntry	string	IRI della lexical entry
label	string	Etichetta contenente il valore da visualizzare nell'albero
type	Array<string>	Contiene i valori relativi al tipo della lexical entry, può contenere i seguenti valori:  - LexicalEntry - Word

		- Affix - MultiWordExpression - Cognate - Etymon
pos	string   null	Part of Speech della lexical entry
language	string   null	Lingua a cui appartiene la lexical entry
stemType	string   null	Radice della lexical entry
morphology	Array< <a href="#">Morphology</a> >	Tratti morfologici della lexical Entry
links	Array< <a href="#">Link</a> >	Riferimento esterni, bibliografia, note, seeAlso, sameAs
author	string	Autore dell'entrata
revisor	string   null	Revisore dell'entrata
status	string	Stato dell'entrata lessicale, gli stati possibili sono:  - working - completed - reviewed
note	string	Note relative all'entrata lessicale
completionDate	string	Data di completamento dell'entrata lessicale
revisionDate	string	Data di revisione dell'entrata lessicale

## Form

Una *form* è un oggetto JSON composto da vari campi che contengono le informazioni relative a una forma di un'entrata lessicale.

CAMPO	TIPO	DESCRIZIONE
creator	string   null	Nome del creatore della lexical entry
lastUpdate	string	Timestamp dell'ultima modifica
creationDate	string	Timestamp della creazione

confidence	number	Affidabilità delle informazioni sulla forma. Può avere due valori (-1 e 0)
form	string	IRI della forma
inheritedMorphology	Array< <a href="#">Morphology</a> >	Tratti morfologici (PoS) ereditati dalla lexical entry di riferimento
lexicalEntry	string	IRI della lexical entry
lexicalEntryLabel	string	Etichetta con il titolo della lexical entry madre
label	Array< <a href="#">Property</a> >	Elementi caratterizzanti di una forma
type	Array<string>	Contiene i valori relativi al tipo della lexical entry, può contenere i seguenti valori:  <ul style="list-style-type: none"> <li>- LexicalEntry</li> <li>- Word</li> <li>- Affix</li> <li>- MultiWordExpression</li> <li>- Cognate</li> <li>- Etymon</li> </ul>
language	string   null	Lingua a cui appartiene la forma (ereditata dalla lexical entry)
morphology	Array< <a href="#">Morphology</a> >	Tratti morfologici della forma
links	Array< <a href="#">Link</a> >	Riferimento esterni, bibliografia, note, seeAlso, sameAs
note	string	Note relative all'entrata lessicale

## Sense

Un *sense* è un oggetto JSON composto da vari campi che contengono le informazioni relative a un senso di un'entrata lessicale.

CAMPO	TIPO	DESCRIZIONE
creator	string   null	Nome del creatore della lexical entry
lastUpdate	string	Timestamp dell'ultima modifica
creationDate	string	Timestamp della creazione

confidence	number	Affidabilità delle informazioni sul senso
explanation	string   null	Spiegazione del senso
sense	string	IRI del senso
senseExample	string   null	Esempio di uso del senso
senseTranslation	string   null	Traduzione del senso
gloss	string   null	Glossa di un significato / senso
links	Array< <a href="#">Link</a> >	Riferimento esterni, bibliografia, note, seeAlso, sameAs
status	string	Stato dell'entrata lessicale, gli stati possibili sono:  - working - completed - reviewed
note	string	Note relative all'entrata lessicale
usage	string	Uso del senso

### **EtymologySet, EtyLink ed Etymology**

Un **EtymologySet** è un oggetto che viene restituito quando si clicca su un'etimologia nell'albero delle entrate lessicali; esso contiene due campi relativi all'etimologia in sé e agli etylinks (riferimenti dell'etimologia) che essa contiene

CAMPO	TIPO	DESCRIZIONE
etyLinks	Array<EtyLink>	Array con tutti gli etyLinks legati all'etimologia
etymology	Etymology	Oggetto contenente i dati dell'etimologia

Ecco un modello di **EtyLink** e, successivamente, di **Etymology**.

