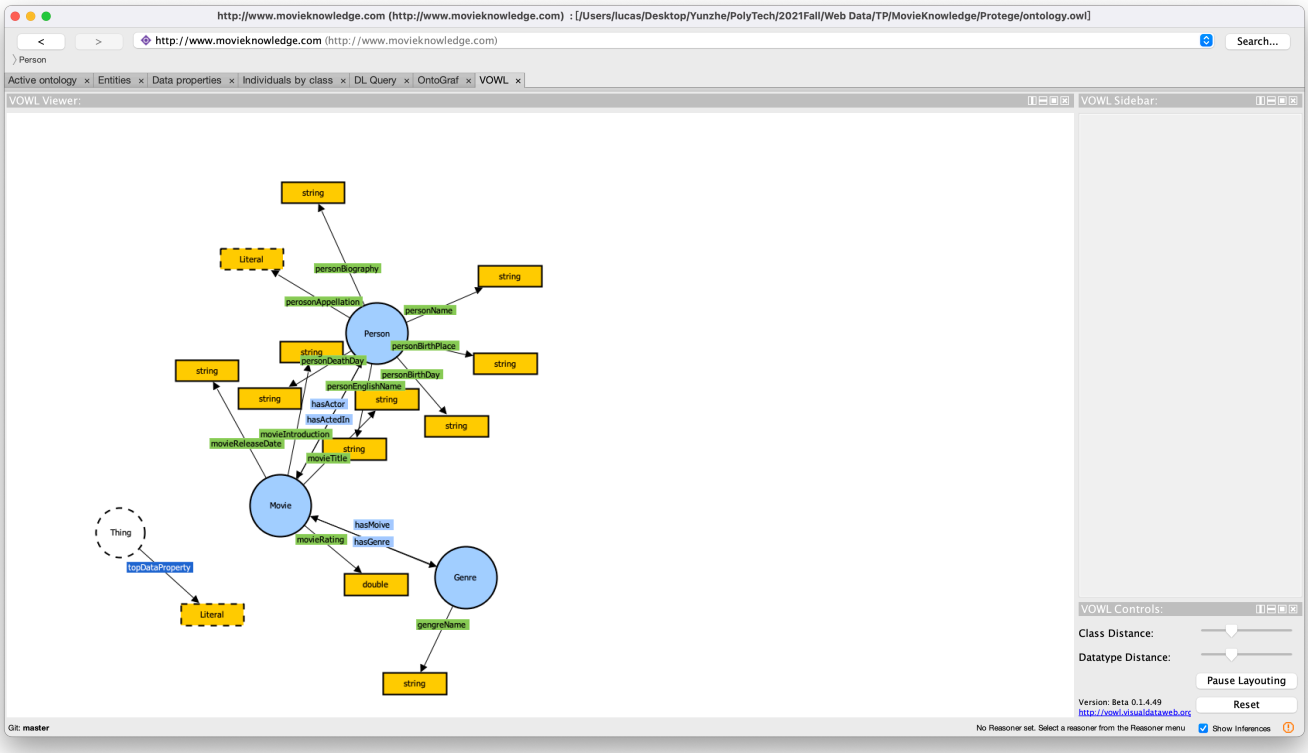
File Strucure

Protege

This file contains the `OWL` file for our project, it contains the entity, property and relationship defined.



