

Official city desk crawler

Mark Sobolev

28. July 2019

Project description

The aim of the project is to create a crawler, which downloads documents published on official boards of individual cities of the Czech Republic.

Proposed solution

In the first step, the program obtains links to official city pages using the list of Czech cities https://cs.wikipedia.org/wiki/Seznam_m%C4%9Bst_v_%C4%8Cesku. Subsequently, the links to the official boards of individual cities are extracted. In the last step, the program downloads files from individual official boards.

Implementation

The project is programmed in Python3 using the Scrapy framework, which simplifies the documents extraction.

2 scripts are used to download documents:

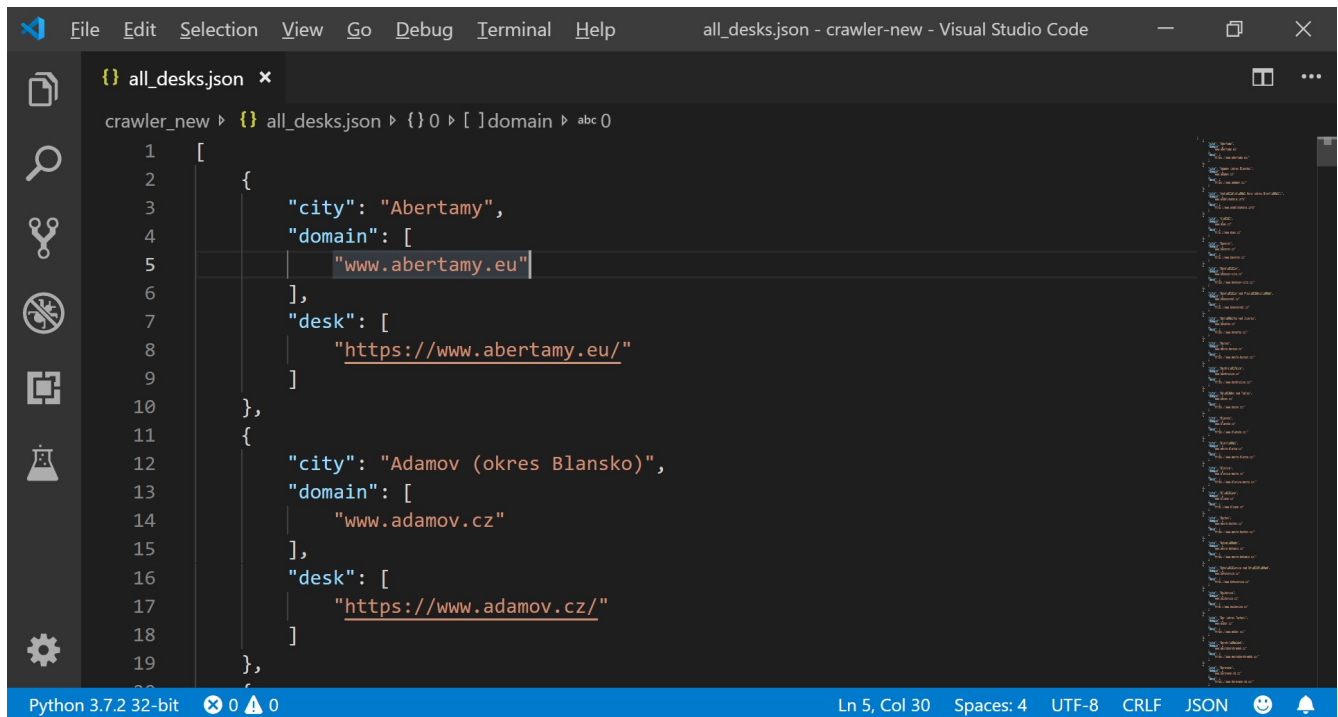
- *download_desks.py* extracts the municipal websites URLs. Extracted URLs are stored in JSON format. For each web page the file contains the name of the city, its website URL and domain.
- *download_documents.py* downloads documents from individual official websites using extracted URLs. Documents are divided into folders - one for each city.

Downloaded documents should be prepared for categorization - it is necessary to eliminate duplicates and assign files names corresponding to the content of the documents.