CAMPO	TIPO	DESCRIZIONE
-------	------	-------------

confidence	number	Affidabilità delle informazioni sull'etimologia
creationDate	string	Data di creazione
creator	string	Creatore del link
etyLinkType	string	Tipo di etyLink. Può avere due valori (inheritance, borrowing)
etySource	string	IRI all'etymon di partenza
etySourceLabel	string	Label dell'etymon di partenza
etyTarget	string	IRI all'etymon di destinazione
etyTargetLabel	string	Label dell'etymon di destinazione
etymologicalLink	string	IRI dell'etyLink
externalIRI	boolean	Se è un IRI esterno o no
label	string	Etichetta dell'etyLink
lastUpdate	string	Ultima modifica
note	string	Note dell'etyLink

CAMPO	TIPO	DESCRIZIONE
confidence	number	Affidabilità delle informazioni sull'etimologia
creationDate	string	Data di creazione
creator	string	Creatore del link
etymology	string	IRI dell'etimologia
hypotesisOf	string	Autore dell'etimologia
label	string	Etichetta dell'etyLink
lastUpdate	string	Ultima modifica
links	Array< <a href="#">Link</a> >	Riferimento esterni, bibliografia, note, seeAlso, sameAs
note	string	Note dell'etyLink

## BibliographyItem

Corrisponde a un elemento bibliografico, contiene tutte le informazioni relative ad un riferimento esterno o cartaceo.

CAMPO	TIPO	DESCRIZIONE
author	string	Autore / autrice del libro
bibliography	string	IRI della bibliografia
confidence	number	Confidenza del riferimento bibliografico
creationDate	stringa	Data di creazione
creator	stringa	Creatore del riferimento bibliografico
date	string	Data di pubblicazione del libro
id	string	Identificativo risorse bibliografica
lastUpdate	string	Ultima modifica
note	string	Note sull'elemento bibliografico
seeAlsoLink	string	Collegamento ad un elemento di tipo seeAlso
textualReference	string	Riferimento testuale
title	string	Titolo opera bibliografica
url	string	URL alla risorsa

## Language

L'oggetto Language contiene i dati relativi a un linguaggio nel lessico, viene restituito quando viene fatta una richiesta di creazione di una nuova lingua o quando si vuole ottenere una lista di lingue presenti nel lessico

CAMPO	TIPO	DESCRIZIONE
catalog	Array<string>	Array contenente i base IRI a cui fa riferimento la lingua

confidence	number	Confidenza sulla conoscenza della lingua
creationDate	string	Data di creazione
creator	string	Creatore lingua
description	string	Descrizione lingua
entries	number	Numero di entrate che fanno riferimento alla lingua
label	string	Etichetta della lingua
language	string	IRI del linguaggio
lastUpdate	string	Timestamp ultima modifica
lexvo	string	URI lexvo del linguaggio

L'oggetto Language contiene i dati relativi a un linguaggio nel lessico, viene restituito quando viene fatta una richiesta di creazione di una nuova lingua o quando si vuole ottenere una lista di lingue presenti nel lessico

CAMPO	TIPO	DESCRIZIONE
catalog	Array<string>	Array contenente i base IRI a cui fa riferimento la lingua
confidence	number	Confidenza sulla conoscenza della lingua
creationDate	string	Data di creazione
creator	string	Creatore lingua
description	string	Descrizione lingua
entries	number	Numero di entrate che fanno riferimento alla lingua
label	string	Etichetta della lingua
language	string	IRI del linguaggio
lastUpdate	string	Timestamp ultima modifica
lexvo	string	URI lexvo del linguaggio

## LexicalFilter

Oggetto che è in risposta alla richiesta di un elenco di Lexical Entries fatta dal filtro dell'albero delle entrate lessicali

