

IleToMieszkanie?

Antoni Szczepanik

22-01-2021

IleToMieszkanie?

The general idea behind this project is as follows:

1. Rent & sale offers are scraped and valuated.
2. Offer price is compared with valuations.
3. You can filter & preview most interesting offers on map.



Figure 1: Screenshot of the page

Architecture Overview

The project is structured as 3 components.

- Scraper which scrapes & valuates the offers.
- Server which allows to query the data.
- Client, just to present the results on a map.

The interaction between the components is presented in the chart below.

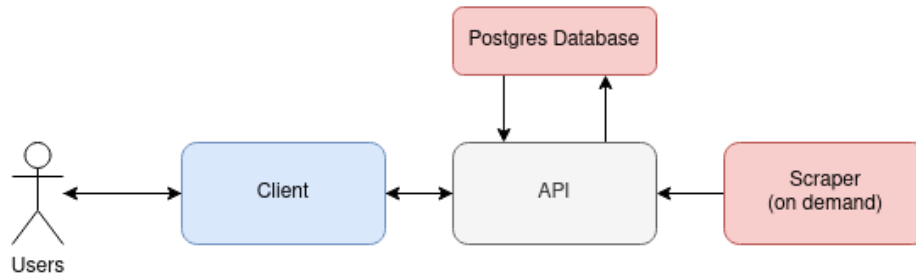


Figure 2: Architecture overview

Scraper

Scraper works as follows:

1. It scrapes offers.
2. Data is cleaned, outliers are filtered, new features are added, etc
3. Newest of previously trained models is applied.
4. Offers are uploaded through API.

Additionally one can run one of on demand tasks:

- Model training, which will read all the data scraped and train a new model based on that. This could be done through Jupyter Notebook to allow supervision of trained model parameters.
- “Coords map” generation. This is a table of neighborhoods with mean prices & additional features assigned. It is generated based on whole data and it is used later when training & applying the model.

Model

The model is Random Forest Regressor. It's trained on all historically scraped data, so the offer price are predicted. Currently the error is around 14% nRMSE for both sale and rent offers.

Webserver

Webservers role is to allow querying all scraped & valuated offers. It offers REST API (with Postgres in the backend) allowing to add and retrieve offers. The endpoint available are presented below.

GET	/api/v1/offers/	Read Offers
POST	/api/v1/offers/	Create Offer
GET	/api/v1/offers/{offer_id}	Read Offer
GET	/api/v1/predict/	Get Prediction

Figure 3: Available endpoints

More complete Swagger documentation is available at `{server_url}{server_port}/docs`.

Client

The client has been written in vanilla Javascript. Leaflet was used for mapping capabilities. You can filter & preview most interesting ones on map. It is possible to query the data by following parameters:

- offer type (both rent & sale offers are supported)
- date when offer was added
- offer price
- size
- difference between valuation and offer price

How to run it?

All components in this project are packaged as docker containers. In order to run it you have to have **docker** and **docker-compose** installed.

The first step is to run MinIO container - it serves as static data storage. It mimics S3 key-value storage, which is used in production.

```
cd scraper && ./setup_minio.sh && cd ..
```

After that run server and database containers. This is where scraped data will be finally uploaded.

```
cd server && docker-compose up -d && cd ..
```

Now you can scrape & process first offers.

```
./run.sh scrape sale && ./run.sh process sale
```

And you just scraped and valued first batch of offers :)

`./run.sh` is used to run all on-demand task in this project. To learn more try:

```
./run.sh --help
```