



KDI ● **Knowledge and Data Integration**

L1-2 Importer and Excel file

Phase: 2.Tools and Tutorials

W8.L15.T15.2.3

Contents

- 1 Excel Importer**
- 2 Excel file**
- 3 Example of excel file**

Contents

- 1 Excel Importer**
- 2 Excel file
- 3 Example of excel file

Import functionality of UKC

Recall the steps involved in the **L1-2 annotation task**, where L1 represent concepts in the UKC and L2 represent the lexicon-semantic elements, which may include production of an Import File for **new concepts/senses** which have to be imported in order to integrate it with the UKC.

This session will provide the understanding of

- creation of such import file in the excel format
- import of the excel file for integration of new knowledge elements into the UKC.

Excel Importer

The goal is to import only XLS files, as the name suggests, in the knowledge base. This importing pipeline has the following features and functionalities,

- the header of the XLS file is used to detect the schema version and thus it is mandatory. It also checks for the file existence, sheets existence.
- complex type, concept relations, domains, gaps, relations, senses, and provenances can be stored.
- can import either a single or a batch of files, and before writing to disk the single or entire batch are validated to prevent half baked imports. If anything goes wrong a rollback occurs.
- accepts Excel files with **.xls** extension and thus a file can have at most 65536 rows per sheet.

Contents

- 1 Excel Importer
- 2 Excel file**
- 3 Example of excel file

File

The excel input file contains 6 sheets, namely, senses, relations, concepts_move, gaps, etypes and domains.

Definition

Given F as a file, formally,

$$F = \{ \text{'senses'}, \text{'relations'}, \text{'concepts_move'}, \text{'gaps'}, \text{'etypes'}, \text{'domains'} \}$$

For L1-2 levels importing the sheets, also allow the import of provenance data respectively ,used are as follows,

- senses, creating new concepts and importing senses
- relations, it is used to establish or modify existing relationships between concepts in the knowledge graph.
- concepts_move
- gaps, importing the lexical gaps.

Contents of the senses sheet

Senses - Column name	Column description
cased word lemma	represents the name of the entry to import and can be expressed as a string and also namespaces are supported.
word forms (optional)	used to indicate exceptions in the forms of the word, i.e., if the word to be imported is 'child', the field word forms can be used to indicate that the plural is 'children' and not 'childs'.
concept uk id (mandatory)	Negative - used to create new concepts and positive - match concepts that are already in the knowledge base.
word sense rank	Negative - used to create new objects and positive - match id of the object that is already in the knowledge base.
concept word rank	Negative - used to create new objects and positive - match id of the object that is already in the knowledge base.
pos (POS tag should follow the standard)	represents a part-of-speech tag noun = '1' or 'n' adjective = '2' or 'a' verb = '3' or 'v' adverb = '4' or 'r'
description	natural language string used to describe the sense
operation	'ADD', 'REMOVE', 'UPDATE'
language	3 letter string that follows the standard ISO 639-2 codes to identify the language of the sense.
examples	the use of the case lemma in a sentence

Contents of the relation sheet

Relation - Column name	Column description
parent concept uk id - (mandatory)	it represents the concept ID of the parent node.
parent concept label	and the label associated with it. This column is not imported but makes the excel file more human-readable
child concept uk id - (mandatory)	the concept ID of the child node.
child concept label	the label associated with the child ID. This column is not imported but makes the excel file more human-readable
relation kind	type of sense relation, synset relation or concept relation
operation	'ADD', 'REMOVE', 'UPDATE'
language	3 letter string that follows the standard ISO 639-2 codes to identify the language of the parent concept label.
child language	3 letter string that follows the standard ISO 639-2 codes to identify the language of the child concept label. Only for cognate relation

Contents of the gaps sheet

This sheet enable the import of the lexical gaps in a language with the description. So, the cased word lemma will be a 'GAP'.