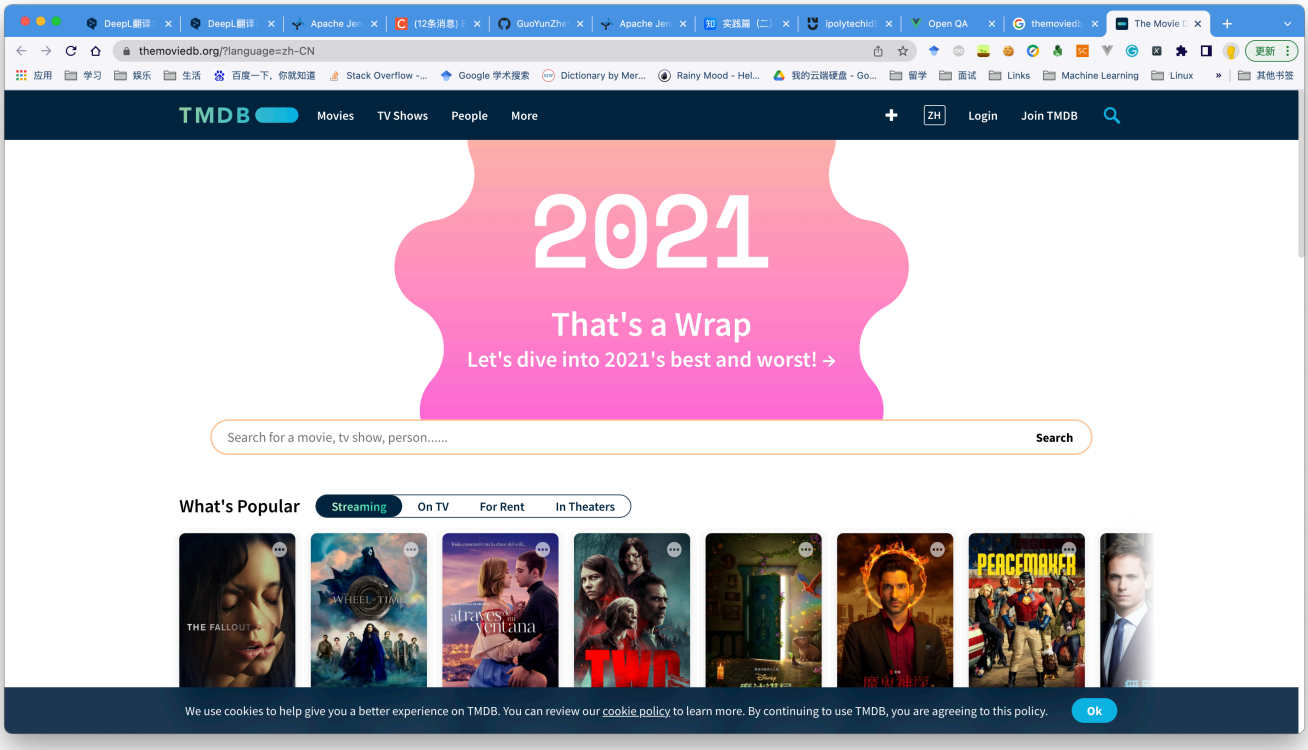
BackEnd

script

This directory contains the scripts and config for uwsgi.

Crawler.py

This file will craw the actor/movie information from [the movie db(TMDB)]() and insert into local database.



Data

MovieSQLData

This directory contains the SQL Insert data which is crawled by our `crawler.py`.

MovieKnowledge_*.json

This two json files contain the person name and movie name, they are used to train a **Entity Classifier** by `spacy`.

BackEnd

This directory contains the `django` framework we used, and the view and controller file.

FontEnd

We used `vue.js` as the frontend template.

TDB

This directory contains the `TDB` data for fuseki.

apache-jena-fuseki-4.3.2

Fuseki is a SPARQL server. It provides REST-style SPARQL HTTP updates, SPARQL queries and SPARQL updates over HTTP using the SPARQL protocol.

```
# apache-jena-fuseki-4.3.2/run/configuration/fuseki_conf.ttl

@prefix :      <http://base/#> .
@prefix tdb:   <http://jena.hpl.hp.com/2008/tdb#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ja:    <http://jena.hpl.hp.com/2005/11/Assembler#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix fuseki: <http://jena.apache.org/fuseki#> .


:service1        a                 fuseki:Service ;
fuseki:dataset                     <#dataset> ;
fuseki:name                        "MovieKnowledge" ;
fuseki:serviceQuery                "query" , "sparql" ;
fuseki:serviceReadGraphStore       "get" ;
fuseki:serviceReadWriteGraphStore "data" ;
fuseki:serviceUpdate               "update" ;
fuseki:serviceUpload               "upload" .


<#dataset> rdf:type ja:RDFDataset ;
    ja:defaultGraph <#model_inf> ;
    .

<#model_inf> a ja:InfModel ;
    ja:MemoryModel <#tdbGraph> ;

    #本体文件的路径
    ja:content [ja:externalContent <file:///Applications/apache-jena-fuseki-
4.3.2/run/databases/ontology.ttl> ] ;

    #启用OWL推理机
    ja:reasoner [ja:reasonerURL <http://jena.hpl.hp.com/2003/OWLFBRuleReasoner>] .

<#tdbGraph> rdf:type tdb:GraphTDB ;
    tdb:dataset <#tdbDataset> ;
    .

<#tdbDataset> rdf:type tdb:DatasetTDB ;
    tdb:location "/Users/lucas/Desktop/Yunzhe/PolyTech/2021Fall/Web
Data/TP/MovieKnowledge/TDB" ;
    ja:context[ ja:cxName "arqTimeout"; ja:cxtValue "1000"]
    .
```

d2rq-0.8.1

`d2rq` is used to get `rdf` data from our `sql` data automaticly.

How to Run

apache-jena-fuseki-4.3.2

Make sure:

The path to `ontology` and `TDB` is right.

```
    ja:content [ja:externalContent <file:///Applications/apache-jena-fuseki-
4.3.2/run/databases/ontology.ttl> ] ;
    ...
    <#tdbDataset> rdf:type tdb:DatasetTDB ;
    tdb:location "/Users/lucas/Desktop/Yunzhe/PolyTech/2021Fall/Web
Data/TP/MovieKnowledge/TDB" ;
    ja:context[ ja:cxName "arqTimeout"; ja:cxtValue "1000"]
    .
```

BackEnd

Change path to : `/MovieKnowledge/BackEnd`

```
python3 manage.py makemigrations
python3 manage.py migrate
python3 manage.py runserver
```

Make sure `BackEnd/BackEnd/settings.py`

```
EXTERN_DATA = [
    {
        "Path":
"/Users/lucas/Projects/Pycharm/MovieKnowledge/Data/MovieKnowledge_person.json",
        "Label": "Person",
        "Pattern": "person_english_name"
    },
    {
        "Path":
"/Users/lucas/Projects/Pycharm/MovieKnowledge/Data/MovieKnowledge_movie.json",
        "Label": "Film",
        "Pattern": "movie_title"
    }
]
```

The path to json data is correct.

FrontEnd

In `src/config.json`:

```
{
    "API_HOST": "http://127.0.0.1:8000/api" # Should be the same with backend
}
```

```
npm install
npm run serve
```