CAMPO	TIPO	DESCRIZIONE
list	Array< <a href="#">LexicalEntry</a>   <a href="#">Form</a>   <a href="#">Sense</a>   <a href="#">LiLaResponse</a> >	Lista delle lexical entries, forme o sensi
totalHits	number	Numero di entrate lessicali ottenute dalla richiesta

## LexicalEntryElements

Una lista di elementi corrispondenti ai "figli" di un'entrata lessicale

CAMPO	TIPO	DESCRIZIONE
label	string	Etichetta corrispondente al tipo di elemento; può avere questi valori:  - form - sense - frame - lexicalConcept - concept - subterm - constituent - etymology
count	number	Counter degli elementi presenti
hasChildren	boolean	Se ci sono nodi figli

## Morphology

L'oggetto morphology è un oggetto JSON che contiene una coppia chiave valore corrispondente al tratto morfologico e al suo valore.

CAMPO	TIPO	DESCRIZIONE
-------	------	-------------

trait	string	tratto morfologico
value	string	valore del tratto morfologico

## Link

L'oggetto Link è un oggetto JSON che contiene una chiave per determinare la tipologia di informazione esterna riferita all'entrata lessicale/forma/senso e gli oggetti legati a quel tipo di informazione.

CAMPO	TIPO	DESCRIZIONE
type	string	Tipologia di dato, può avere i seguenti valori: <ul style="list-style-type: none"> <li>- Reference (seeAlso, sameAs)</li> <li>- Bibliography (references)</li> <li>- Multimedia</li> <li>- Attestation</li> <li>- Other</li> </ul>
elements	Array<LinkElement>	Informazioni sul tipo di elemento

Viene qui riportata la struttura di un **LinkElement**

CAMPO	TIPO	DESCRIZIONE
label	string	Etichetta con il nome del tipo di dato
count	Array<LinkElement>	Quanti elementi ci sono
hasChildren	boolean	Se ci sono dei nodi figli per questo tipo di elemento

## Property

L'oggetto Property è un oggetto JSON che contiene una coppia chiave valore che rappresentano proprietà di diversi oggetti (varia in base a dove viene utilizzato)

CAMPO	TIPO	DESCRIZIONE
propertyID	string	Tipologia di dato / IRI del tratto morfologico
propertyValue	string	Valore della chiave

## MorphologyValue

L'oggetto MorphologyValues è un oggetto JSON che contiene tre campi di tipo stringa con l'IRI della proprietà morfologica e una lista di valori a cui fanno riferimento alla proprietà.

CAMPO	TIPO	DESCRIZIONE
propertyDescription	string	Descrizione del tratto morfologico
propertyId	string	IRI del tratto morfologico
propertyLabel	string	Etichetta del tratto morfologico
propertyValues	Array< <a href="#">PropertyValue</a> >	Valori legati al tratto morfologico

## PropertyValue

L'oggetto PropertyValue è un oggetto JSON che contiene tre campi stringa relativi alla descrizione della proprietà appartenente al tratto morfologico, con annesso IRI ed etichetta

CAMPO	TIPO	DESCRIZIONE
valueDescription	string	Descrizione del valore
valueId	string	IRI del valore
valueLabel	string	Etichetta del valore

## LinguisticRelation, Generic Relation

Una *linguistic relation* corrisponde a una relazione linguistica che occorre tra un'entità Lexical Entry e una sua proprietà.

CAMPO	TIPO	DESCRIZIONE
entity	string	IRI dell'entità
entityType	Array<string>	Tipologia dell'entità ("Word", "Cognate")
inferred	boolean	Valore booleano che indica se l'entità è stata inferita
label	string	Etichetta dell'entità
link	string	IRI della relazione
linkType	string	Tipologia del link ("internal", "external")

## UpdateResponse

Una *UpdateResponse* è un oggetto che viene restituito quando viene effettuata una modifica al lessico e riporta un unico campo contenente il timestamp della modifica appena avvenuta

