Practical Assignment 2

Your Library

Web Semântica
Teacher: Hélder Zagalo
31/05/2023

Inês Leite, 92928
Renan Ferreira, 93168
Tiago Coelho, 98385

Table of Contents

01 Introduction

02 New Components

03 Completion of the data set

04 RDFa & Microformats

05 Demo

06 Conclusion

01

Introduction

Introduction

Goal

Improve the book information system from the 1st project with the goal of exposing and managing all the information in the system, including the ad-hoc creation of new inferences.

Vision

Obtain an information system capable of serving as a book display and able to manage the read books, also taking into account their various information.

02

New Components

New Components

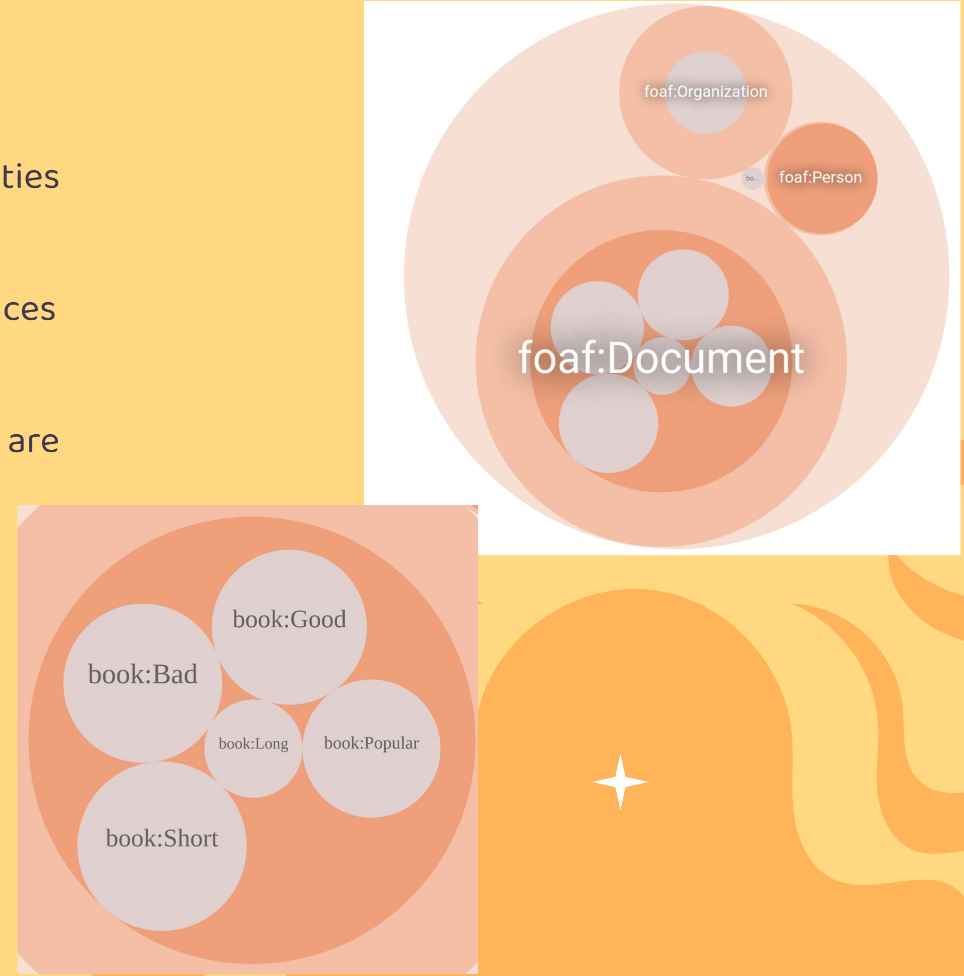
- Use of SparqlWrapper to create new inferences to be persisted.
- We obtained additional Information.
    - Book Cover Image
    - Author Image
- Use of Protege to create final information file.

Ontology

Signals classes and hierarchies between entities and their properties.

Also, we created subclasses to make inferences later:

- Short, Long, Good, Bad and Popular are subclasses of Book



Inferences

The new Semantic System uses inferences to create new classification categories to the books. Prior to request any resource, new triples for this classification are created.

Obs: We were not able to obtain inferences in runtime to associated queries with it.

