

# Web Scraper + API

## Take-Home Challenge

**Background:** Drugbank is an online curated knowledgebase for approved, withdrawn and investigational drugs. Along with information of drug structures and clinical trials, DrugBank provides information on the gene targets of each drug, which is crucial in better understanding a drug's mechanism of action. While DrugBank does provide an XML download of its contents, it is frequently updated online in between releases, making web scraping necessary. We would like you to build a web scraper that can capture the gene name of all targets for a list of given drugs.

**Assignment:** We're asking that you 1) build a web scraper that collects the gene name of all targets for a predefined set of DrugBank IDs, and 2) build an API endpoint that accepts a DrugBank ID and returns the gene targets.

**Task 1 Web Scraper Functional Requirements:** Your program should do the following:

- Make network requests to fetch a list of gene names of all targets (when available) for a predefined set of DrugBank IDs
- Persist the DrugBank IDs along with their gene targets
- Be able to be run by a single terminal command

**Task 2 API Functional Requirements:** Your program should do the following:

- Accept a DrugBank ID and return gene target in a modern data exchange format
- Instructions showing how to stand up your API on a linux or macos machine
- Example GET url request or bash script that makes a curl request to your API

**Non-Functional Requirements:**

- Python, please!

**Library suggestions & Ideas:**

You have complete freedom to use the environment/packages you are most comfortable/familiar with!

**Deliverable:** You should zip your code and e-mail the submission to [coryandar@onethree.bio](mailto:coryandar@onethree.bio). A github repo would be even better.

**Evaluation:** You will be evaluated on the correctness of your implementation and the quality of your code. Proper functionality is the most important. Ideally, the code should be clean, easy to read, and well commented. Your code should be modular, with proper separation of concerns.

### Task 1 Details (Predefined set of DrugBank IDs):

Scrape and store the approximate synonyms of the following IDs of interest\*:

-DB00619  
-DB01048  
-DB14093  
-DB00173  
-DB00734  
-DB00218  
-DB05196  
-DB09095  
-DB01053  
-DB00274

**\*Note:** extra credit if you can programmatically accomplish this for [all drugs within DrugBank](#)

An example of the data we are interested in (for DrugBank ID DB00317) can be seen under the “Targets” section in the following link: <https://www.drugbank.ca/drugs/DB00317>. We have also attached a screenshot below.

The screenshot shows the 'Targets' section for Gefitinib (DB00317) on the DrugBank website. The target is identified as the Epidermal growth factor receptor (EGFR). The page includes a table with the following information:

Kind	Protein	General Function	Ubiquitin protein ligase binding
Organism	Humans	Specific Function	Receptor tyrosine kinase binding ligands of the EGF family and activating several signaling cascades to convert extracellular cues into appropriate cellular responses. Known ligands include EGF, TG...
Pharmacological action	Yes	Gene Name	EGFR
Actions	Antagonist	Uniprot ID	P00533
		Uniprot Name	Epidermal growth factor receptor
		Molecular Weight	134276.185 Da

Below the table, there is a 'References' section with a list of six references related to the target and the drug.

As an example, our final datastore would contain all of the gene targets for DB00317 (Gefitinib), which includes only one target, EGFR.

**Task 2 Details (API Endpoint):**

Feel free to use any technologies you are most comfortable with. Again, the API should accept a DrugBank ID and return the approximate synonyms. To verify functionality, you should expect to run the API locally on your machine and provide a curl command in bash that sends the request and displays the response to us.