CAMPO	TIPO	DESCRIZIONE
lastUpdate	string	Timestamp ultima modifica

## StatisticElement

Uno *StatisticElement* è un tipo di dato corrispondente in un oggetto costituito da una coppia chiave valore, che rappresentano il tipo di dato e la sua occorrenza statistica nel lessico

CAMPO	TIPO	DESCRIZIONE
label	string	Etichetta della proprietà
count	number	Numero di occorrenze

### LiLaResponse

Corrisponde a un modello di dati in risposta a una richiesta fatta a un repository semantico esterno (in questi casi LiLa)

CAMPO	TIPO	DESCRIZIONE
key	string	Chiave della proprietà
value	string	Valore della proprietà

#### 4.5.4.2 CASH Data Model

In questa sezione vengono elencati tutti i modelli dei dati relativi alla gestione delle attestazioni tra testo epigrafico e lessico, ovvero i tipi di dati ottenuti in risposta alle richieste al back-end *CASH-server*.

### Annotation

Questo modello corrisponde a un *body* da mandare come richiesta per creare una nuova attestazione tra testo e lessico. L'oggetto contiene valori come l'IRI della forma attestata, l'intervallo di caratteri del token in cui è attestata l'entità lessicale ed eventuali metadati.

CAMPO	TIPO	DESCRIZIONE
layer	string	Livello attestazione
value	string	Valore contenuto dall'attestazione; generalmente, contiene l'IRI della forma
imported	boolean	Booleano per indicare se la risorsa è importata o meno

attributes	<a href="#">AnnotationAttributes</a>	Attributi dell'attestazione
spans	Array< <a href="#">AnnotationSpan</a> >	Intervallo corrispondente alla sotto-stringa del token a cui fa riferimento
id	number	Identificativo dell'attestazione

### UnstructuredBody

Questo modello corrisponde a un *body* da mandare come richiesta per chiedere che tipologia di dati non strutturati sono presenti in un file.

CAMPO	TIPO	DESCRIZIONE
unstructured	<a href="#">AnnotationAttributes</a>	Attributi dell'attestazione
element-id	number	Identificativo dell'attestazione

### TextResponse

Risposta mandata da CASH alla richiesta di ottenere una lista di tokens presenti in un documento epigrafico / file nuovo.

CAMPO	TIPO	DESCRIZIONE
requestUUID	string	Identificativo della richiesta
text	string	Stringa contenente il testo dell'intero file / il contenuto XML di un file

### AnnotationResponse

Risposta mandata da CASH alla richiesta di ottenere una lista di tokens presenti in un documento epigrafico / file nuovo.

CAMPO	TIPO	DESCRIZIONE
requestUUID	string	Identificativo della richiesta

annotation	Array< <a href="#">Annotation</a> >	Lista con tutte le attestazioni presenti in un file
------------	-------------------------------------	---

### TokenResponse

Risposta mandata da CASH alla richiesta di ottenere una lista di tokens presenti in un documento epigrafico / file nuovo.

CAMPO	TIPO	DESCRIZIONE
requestUUID	string	Identificativo della richiesta
tokens	Array< <a href="#">Token</a> >	Lista con tutti i token del file

### Token

Questo modello corrisponde a un *body* da mandare come richiesta per creare un token all'interno di un file a partire da una stringa di testo.

CAMPO	TIPO	DESCRIZIONE
text	string	Testo selezionato dell'attestazione
xml-id	number	Identificativo del token
position	number	Posizione del token all'interno del testo
begin	number	Porzione iniziale dell'intervallo corrispondente alla sottostringa del testo dell'annotazione
end	number	Porzione finale dell'intervallo corrispondente alla sottostringa del testo dell'annotazione
node	number	Identificativo corrispondente all'ID del testo
source	string	Origine del testo
imported	boolean	Se il testo è stato importato o creato da zero
id	number	Identificativo del testo

## AnnotationAttributes

Questo modello corrisponde a un *body* da mandare come richiesta per creare una nuova attestazione tra testo e lessico. L'oggetto contiene valori come l'IRI della forma attestata, l'intervallo di caratteri del token in cui è attestata l'entità lessicale ed eventuali metadati.

