# Scraper task

## Task description

`In this task you are required to create a scraper for the site which will get the information needed and store it in the database. After that, you should create an API interface for getting the info using http protocol.`

### VCS requirements

- use git
- create github repo for review
- commit as often as possible. But be sure to only commit working versions of apps.
- write meaningful commit messages. Be consistent in commit message format. It's a holy-war point, but we'd recommend to stick to the following format: "Added new features to abc module."
  - Start with a capital letter.
  - Use the verb in past tense, so you describe what you did.
  - Put a period in the end.
- Create feature branches or, at least, a branch per task phase.
- To make the checking process easier and closer to real world, create pull requests from phase branches to the main branch.

### Scraping requirements

- site with data: OMA's catalog https://www.oma.by/catalog/
- get categories, subcategories, products into db
- for products save name, price, description, charachteristics, similar products, product image link and if product is a hit or not
- use beautifulsoup (https://www.crummy.com/software/BeautifulSoup/bs4/doc/) and requests (https://3.python-requests.org/) packages

## Database requirements

- use postgresql for storage
- use sqlalchemy or peewee as orm
- create queries using pure sql for db creation and selecting products with categories
  - get product with similar products,
  - get products with categories
  - get categories with count of products for each category

## API requirements

- use flask (http://flask.pocoo.org/) or tornado (https://www.tornadoweb.org/en/stable/) for API
- create REST API https://www.restapitutorial.com/lessons/whatisrest.html
- implement CRUD operations on products and categories
- implement soft delete feature
- implement both individual and list endpoints for categories, subcategories and products
- implement paging for list endpoints
- implement searching feature for products (search string, price range, category, subcategory)
- use pure queries implemented for db for crud API operations