Gaps - Column name	Column description
Cased Word Lemma:	represents the name of the entry to import and can be expressed as a string and also namespaces are supported.
Concept UK ID:	Negative - used to create new concepts and positive - match concepts that are already in the knowledge base.
Operation:	'ADD', 'REMOVE', 'UPDATE'
Language:	3 letter string that follows the standard ISO 639-2 codes to identify the language of the parent concept label.
Literal Translation:	this field contains the literal translation of a concept that represents a lexical gap.

Provenance data

It is the common set of information for all the previously mentioned sheets in the excel import file.

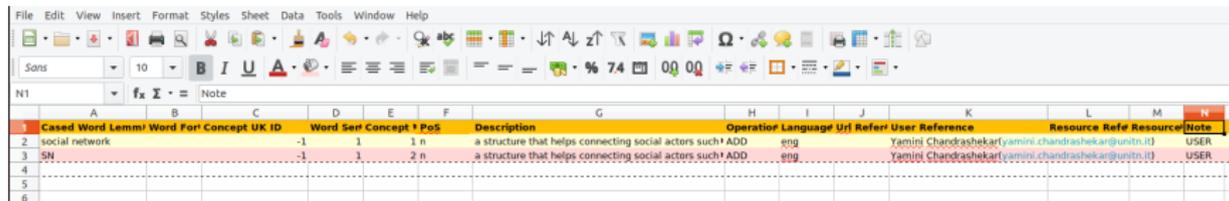
Provenance - Column name	Column description
Url Reference	external link to the resource
User Reference:	enter <i>EXACTLY</i> the format NAME SURNAME (EMAIL)
Resource Reference:	holds a link/name to an Entity which has provided the data. ResourceReference must be a unique entity name. Because of this, to better specify the provenance to add a Resource Part Reference or a Note is highly suggested.
Resource Part Reference:	If a Resource Part Reference field is specified, the Resource Reference will become the parent Resource of each Resource Part Reference naming it. Identifiers can be in the format prefix:value . Prefixes must be alphanumeric and short, in particular, they must begin with a letter and the rest of the characters may only be Unicode letters, digits, dots '.' and dashes '-'. Spaces and underscores are not allowed.
Note	'USER' when the provenance is the user. Otherwise, it can be any extra information.

Contents

- 1 Excel Importer
- 2 Excel file
- 3 Example of excel file**

Example -1

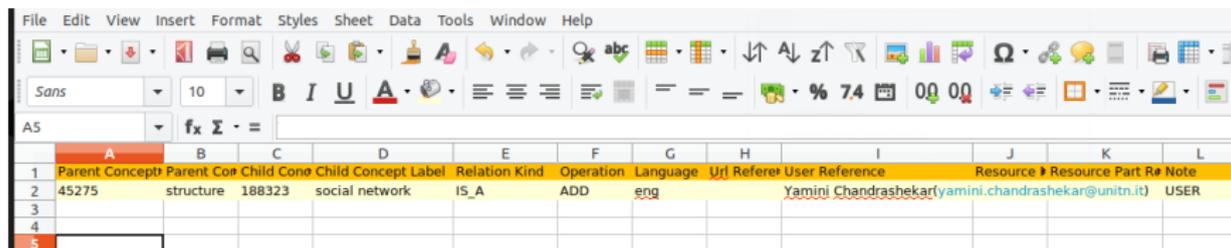
1. Add a new concept, a synset and provenance



Caseid	Word Lemm	Word For	Concept UK ID	Word Sarr	Concept P	Pos	Description	Operator	Language	Uri	Refer	User	Reference	Resource	Refe	Resource	Note
2	social network			-1	1	1 n	a structure that helps connecting social actors such	ADD	eng			Yamini Chandrashekar	[yamini.chandrashekar@unibo.it]				USER
3	SN			-1	1	2 n	a structure that helps connecting social actors such	ADD	eng			Yamini Chandrashekar	[yamini.chandrashekar@unibo.it]				USER
4																	
5																	
6																	

- *social network* is to be added to the UKC and
- it has a synset with two words – *social network* with the synset word rank 1 and *SN* with the synset word rank 2.
- In this synset, the first word is the social network, which has the word sense rank 1 and the second one is *SN*, which has the same word sense rank as the first one.
- If users cannot decide its rank they can put -1.
- Provenance of the concept as well as the synset is “Yamini Chandrashekar”, as a user.