CAMPO	TIPO	DESCRIZIONE
author	string	Autore dell'attestazione
creator	string	Creatore dell'attestazione
note	string	Nota dell'attestazione
confidence	number	Confidenza dell'attestazione
form_id	string	Nome dell'istanza lessicale
validity	string	Validità dell'attestazione
label	string	Etichetta da visualizzare
timestamp	string	Timestamp della creazione dell'attestazione
bibliography	Array< <a href="#">AnnotationBibliography</a> >	Elementi bibliografici legati all'attestazione
node_id	number	Identificativo del nodo in cui è avvenuta l'attestazione; corrisponde al testo

## AnnotationBibliography

Corrisponde ad un elemento bibliografico relativo ad un'attestazione presente in un testo.

CAMPO	TIPO	DESCRIZIONE
DOI	string	Codice identificativo (Digital Object Identifier)
key	string	Valore contenuto dall'attestazione; generalmente, contiene l'IRI della forma

url	string	Eventuale URL alla risorsa
ISSN	string	International Standard Serial Number
date	string	Data pubblicazione
note	string	Note sull'elemento bibliografico
tags	Array< <a href="#">BibliographyTags</a> >	Lista di tags a cui appartiene il testo
extra	string	Informazioni aggiuntive
issue	string	Note su varie modifiche durante la stampa di un'edizione
pages	string	Pagine del libro / rivista
title	string	Titolo dell'elemento bibliografico
rights	string	Diritti d'autore
series	string	Serie di appartenenza
volume	string	Volume di appartenenza
creators	Array< <a href="#">BibliographyCreator</a> >	Lista dei creatori dell'opera
itemType	string	Tipologia della risorsa
language	string	Lingua
dateAdded	string	Timestamp dell'aggiunta della risorsa
relations	object	Eventuali relazioni con altre opere
accessDate	string	Data di accesso
callNumber	string	Numero di locazione in libreria
seriesText	string	Serie a cui appartiene la risorsa
shortTitle	string	Versione breve del titolo
textualRef	string	Riferimenti testuali
collections	Array<string>	Lista delle collezioni a cui appartiene
seriesTitle	string	Titolo della serie
abstractNote	string	Note riassuntive sull'opera
dateModified	string	Data di modifica
libraryCatalog	string	Catalogo bibliografico di appartenenza
archiveLocation	string	Luogo di conservazione e

		archiviazione
publicationTitle	string	Titolo pubblicazione
journalAbbreviation	string	Abbreviazione del giornale che ospita l'articolo

### BibliographyTags

È una lista di oggetti contenenti un solo campo, corrispondente al tag che contiene l'elemento bibliografico.

CAMPO	TIPO	DESCRIZIONE
tag	string	Tag a cui appartiene l'opera bibliografica

### BibliographyCreator

Una lista di oggetti che contengono nome, cognome e tipologia di contributo di tale soggetto alla stesura dell'opera bibliografica.

CAMPO	TIPO	DESCRIZIONE
lastName	string	Cognome del creatore
firstName	string	Nome del creatore
creatorType	string	Contributo del soggetto

### AnnotationSpan

Lista di oggetti contenente gli indici corrispondenti all'intervallo della stringa di testo selezionata per effettuare l'attestazione

CAMPO	TIPO	DESCRIZIONE
start	number	Parte iniziale della stringa
end	number	Parte finale della stringa

### TestSearchResponse

Lista contenente un path relativo a un file a partire da una query.

CAMPO	TIPO	DESCRIZIONE
paths	Array<string>	Path del file

### MetadataModel

Oggetto contenente coppie chiave valore, ognuna corrispondente a un metadato ben preciso.

CAMPO	TIPO	DESCRIZIONE
~metadata_name_n~	string	Campo di un metadato

### SearchFileRequest

Body per fare una richiesta di filtraggio sulla collezione dei testi.