2 scripts are used to prepare documents:

- *remove_duplicates.sh* removes documents which appear in folder more than once (Website may have several links to the same document, so the same document may be downloaded more than once)

- *extract_names.py* extracts the text from the first line of the pdf file and uses it as the new name for the file.



```
{} all_desks.json x
crawler_new ▸ {} all_desks.json ▸ {} 0 ▸ [ ] domain ▸ abc 0
1  [
2    {
3      "city": "Abertamy",
4      "domain": [
5        "www.abertamy.eu"
6      ],
7      "desk": [
8        "https://www.abertamy.eu/"
9      ]
10   },
11   {
12     "city": "Adamov (okres Blansko)",
13     "domain": [
14       "www.adamov.cz"
15     ],
16     "desk": [
17       "https://www.adamov.cz/"
18     ]
19   },
20 ]
```

Python 3.7.2 32-bit 0 0 Ln 5, Col 30 Spaces: 4 UTF-8 CRLF JSON

Fig 1. List of official municipal webpages in JSON format

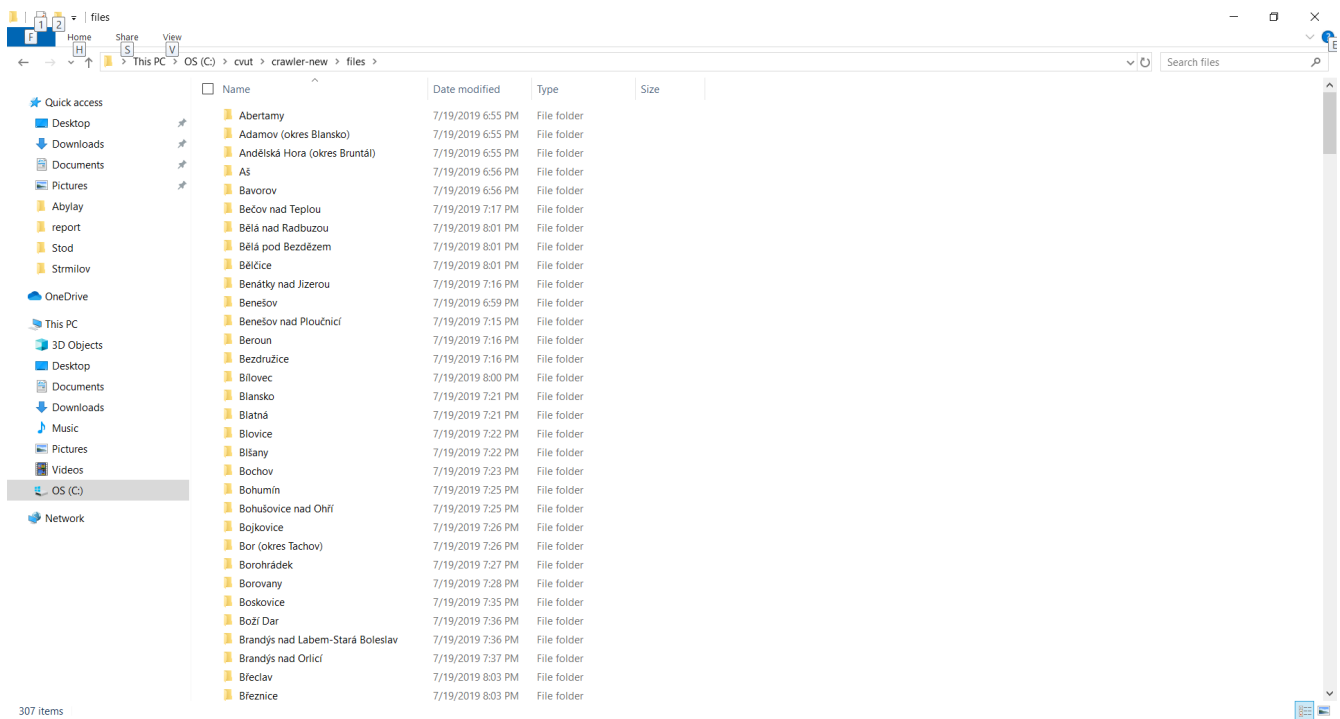


Fig 2. Folders with downloaded documents

How to use scripts

To work with scripts for downloading documents, one needs to install the Scrapy tool, which can be downloaded using the *pip install scrapy* command.

First run *python download_desks.py* to extract the municipal websites URLs. Then run *python download_documents.py* with appropriate arguments to download documents from individual official websites using extracted URLs.