Categories:

- Good Book
- Bad Book
- Popular Book
- Short Book
- Long Book

03

Completion of the data set

Completion of the data set

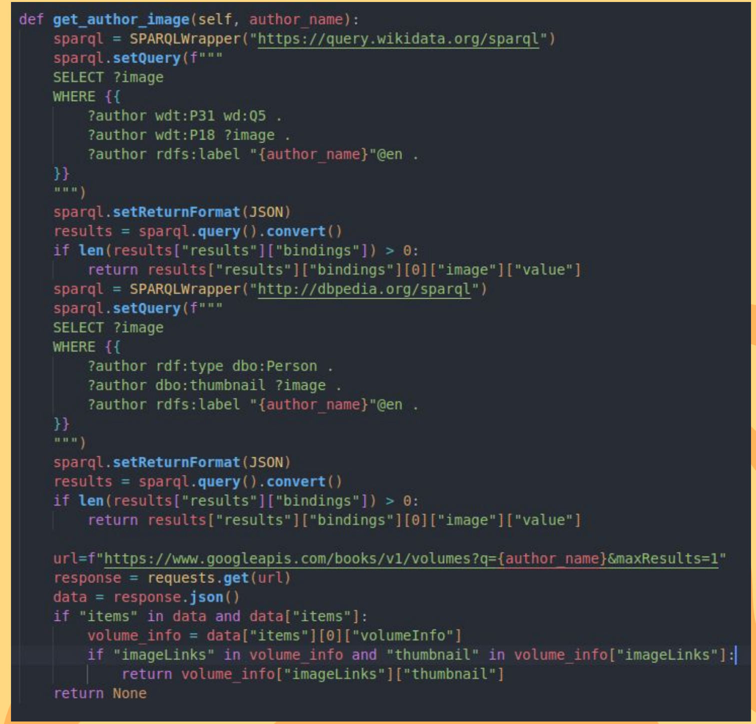
New data:

- Book cover image;
- Author image;

Data taken from:

- Wikidata;
- DBpedia;
- GoogleAPI by url;

We also explored other data (gender, appropriate age and description of the author) that we could take advantage of, but often only existed for a few books, so we chose not to add it to the site.



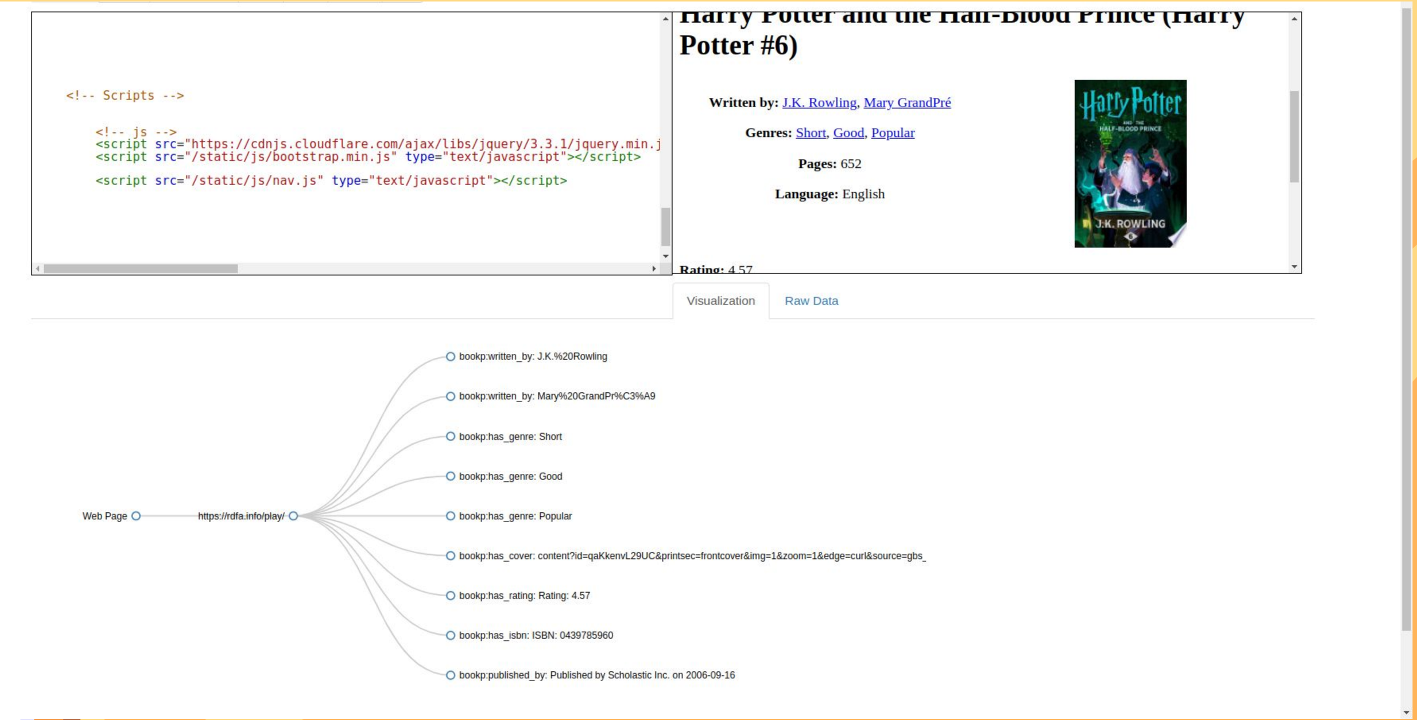
```python
def get_author_image(self, author_name):
    sparql = SPARQLWrapper("https://query.wikidata.org/sparql")
    sparql.setQuery(f"""
    SELECT ?image
    WHERE {{
        ?author wdt:P31 wd:Q5 .
        ?author wdt:P18 ?image .
        ?author rdfs:label "{author_name}"@en .
    }}
    """)
    sparql.setReturnFormat(JSON)
    results = sparql.query().convert()
    if len(results["results"]["bindings"]) > 0:
        return results["results"]["bindings"][0]["image"]["value"]
    sparql = SPARQLWrapper("http://dbpedia.org/sparql")
    sparql.setQuery(f"""
    SELECT ?image
    WHERE {{
        ?author rdf:type dbo:Person .
        ?author dbo:thumbnail ?image .
        ?author rdfs:label "{author_name}"@en .
    }}
    """)
    sparql.setReturnFormat(JSON)
    results = sparql.query().convert()
    if len(results["results"]["bindings"]) > 0:
        return results["results"]["bindings"][0]["image"]["value"]

    url=f"https://www.googleapis.com/books/v1/volumes?q={author_name}&maxResults=1"
    response = requests.get(url)
    data = response.json()
    if "items" in data and data["items"]:
        volume_info = data["items"][0]["volumeInfo"]
        if "imageLinks" in volume_info and "thumbnail" in volume_info["imageLinks"]:
            return volume_info["imageLinks"]["thumbnail"]
    return None
```