CAMPO	TIPO	DESCRIZIONE
requestUUID	string	Identificativo richiesta
contains	boolean	Ricerca all'interno della stringa
metadata	<a href="#">MetadataModel</a>	Oggetto contenente metadati utili per la ricerca
search-text	string	Stringa di testo da cercare
start-width	boolean	Ricerca a partire dall'inizio di una stringa

user-id	number	Identificativo utente
import-date	string	Timestamp della data di riferimento per la ricerca
exact-date	boolean	Filtro per la data esatta
from-date	boolean	Filtro a partire da una certa data
util-date	boolean	Filtro sulla data

### SearchFileResponse

Oggetto contenente la risposta alla chiamata per il filtraggio dei testi all'interno della collezione.

CAMPO	TIPO	DESCRIZIONE
requestUUID	string	Identificativo richiesta
results	number	Numero dei risultati restituiti
files	Array< <a href="#">File</a> >	Oggetto contenente metadati utili per la ricerca

### File

Oggetto che corrisponde ad un file caricato nella piattaforma; a questo livello sono presenti soltanto informazioni di tipo *sistemistico* (path del file, id, metadati...); non sono presenti informazioni relative al testo in sé.

CAMPO	TIPO	DESCRIZIONE
path	string	Path del file
name	string	Nome del file
metadata	<a href="#">MetadataModel</a>	Metadati del file
element-id	number	Identificativo del file

### DownloadRequest, RemoveRequest, DeleteMetadataRequest, AddFolderRequest

Richiesta per effettuare il download / rimozione di una risorsa o dei suoi metadati. Viene usata anche come richiesta per aggiungere una cartella al *file-system*.

CAMPO	TIPO	DESCRIZIONE
requestUUID	string	Identificativo richiesta
user-id	number	Identificativo utente
element-id	number	Identificativo del file

### CashResponse

Oggetto restituito da CASH come risposta alla richiesta di download di una risorsa.

CAMPO	TIPO	DESCRIZIONE
requestUUID	string	Identificativo richiesta
response-status	number	Stato della risposta

### UploadResponse, CreateFileResponse, AddFolderResponse

Oggetto restituito da CASH come risposta alla richiesta di upload / creazione di una risorsa.

CAMPO	TIPO	DESCRIZIONE
requestUUID	string	Identificativo richiesta
node	<a href="#">Node</a>	Nodo del <i>file-system</i> di CASH
response-status	number	Stato della risposta

### Node

Oggetto corrispondente a una risorsa caricata in CASH visualizzabile nell'albero dei testi.

CAMPO	TIPO	DESCRIZIONE
name	string	Nome del file
path	string	Path virtuale del file
children	Array< <a href="#">Node</a> >	Eventuali figli del nodo
metadata	<a href="#">MetadataModel</a>	Metadati del nodo

### UpdateMetadataRequest

Corrisponde alla richiesta per l'aggiornamento dei metadati di un file presente in CASH.

CAMPO	TIPO	DESCRIZIONE
requestUUID	string	Identificativo richiesta
metadata	<a href="#">MetadataModel</a>	Metadati del nodo
user-id	number	Identificativo dell'utente che ha fatto la richiesta
element-id	number	Identificativo del file

### RenameFolderRequest

Body per la richiesta di rinominazione di una cartella

CAMPO	TIPO	DESCRIZIONE
requestUUID	string	Identificativo richiesta
rename-string	string	Stringa da applicare alla rinominazione
user-id	number	Identificativo dell'utente che ha fatto la richiesta
element-id	number	Identificativo del file

### MoveFileRequest, MoveFolderRequest

Body per la richiesta di spostamento di una cartella / file all'interno del *file-system* di CASH.

CAMPO	TIPO	DESCRIZIONE
requestUUID	string	Identificativo richiesta
target-id	number	Identificativo del file/cartella di destinazione
user-id	number	Identificativo dell'utente che ha fatto la richiesta
element-id	number	Identificativo del file

### CreateFileRequest

Body per la richiesta di spostamento di una cartella / file all'interno del *file-system* di CASH.