Examples of valid argument combinations:

python download_documents.py --file all_desks.json --cities adamov bavorov
downloads documents for Adamov and Bavorov

python download_documents.py --file all_desks.json --begin adamov --end bavorov
downloads documents for cities starting with Adamov and ending with Bavorov. Cities in the list should be sorted lexicographically.

python download_documents.py --file all_desks.json --begin adamov --count 5

downloads documents for 5 cities starting with Adamov.

python download_documents.py --file all_desks.json --begin adamov downloads documents for all cities in the list starting with Adamov.

python download_documents.py --file all_desks.json downloads documents for all cities in the list.

Arguments for <i>download_documents.py</i>			
Argument	Description	Example	Note
<i>--file</i>	Name of the file with the URLs which is used for downloading documents.	<i>--file all_desks.json</i>	Required. Should be the first argument.
<i>--cities</i>	Names of the cities to download documents for.	<i>--cities adamov bavorov</i>	Optional. Can not be combined with <i>--begin</i> . Case insensitive.
<i>--begin</i>	Name of the city to start with.	<i>--begin adamov</i>	Optional. Can not be combined with <i>--cities</i> . Case insensitive. If <i>--count and --end were not used, documents will be downloaded for</i>
<i>--count</i>	Number of cities to download documents for.	<i>--count 5</i>	Optional. Must be preceded by <i>--begin</i> . Case insensitive.
<i>--end</i>	Name of the last city to download documents for.	<i>--end bavorov</i>	Optional. Must be preceded by <i>--begin</i> . Case insensitive.

Then you can run *remove_duplicates.sh* and *extract_names.py* to prepare documents for categorization. Both scripts should be placed in one folder with folders containing downloaded documents. *remove_duplicates.sh* removes duplicates from all folders in the same directory and *extract_names.py* extracts names from all folders in the same directory.

extract_names.py requires *Pillow* (Python Imaging Library), *pdf2* (wrapper around the pdftoppm and pdftocairo command line tools to convert PDF to a PIL Image list) and *pytesseract* (python wrapper for Google's Tesseract-OCR).

Results

During testing, the task was to download documents for 307 cities. As a result, documents for 291 cities were downloaded (95% success rate).