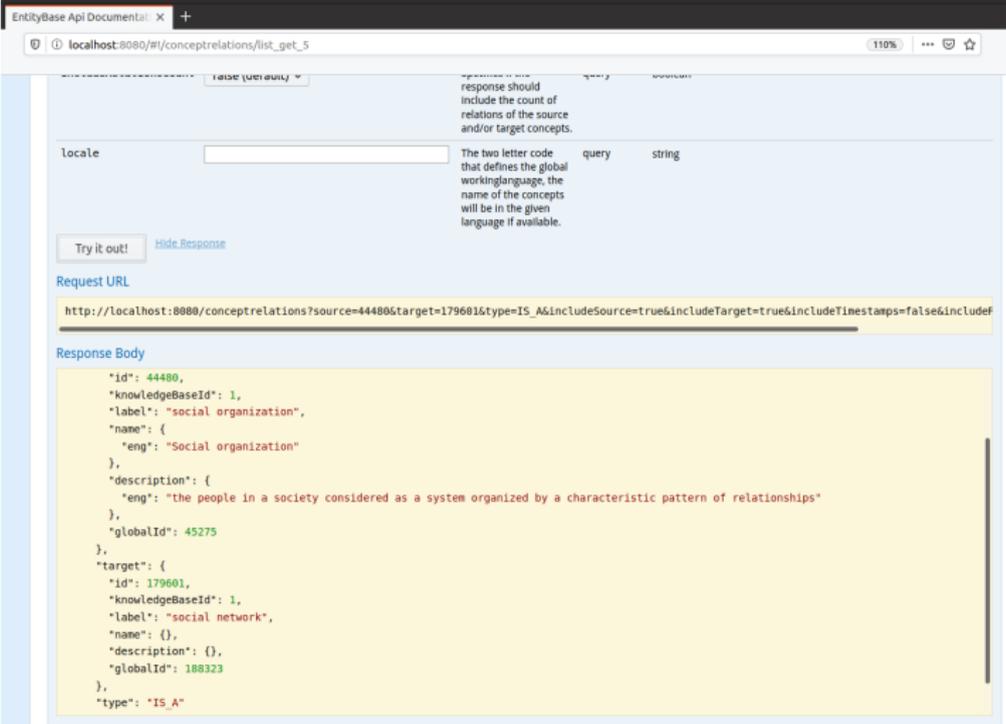
2. Add a semantic relation



	A	B	C	D	E	F	G	H	I	J	K	L	
1	Parent Concept	Parent Code	Child Code	Child Concept Label	Relation Kind	Operation	Language	Url	Referer	User Reference	Resource	Resource Part	Note
2	45275	structure	188323	social network	IS_A	ADD	eng			Yamini Chandrashekar	yamini.chandrashekar@unitn.it		USER
3													
4													
5													

- *social network* and *structure* are the nodes in the graph
- these are connected by the semantic relation *IS-A*
- NOTE: Note that the following attributes are optional: parent concept label, child concept label and language. The concept labels, if provided, help user as they increase readability.

