
3 DMLex Core

The DMLex Core provides data types for modelling monolingual dictionaries (called lexicographic resources in DMLex) where headwords, definitions and examples are all in one and the same language. DMLex Core gives you the tools you need to model simple dictionary entries which consist of headwords, part-of-speech labels, senses, definitions and so on.

3.1 lexicographicResource

Represents a dictionary. A lexicographic resource is a dataset which can be used, viewed and read by humans as a dictionary and – simultaneously – ingested, processed and understood by software agents as a machine-readable database. Note that the correct name of this data type in DMLex is lexicographic, not lexical, resource.

Contents

- `id` REQUIRED (exactly one). An identifier of the lexicographic resource, unique among all lexicographic resources in the implementation. If the implementation contains only one lexicographic resource then `id` MAY be understood as implicit and MAY be left unimplemented.
- `title` OPTIONAL (zero or one). Non-empty string. A human-readable title of the lexicographic resource.
- `uri` OPTIONAL (zero or one). The URI of the lexicographic resource, identifying it on the Web.
- `language` REQUIRED (exactly one). The IETF language code of the language that this lexicographic resource describes.
- `entry` OPTIONAL (zero, one or more)
- `tag` OPTIONAL (zero, one or more)

Comments

- `language` identifies the language of headwords, definitions and examples in this dictionary. DMLex is based on the assumption that all headwords in a lexicographic resource are in the same language, and that definitions and examples, if any occur in the lexicographic resource, are in that language too. The `language` child object of `lexicographicResource` informs potential users of the lexicographic resource which language that is.
- The main role of a lexicographic resource is to contain entries (`entry` objects). The other object type that can optionally occur inside a `lexicographicResource`, `tag`, is for lists of look-up values such as part-of-speech labels.

Example 1. XML

```
<lexicographicResource id="..." uri="..." language="...">
  <title>...</title>
  <entry.../>
  <tag.../>
</lexicographicResource>
```

Example 2. JSON

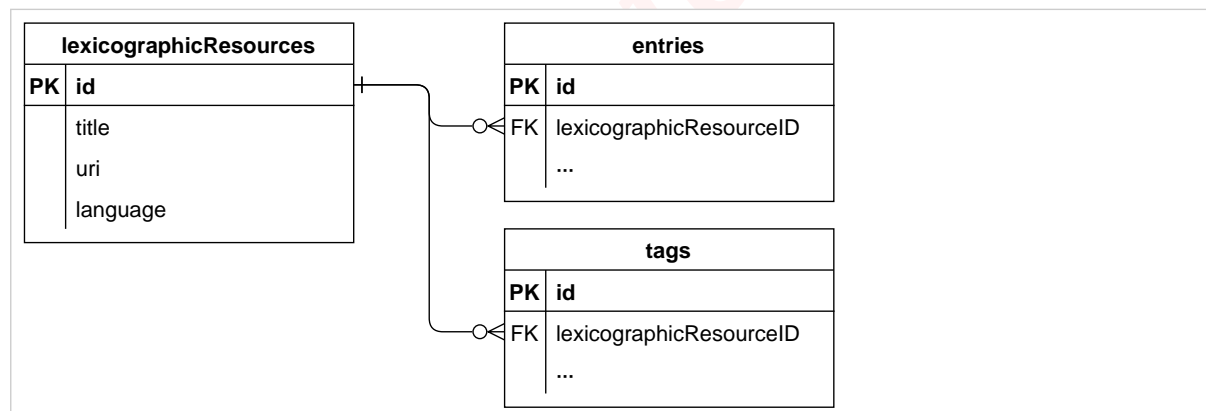
```
{
  "id": "...",
  "title": "...",
  "language": "...",
  "entries": [...],
  "tags": [...]
}
```

Example 3. RDF

```
@prefix dmlex: <http://www.oasis-open.org/to-be-confirmed/dmlex> .

<id> a dmlex:Lexicographic Resource ;
  dmlex:title "...";
  dmlex:uri "...";
  dmlex:language "...";
  dmlex:entry <entry1> , ... ;
  dmlex:tag ... .
```

Example 4. Relational database



3.2 entry

Represents a dictionary entry. An entry contains information about one headword.

Contents

- **id** REQUIRED (exactly one). An identifier of the entry, unique among all lexicographic resources in the implementation.
- **headword** REQUIRED (exactly one). Non-empty string. The entry's headword.
- **homographNumber** OPTIONAL (zero or one). The entry's homograph number, as a guide for humans to distinguish entries with the same headword.
- **partOfSpeech** OPTIONAL (zero, one or more).
- **label** OPTIONAL (zero, one or more).

- `pronunciation` OPTIONAL (zero, one or more).
- `inflectedForm` OPTIONAL (zero, one or more).
- `sense` OPTIONAL (zero, one or more).

Comments

- The headword can be a single word, a multi-word expression, or any expression in the source language which is being described by the entry.
- Entries in DMLex do not have an explicit listing order. An application can imply a listing order from a combination of the headword and the homograph number.
- DMLex Core does not have a concept of "subentry". To model subentries (ie. entries inside entries) in a lexicographic resource, you should use object types from the [Linking Module](#) for that.

Example 5. XML

```
<entry id="..." homographNumber="...">
  <headword>...</headword>
  <partOfSpeech.../>
  <label.../>
  <pronunciation.../>
  <inflectedForm.../>
  <sense.../>
</entry>
```

Example 6. JSON

```
{
  "id": "...",
  "headword": "...",
  "homographNumber": "...",
  "partsOfSpeech": [...],
  "labels": [...],
  "pronunciations": [...],
  "inflectedForms": [...],
  "senses": [...]
}
```

Example 7. RDF

```
<id> a dmlex:Entry ;
  dmlex:headword "...";
  dmlex:homographNumber ... ;
  dmlex:partOfSpeech ... ;
  dmlex:label ... ;
  dmlex:pronunciation ... ;
  dmlex:inflectedForm ... ;
  dmlex:sense ... .
```

Example 8. Relational database