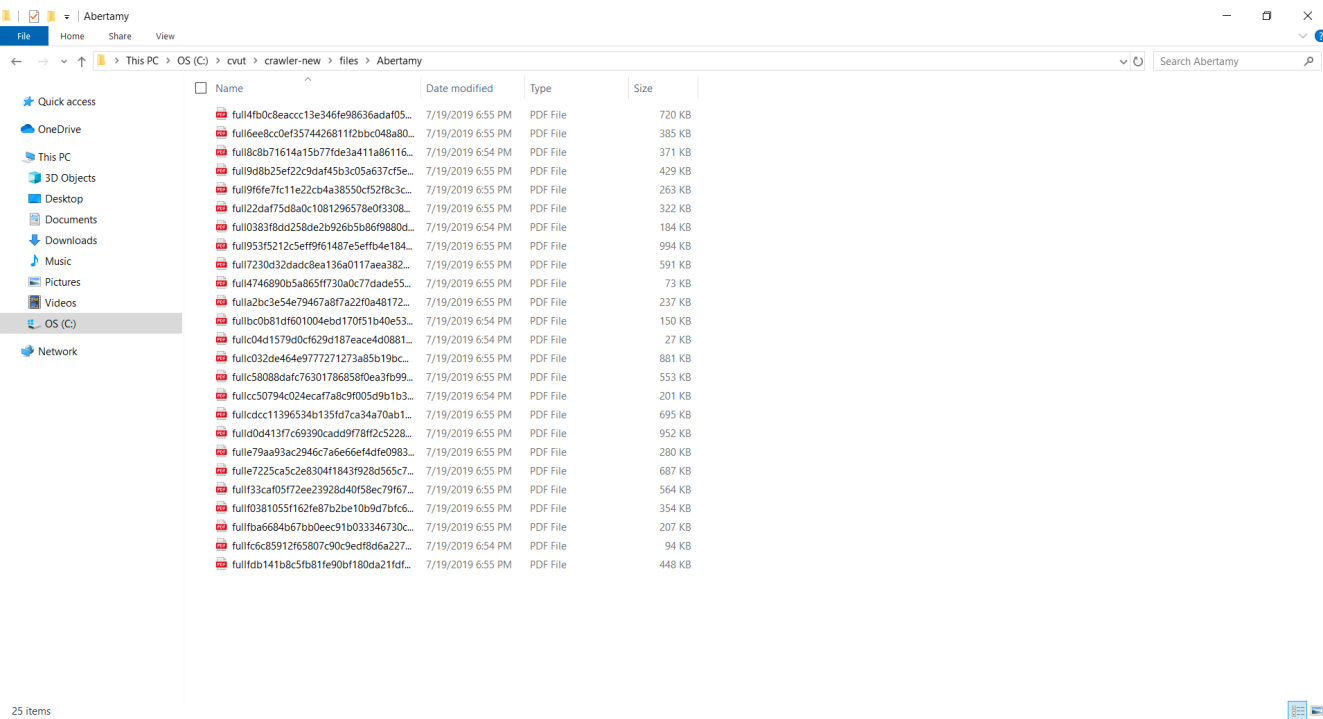


Fig 3. Downloaded documents

Downloaded documents were then renamed using *extract_names.py*. As seen on Figure 4 not all the names extracted correspond to the actual titles of the documents.

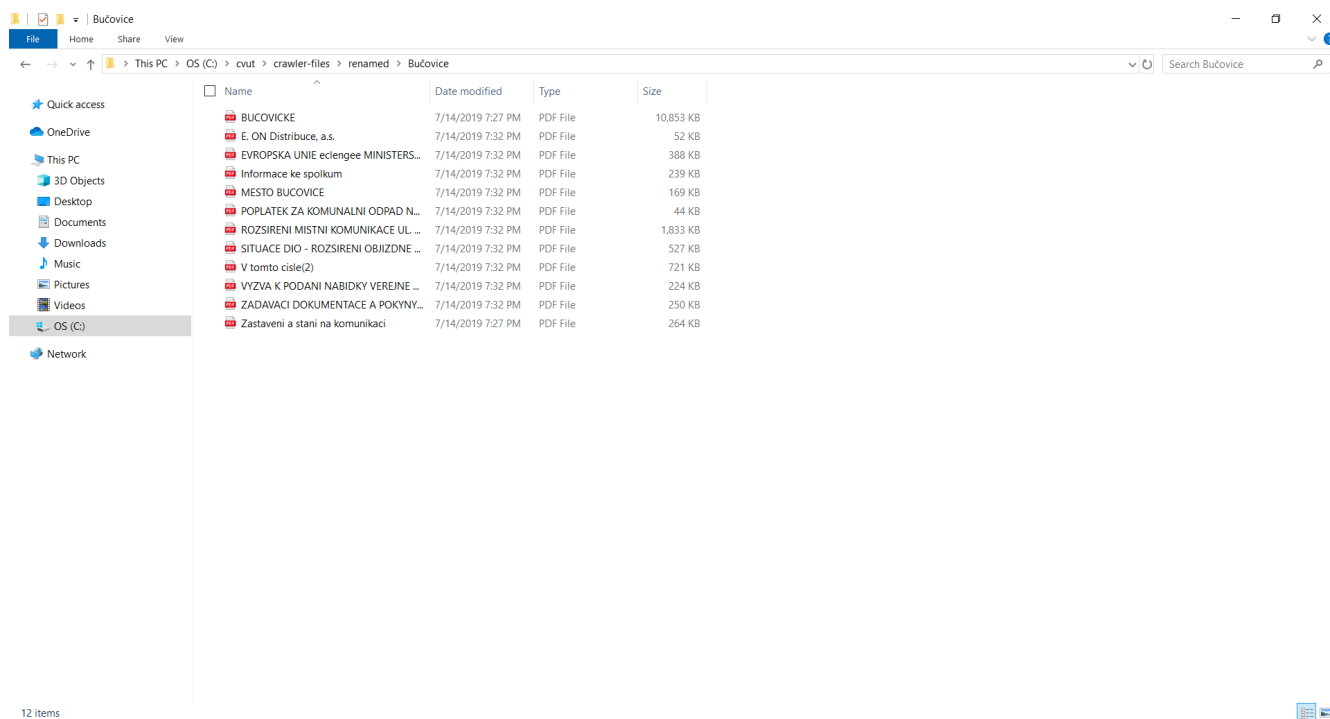


Fig 4. Renamed documents

Conclusion

Byly staženy dokumenty z úředních desek pro prvních 10 měst. Dalším cílem je rozšíření crawleru pro manipulaci s více různými typy úředních desek a kategorizace dokumentu.