Output



The screenshot shows a web browser window with the URL `localhost:8080/#/conceptrelations/list_get_5`. The page displays a form for a query with a `locale` field. Below the form, there is a "Try it out!" button and a "Request URL" section showing the full API endpoint: `http://localhost:8080/conceptrelations?source=44480&target=179601&type=IS_A&includeSource=true&includeTarget=true&includeTimestamps=false&includeF`. The "Response Body" section shows a JSON object representing a concept relation.

response should include the count of relations of the source and/or target concepts.

locale The two letter code that defines the global working language, the name of the concepts will be in the given language if available. query string

Try it out! [Hide Response](#)

Request URL

```
http://localhost:8080/conceptrelations?source=44480&target=179601&type=IS_A&includeSource=true&includeTarget=true&includeTimestamps=false&includeF
```

Response Body

```
{
  "id": 44480,
  "knowledgeBaseId": 1,
  "label": "social organization",
  "name": {
    "eng": "Social organization"
  },
  "description": {
    "eng": "the people in a society considered as a system organized by a characteristic pattern of relationships"
  },
  "globalId": 45275
},
"target": {
  "id": 179601,
  "knowledgeBaseId": 1,
  "label": "social network",
  "name": {},
  "description": {},
  "globalId": 188323
},
"type": "IS_A"
```

Figure: updated KB with new concept relation data for existing concept

Example - 2

Problem

Import the following data,

Concept	UKC ID	Parent Concept + UK_ID	Word sense rank
DataType	-1	Class 43482	
Boolean	97088		
AchieveAction	152		3
LoseAction	105645		2

In the above mentioned problem there are three scenarios,

- a new concept with id = -1
- a new word to already existing concept
- a new synset to already existing concept with specific word rank

Solution

- *Format the excel file as in the figure below*
- *upload the import file in the **/files** endpoint*
- *run the importer by adding the response from previous step to the **/kbimport/excel** endpoint in the UI,*

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Cased Word Lemma	Word Forms	Concept UK ID	Word Sense Rank	Concept Word Rank	PoS	Description	Operation	Language	Examples	Url Reference	User Reference	Resource Reference	Resource Part	Resource Note
2	DataType		-1	1	1	n	a data type or simply type is an attribute of data which tells the compiler or interpreter how the programmer intends to use the data	ADD	eng			Yamini Chandrashekar yamini.cf			USER
3	Boolean		97088	-1	-1	n		ADD	eng			Yamini Chandrashekar yamini.cf			USER
4	AcheiveAction		152	3	-1	v		ADD	eng			Yamini Chandrashekar yamini.cf			USER
5	LoseAction		105645	2	-1	v		ADD	eng			Yamini Chandrashekar yamini.cf			USER
6															
7															

Figure: excel input file

localhost:8080/#/kbimport/importExcel_post_2

Concepts: endpoint for the operations related to the files

files: Endpoint for the operations related to the files

GET /files/{id} Reads a file content by id

DELETE /files/{id} Deletes a file base

POST /files Creates a new file

Parameters

Parameter	Value	Description	Parameter Type	Data Type
entityBase	1	The ID of the entity base	query	long
file	Browse... sample_output.xls	The file	query	file

Try it out! [Hide Response](#)

Request URL

```
http://localhost:8080/files?entityBase=1&file=C:\3A\5Cfakepath\5Csample_output.xls
```

Response Body

```
102
```

Response Code

```
201
```

Response Headers

```
{
  "Content-Type": "application/json;charset=UTF-8"
}
```

Figure: file upload in the API

localhost:8080/#/kbimport/importExcel_post_2

110%

senserelations : Endpoint for the operations related to the sense relations [Show/Hide](#) [List Operations](#) [Expand Operations](#) [Raw](#)

attributedefinitions : Endpoint for the operations related to the attribute definitions [Show/Hide](#) [List Operations](#) [Expand Operations](#) [Raw](#)

categories : Endpoint for the operations related to the categories [Show/Hide](#) [List Operations](#) [Expand Operations](#) [Raw](#)

text : Endpoint for the operations related to NLP [Show/Hide](#) [List Operations](#) [Expand Operations](#) [Raw](#)

entitybases : Endpoint for the operations related to the entity bases [Show/Hide](#) [List Operations](#) [Expand Operations](#) [Raw](#)

conceptrelations : Endpoint for the operations related to the concept relations [Show/Hide](#) [List Operations](#) [Expand Operations](#) [Raw](#)

instances : Endpoint for the operations related to instances plus search [Show/Hide](#) [List Operations](#) [Expand Operations](#) [Raw](#)

kbimport : Endpoing for the operation related to importing knowledge core files [Show/Hide](#) [List Operations](#) [Expand Operations](#) [Raw](#)