CAMPO	TIPO	DESCRIZIONE
requestUUID	string	Identificativo richiesta
filename	string	Nome del file da creare
user-id	number	Identificativo dell'utente che ha fatto la richiesta
element-id	number	Identificativo del file

### GetDocumentSystemResponse

Corrisponde alla risposta di CASH alla richiesta di ottenere l'alberatura completa dei files e delle cartelle.

CAMPO	TIPO	DESCRIZIONE
requestUUID	string	Identificativo richiesta

results	number	Numero dei risultati restituiti
files	Array< <a href="#">Node</a> >	Oggetto contenente i nodi del <i>file-system</i>

### GetUsersResponse

Lista con all'interno l'elenco degli utenti che hanno interagito con CASH.

CAMPO	TIPO	DESCRIZIONE
requestUUID	string	Identificativo richiesta
results	number	Numero dei risultati restituiti
users	Array<User>	Oggetto contenente i nodi del <i>file-system</i>

### User

Oggetto contenente identificativo, nome e cognome di un'utenza che ha interagito con CASH

CAMPO	TIPO	DESCRIZIONE
user-id	number	Identificativo utente
first-name	string	Nome dell'utente
last-name	string	Cognome dell'utente

#### 4.5.4.3 Keycloak Data Model

In questa sezione vengono elencati tutti i modelli dei dati relativi al contesto di Keycloak per la gestione delle autenticazioni e dei ruoli che hanno gli utenti; quest'ultimo aspetto conta nel caso si volessero fare diverse distinzioni tra le figure che possono operare all'interno dell'editor.

## UserModel

Oggetto JSON che contiene diversi dati relativi a un'utenza.

CAMPO	TIPO	DESCRIZIONE
createdTimestamp	number	Timestamp creazione utente
disableableCredentialTypes	Array<any>	Lista con parametri per disabilitare tipologie di credenziali
emailVerified	boolean	Se l'utenza è verificata
enabled	boolean	Se l'utenza è abilitata
id	string	Identificativo utenza
notBefore	number	Parametro per la gestione dei logout
requiredActions	Array	Lista di azioni da compiere su un'utenza
totp	boolean	Autenticazione a due fattori
username	string	Nome utente

## RolesModel

Oggetto JSON che contiene diversi dati relativi a un'utenza.

CAMPO	TIPO	DESCRIZIONE
clientRole	boolean	Booleano per capire se è un ruolo interno del client
composite	boolean	Booleano per capire se è un ruolo composito
containerId	string	Identificativo del client
id	string	Identificativo del ruolo
name	string	Nome ruolo

## CreateUserRequestParam

Oggetto JSON per la richiesta di creazione di un'utenza.

CAMPO	TIPO	DESCRIZIONE
enabled	boolean	Per capire se un'utenza è abilitata
email	string	Email dell'utenza
username	string	Nome utente
credentials	Array< <a href="#">Credential</a> >	Lista con credenziali legate all'utenza

## Credential

Credenziali relative ad un'utenza.

CAMPO	TIPO	DESCRIZIONE
type	string	Tipologia credenziale
value	string	Password o valore credenziale
temporary	boolean	Se la credenziale è temporanea

## UpdateUserRequest

Body per la richiesta di aggiornamento delle informazioni di un'utenza.

CAMPO	TIPO	DESCRIZIONE
enabled	boolean	Se la credenziale è temporanea o no
email	string	Email utenza
username	string	Nome utenza

