WikipediaBase Functionality

Chris Perivolaropoulos

May 9, 2016

Contents

As far as the ontology that START assumes for WikipediaBase, each infobox type corresponds to a START class and each valid infobox attribute is a START class attribute. Furthermore commands all the mentioned classes inherit from the `wikipediabase-term` START class which supports the following attributes:

- `IMAGE-DATA` : The infobox image
- `SHORT-ARTICLE` : A short version of the article, typically the first paragraph
- `URL` : The url of the article
- `COORDINATES` : Wherever it makes sense, the coordinates of the concept of the article
- `PROPER` : Whether the article refers to a proper noun (eg The Beatles, United States etc)

- `NUMBER` : `#t` if the concept of the article refers to many things, `#f` if it refres to one.

All commands and return values are encoded into s-expressions.

1 get

Given a class, object name, and typed attribute, return the value as a lisp-readable form.

Valid attribute typecodes are

- `:code` for an attribute name as in infobox wiki markup format
- and `:rendered` for an attribute name in the rendered form of the infobox.

1.1 Types

All retrun values are typed. Below is a comprehensive list of all the supported types.

1. `:HTML`

   A string suitable for rendering as paragraph-level HTML. The string must be escaped for lisp, meaning double quoted, and with double quotes and backslashes escaped with backslashes. For example:

   ```
   (get "wikipedia-sea" "Black Sea" (:code "AREA"))
   => ((:html "436,402 km2 (168,500 sq mi)"))

   (get "wikipedia-president" "Bill Clinton" (:code
      "SUCCESSOR"))
   => ((:html "George W. Bush"))

   (get "wikipedia-president" "Bill Clinton" (:
      rendered "Succeeded by"))
   => ((:html "George W. Bush"))
   ```

2. `:YYYYMMDD`

   Parsed dates are represented in the format `[-]<4 digit year><2 digit month><2 digit day>`. Unparsable dates are represented as `:html` types

```
(get "wikibase-person" "Barack Obama" (:ID "BIRTH
   -DATE"))
=> ((:yyyymmdd 19610804))

(get "wikibase-person" "Julius Caesar" (:ID "
   BIRTH-DATE"))
=> ((:YYYYMMDD -1000713))
```

3. `:CALCULATED`

   The type of calculated properties based on characteristics of the article, e.g., *GENDER* and *NUMBER*. See below under Special Attributes for a complete list of calculated attributes.

4. `:CODE`

   Deprecated, old synonym for `:HTML`.

5. `:STRING`

   Deprecated, old synonym for `:HTML`.

1.2 Special Attributes

Besides the attributes that are fetched as attributes of the infobox, the rest of the available attributes are special in that they are calculated from the contents of the article. They are also special in that they are hardocoded, ie the value of the attribute is calculated, not the attribute itself. These attributes should be specific to `wikibase-term`, `wikibase-person`, and `wikipedia-paragraphs`.

1. `SHORT-ARTICLE`, `wikibase-term`

   The first paragraph of the article, or if the first paragraph is shorter than 350 characters, then the value of `short-article` is the the first paragraphs such that the sum of the rendered characters is at least 350.

2. `URL`, `wikibase-term`

   The URL of the article as `((:url URL))`

3. `IMAGE-DATA`, `wikibase-term`

   A list of URLs for images in the article content (excludes images that are in the page but outside of the article content). The "best"

image should be the first URL in the list; if there is a picture at the top of the infobox, this is considered to be the best image, or otherwise the first image that appears anywhere in the article. If there is no caption, the caption value should be omitted, e.g., `((0 "Harimau_Harimau_cover.jpg"))` rather than `((0 "Harimau_Harimau_cover.jpg" ""))`.

4. `COORDINATES`, `wikibase-term`

   Computed from latitude and longitude attributes given in the article header or, if none can be found, the infobox. The value is a list of the latitude and longitude, e.g., ((`:coordinates` latitude longitude))

Figure 1: An example of coordinates in the header

5. `BIRTH-DATE`, `wikibase-person`

   Fetched from the infobox, or, if it is not found, from the artle, or, if it is not found, the category information of the article. Always relies on the first date of birth found, matching one of several supported formats. If this attribute has a value, then the object is considered to be a person with respect to the GENDER attribute (see below). The value can be a parsed or unparsed date. Parsed dates are represented as numbers, using YYYYMMDD format with negative numbers representing B.C. dates. Unparsed dates are strings.

6. `DEATH-DATE`, `wikibase-person`

   Fetched similarly to BIRTH-DATE. Returns the same value types as BIRTH-DATE, except if the person is still alive, throws an error with the reply "Currently alive".

7. `GENDER`, `wikibase-person`

   Computed from the page content based on heuristics such as the number of times that masculine vs. feminine pronouns appear. Valid values are `:masculine` and `:feminine`.

8. `NUMBER`, `wikibase-term`

Computed from the page content based on heuristics such as number of times the page's title appears plural. Valid for all objects. Returns `#t` if many, `#f` if one.

(a) `PROPER`, `wikibase-term`
Computed from the page content based on heuristics such as number of times the page's title appears capitalized when not at the start of a sentence. Valid for all objects. Returns `#t` if proper and `#f` if not.

2 get-classes

Given an object name, return a list of all classes to which the object belongs, with classes represented as lisp-readable strings. Class names are conventionally given in lower case, but this is not an absolute requirement. E.g.,

```
(get-classes "Cardinal (bird)")
=> ("wikibase-term" "wikipedia-paragraphs" "wikipedia
   -taxobox")

(get-classes "Hillary Rodham Clinton")
=> ("wikibase-term" "wikipedia-paragraphs" "wikibase-
   person" "wikipedia-officeholder" "wikipedia-person
   ")
```

3 get-attributes

Given a class name, return a list of all attributes that the class implements (that is, all variables that the infobox implements), as lisp-readable strings. Also sometimes given is the human-readable rendering of the attribute and/or the value typecode for the attribute. Attribute names are conventionally given in upper case, but this is not an absolute requirement. E.g.,

```
(get-attributes "wikipedia-officeholder" "Barack
   Obama")
=> ((:CODE "TERM_END3" :VALUE :YYYYMMDD) ...)
```

4 ort-symbols

`sort-symbols` takes any number of symbols and sorts them into subsets by the length of the associated article. E.g.,

```
(sort-symbols  "Obama (surname)" "Barack Obama")
=> (("Barack Obama") ("Obama (surname)"))
```

5 sort-symbols-named

`sort-symbols-named` takes a synonym and any number of symbols and sorts the symbols into subsets; if any symbol name is the same as the synonym, it and its subset are sorted to the front. (This should be a case insensitive match, but is it? And again, what's with the subsets?) E.g.

```
(sort-symbols-named "cake" "Cake (TV series)" "Cake (
   firework)" "Cake (film)" "Cake (drug)"
"Cake" "Cake (band)" "Cake (advertisement)" "The Cake
   ")
=> (("Cake") ("Cake (band)") ("Cake (advertisement)")
    ("Cake (TV series)")
("The Cake") ("Cake (film)") ("Cake (firework)") ("
   Cake (drug)"))
```