04

RDFa & Microformats

RDFa

Added RDFa attributes (property, resource, etc) to the Book page and Author page.



```
<!-- Scripts -->

    <!-- js -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.j
    <script src="/static/js/bootstrap.min.js" type="text/javascript"></script>

    <script src="/static/js/nav.js" type="text/javascript"></script>
```

Harry Potter and the Half-Blood Prince (Harry Potter #6)

Written by: J.K. Rowling, Mary GrandPré

Genres: Short, Good, Popular

Pages: 652

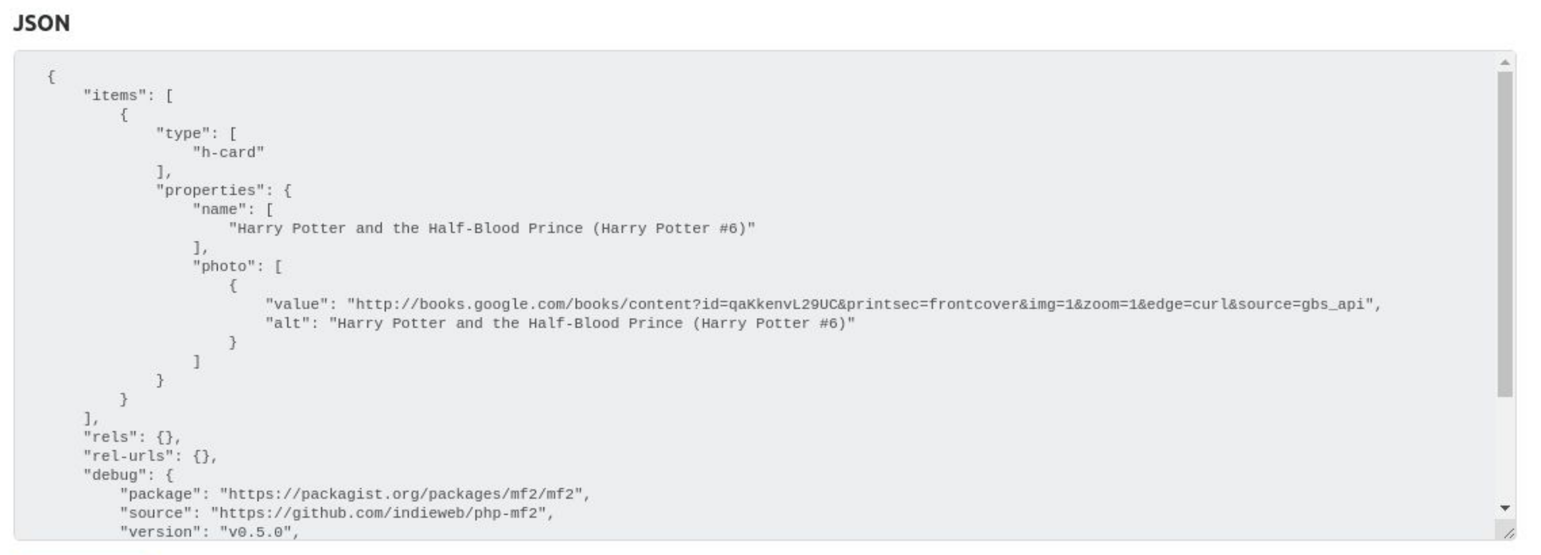
Language: English

Rating: 4.57

Web Page
https://rdfa.info/play/
bookp:written_by: J.K.%20Rowling
bookp:written_by: Mary%20GrandPr%C3%A9
bookp:has_genre: Short
bookp:has_genre: Good
bookp:has_genre: Popular
bookp:has_cover: content?id=qaKkenvL29UC&printsec=frontcover&img=1&zoom=1&edge=curl&source=gbs_
bookp:has_rating: Rating: 4.57
bookp:has_isbn: ISBN: 0439785960
bookp:published_by: Published by Scholastic Inc. on 2006-09-16

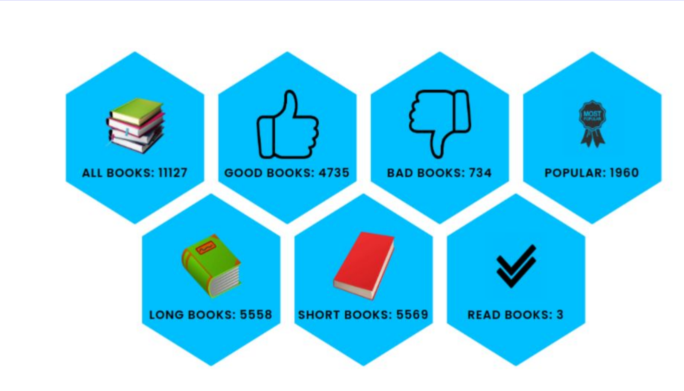
Microformats

Added to Book and Author pages to gather information about the name and photo.



JSON

```
{
    "items": [
        {
            "type": [
                "h-card"
            ],
            "properties": {
                "name": [
                    "Harry Potter and the Half-Blood Prince (Harry Potter #6)"
                ],
                "photo": [
                    {
                        "value": "http://books.google.com/books/content?id=qaKkenvL29UC&printsec=frontcover&img=1&zoom=1&edge=curl&source=gbs_api",
                        "alt": "Harry Potter and the Half-Blood Prince (Harry Potter #6)"
                    }
                ]
            }
        }
    ],
    "rels": {},
    "rel-urls": {},
    "debug": {
        "package": "https://packagist.org/packages/mf2/mf2",
        "source": "https://github.com/indieweb/php-mf2",
        "version": "v0.5.0",
```



ALL BOOKS: 11127

GOOD BOOKS: 4735

BAD BOOKS: 734

POPULAR: 1960

LONG BOOKS: 5558

SHORT BOOKS: 5569

READ BOOKS: 3

Demo 05

Conclusion

06

Conclusion

With this second project, we were able to learn more about the process of inferences and how wikidata can be a very useful addition to a semantic system.

Thus, we developed a functional user library, able to show several thousand books with filters, sorts and search functions implemented, including create new information ad-hoc, similar to inference.

Regarding future work, we would like to implement authentication and authorization and an admin view to do the CRUD operations.

Also, we would like to have a more robust ontology in order to be have more information for the user.

Thanks!

CREDITS: This presentation template was created by **Slidesgo,** including icons by **Flaticon,** and infographics & images by **Freepik**

Dataset Flow

01

Dataset

Finding the right dataset for the application objective

02

Conversion

Convert CSV file to triple-store, editing data.

03

Create Ontology

Add ontology and triples to **Protege** to create final .ttl file.

04

GraphDB

Import the .ttl file into GraphDB

Data

Origin: Kaggle - CSV File
About 11 000 books from GoodReads.com

Dataset information:

- Book title and authors;
- Rating and number of reviews;
- ISBN;
- Number of pages and book language;

Later we added other information such as categories and a variable for the user to mark the book as read.