## 4.5 Problemi aperti e sviluppi futuri

Attualmente, l'applicazione web è da considerarsi completa, con qualche piccola funzionalità da integrare, una tra queste è la gestione dei concetti lessicali (Lexical Concept) trattati con il vocabolario SKOS. La gestione di questi concetti consentirebbe all'utente e allo studioso di poter aggiungere un livello di astrazione in più per poter rappresentare meglio le informazioni relative a un lessico / materiale epigrafico. Una delle funzionalità a cui si sta attualmente pensando è quella relativa ad un export in formato RDF di un testo XML caricato sulla piattaforma: l'idea sarebbe quella di riportare le informazioni di un testo attraverso la sintassi soggetto-predicato-oggetto come nella struttura RDF di tipo ontologico, per consentire una migliore fruizione delle informazioni da parte delle macchine che devono elaborare le informazioni. Altra caratteristica importante, ancora in fase di sviluppo e di modellizzazione, è quella relativa alla ricerca su più SPARQL endpoint, in modo da avere accesso a molte più informazioni. Una della modalità potrebbe essere quella di poter "configurare" gli endpoint attraverso a un meccanismo di configurazione generale della piattaforma, in modo da poter scegliere arbitrariamente quali endpoint selezionare e quali magari disattivare o togliere. Lo stesso discorso può valere anche nella scelta dei namespaces, in modo da poter etichettare in maniera personalizzata la propria base di dati.

Attualmente si sta lavorando allo sviluppo di una seconda piattaforma dedicata alla fruizione dei dati prodotti dalla piattaforma di editing dei lessici/testi, in modo che qualsiasi utente possa esplorare questi dati senza particolari competenze tecniche in maniera: l'interfaccia infatti dovrà essere pensata per essere il più accessibile possibile in modo che anche gli utenti meno esperti possano esplorare le collezioni di dati in maniera semplice e intuitiva.

La piattaforma EpiLexo si pone come una soluzione innovativa e modulare per la creazione, gestione e fruizione di materiali linguistici per lingue e arcaiche in un ambiente collaborativo multi-utente, rendendo accessibili e facilmente fruibili le risorse digitali per le lingue dell'Italia antica, e machine-understandable/ machine actionable, nella speranza che questo lavoro possa apportare un contributo, anche minimo, all'interno della comunità dell'epigrafia digitale e della linguistica storica.

## 5. Bibliografia

1. M. Berti. "Perseus 2009. Rappresentazione digitale dei testi frammentari" and "Le tecnologie digitali e l'epigrafia: esempi applicativi": *Antichità Classiche e Digital Humanities. Le discipline antichistiche nella rivoluzione digitale*, Università di Roma Tor Vergata, October 20-26, 2009
2. Bellandi, A. Building linked lexicography applications with LexO-server. *Digital Scholarship in the Humanities*, Volume 38, Issue 3, September 2023, Pages 937–952, <https://doi.org/10.1093/lhc/fqac095>
3. G. Canfora, D. Di Fatta, Giovanni Pilato, *Ontologie e Linguaggi Ontologici per il Web Semantico*, Rapporto Tecnico N.6: RT-ICAR-PA-04-06, aprile 2004, Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni
4. Julia Bosque-Gil, Jorge Gracia, John McCrae, Philipp Cimiano, Sander Stolk, Fahad Khan, Katrien Depuydt, Jesse de Does, Francesca Frontini, Ilan Kernerman, *The OntoLex Lemon Lexicography Module*, 17 Settembre 2019, W3C Community Group, Final Report
5. Vannini, Lucia. "Review of papyri.info." *RIDE* 9 (2018). doi: 10.18716/ride.a.9.4. Accessed: 11.03.2021.
6. Michael Zellmann-Rohrer, *Review: Papyri.info: A Searchable Database of Papyri and Translations Online*, Sep 24, 2018 <https://classicalstudies.org/scs-blog/michael-zellmann-rohrer/review-papyriinfo-searchable-database-papyri-and-translations> - Ultimo accesso: 23/03/2021 ore 15:40
7. Prag, Jonathan R. W. and Chartrand, James. "19 I. Sicily: Building a Digital Corpus of the Inscriptions of Ancient Sicily". *Crossing Experiences in Digital Epigraphy*, edited by Annamaria De Santis and Irene Rossi, Warsaw, Poland: De Gruyter Open Poland, 2019, pp. 240-252. <https://doi.org/10.1515/9783110607208-020>