POST [/kbimport/ukdump](#) Imports in the knowledge base the content of the ukdump file (previously uploaded using /files) using the kb import module

POST [/kbimport/conceptual](#) Imports in the knowledge base the content of the excel file (previously uploaded using /files) using the ukupdate module

POST [/kbimport/excel](#) Imports in the knowledge base the content of the excel file (previously uploaded using /files) using the ukupdate module

Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	102	The id of the file uploaded	body	long

Parameter content type: [application/json](#)

[Try it out!](#) [Hide Response](#)

Request URL

`http://localhost:8080/kbimport/excel`

Response Body

Figure: import using the API

Output - concept

The screenshot shows a web browser window displaying an API documentation page for a concept. The browser's address bar shows the URL: `localhost:8080/#/concepts/list_get_12`. The page has a light blue header with a search bar labeled "locale" and a "Try it out!" button. Below the header, there is a text block explaining the "query" parameter, which is a "string". The main content area is titled "Request URL" and shows the full URL: `http://localhost:8080/concepts?pageIndex=1&pageSize=10&knowledgeBase=1&considerTokens=false&excludeFirstToken=false&label=datatype&includeTimestamp`. Below this is the "Response Body" section, which contains a JSON array of two objects. The first object has the following structure:

```
{
  "id": 179901,
  "knowledgeBaseId": 1,
  "label": "datatype",
  "name": {
    "eng": "DataType"
  },
  "description": {
    "eng": "a data type or simply type is an attribute of data which tells the compiler or interpreter how the programmer intends to use the ..."
  },
  "globalId": 188326
}
```

. The second object is identical. At the bottom, the "Response Code" section shows "200".

Figure: updated KB with new concept

Output - sense

EntityBase Api Documental: X +

localhost:8080/#/senses/read_get_1 110%

include the count of relations of the sense.

Try it out! [Hide Response](#)

Request URL

http://localhost:8080/senses/1301123?includeWord=true&includeSynset=true&includeNestedSenses=false&includeTimestamps=false&includeRelationsCount=f

Response Body

```
{
  "id": 1301123,
  "vocabularyId": 1,
  "word": {
    "id": 776410,
    "vocabularyId": 1,
    "lemma": "loseaction",
    "forms": []
  },
  "synset": {
    "id": 97894,
    "vocabularyId": 1,
    "conceptId": 97894,
    "partOfSpeech": "VERB",
    "gloss": "fail to win"
  },
  "casedLemma": "LoseAction",
  "senseFrequency": 0,
  "wordSenseRank": 2,
  "synsetWordRank": -1
}
```

Response Code

Figure: updated KB with new sense data for existing concept

Output - provenance

The screenshot shows an API documentation interface for the EntityBase API. The selected endpoint is `PUT /provenances/{id}`, which updates a provenance. The request URL is `http://localhost:8080/provenances/187002`. The response body is a JSON object representing the updated provenance.

Parameter	Value	Description	Parameter Type	Data Type
<code>id</code>	<code>187002</code>	The id of the provenance to read	path	long

```
{
  "id": 187002,
  "source": {
    "id": 150002,
    "userDetails": {
      "id": 1401,
      "name": "yamini.chandrashekar"
    },
    "userRole": "KNOWLEDGE_IMPORTER",
    "referenceType": "USER"
  },
  "note": "USER",
  "elements": [
    {
      "subjectId": 179901,
      "type": "CONCEPT",
      "note": "USER"
    },
    {
      "subjectId": 1401,
      "type": "USER",
      "note": "USER"
    }
  ]
}
```

Figure: updated KB with new provenance



W8.L15.T15.2.3



L1-2 Importer and Excel file

Phase: 2.Tools and Tutorials