



Utrition Setup Manual

Note: Sections 1, 3, and part of Section 4 are from Utrition 2's Utrition Setup Manual.

This document will take you through all the steps necessary to get the project to run.

Namely, it will take you through the following steps:

1. Cloning our source code and installing any dependencies
2. Setting up the PostgreSQL database and loading it with data
3. Setting up Elasticsearch and loading it with data
4. Running the Flask webserver to serve the application

By the end of the guide, you should be able to access the web app in your browser and have everything work.

Part 1: Cloning source code, installing dependencies

1. First clone the project's source code from GitHub to your local machine:

```
$ git clone https://github.com/EngageHealth/Utrition
```

2. Next create a Python virtual environment to store the project's dependencies:

```
$ virtualenv utritionVenv
```

Note that a "virtual environment" is basically just a folder that will store your project's dependencies. They are not technically required but they are a best practice, because they will keep the packages for each of your projects totally separate and isolated. Feel free to see the included "Python virtual environments explanation" file if you'd like to know more about virtual environments; otherwise it's enough just to run the commands.

Also note: if the 'virtualenv' command is not found, you need to install it with a `$ pip install virtualenv`.

3. Enter/Activate the Python virtual environment:

```
$ source utritionVenv/bin/activate
```

After this command, pip will install all your dependencies into the virtual environment (into the folder) for this project. This keeps them separate from all your

other projects and from your global packages, hence why it's a best practice.

You will know that you are inside the virtual environment because a "(utritionVenv)" will be added to the start of your command prompt. Again -- this virtual environment contains all of the dependencies the project needs -- so make sure that you are inside it whenever you are running the project and for the remainder of the guide.

When you are done working on the project, you can simply run `$deactivate` to exit the virtual environment and go back to your global packages.

4. Install the project's dependencies:

```
$ pip install -r requirements.txt
```

This will install (namely) the following Python packages:

- Flask: for running the web server
- Elasticsearch: for talking with Elasticsearch

You may run a `$pip list` to confirm that the package were successfully installed.

Part 2: Setting up PostgreSQL database, loading it with data

5. Install PostgreSQL on your machine

a. MacOS users may install PostgreSQL using postgres.app

b. Installing PostgreSQL can also be done through:

i. [PostgreSQL](https://www.postgresql.org/)

ii. Homebrew install:

```
$ brew install postgresql
```

c. PostgreSQL will be correctly installed if you run `$ psql` and you don't receive an error like "command not found"

6. Create a user on the server that will be used to administer the Utrition database.

a. It's important that you remember both the Master and User password

7. If not done already, download pgAdmin4. This is the GUI used to interact with the database. This isn't strictly required, but will make interacting with the database simpler.

a. Download pgAdmin4 [HERE](#)

8. Download the DBSetup folder located in the SP 22 folder on Google Drive and unzip the downloaded DBSetup.zip file. You may also watch the corresponding DBSetup.mp4 video for a video tutorial of steps 8-11.
9. Unzip dataDB.zip
 - a. NOTE: Please see the Appendix and Extra Notes for important notes on the data
10. Update the populateDB.txt file with absolute paths that point to the data files in the dataDB folder.
 - a. Note that Windows may only require relative paths. Mac users must use absolute paths
11. Update the configUtrition.txt file with [absolute paths] that point to populateDB.txt and createDB.txt
12. Use the PSQL commands provided in the configUtrition.txt file to initialize the database.

To verify that the Utrition 4 database was successfully created:

```
$ psql \utrition4
```

Other versions of this command are:

```
$ psql -h localhost -d utrition4
```

```
$ psql -h localhost -d utrition4 -U <ADMIN_USER>
```

To verify that all tables were successfully created:

```
$ psql \utrition4
```

```
=# \dt
```

Part 3: Setting up Elasticsearch, loading it with data

13. Install Elasticsearch on your system.
Use the installation instructions provided [here](#).

Note that like many SQL databases, the way Elasticsearch(ES) works is that it runs a server in the background. Then, all client applications simply talk with that ES server in order to insert data, delete data, run queries, etc. So fundamentally, you make HTTP requests that go to the ES server in order to do anything with Elasticsearch.

Thus, to verify that your installation worked, try hitting the ES server with curl -- specifically try running `$ curl localhost:9200`

. If the installation was successful, you should get a response like this from the ES server:

```
(utritionVenv) jared@jared-HP-ENVY-PC:~/.../Utrition$ curl localhost:9200
{
  "name" : "jared-HP-ENVY-PC",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "Nx0qV9ZuQ3GdGL-Im4zPSw",
  "version" : {
    "number" : "7.11.1",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "ff17057114c2199c9c1bbecc727003a907c0db7a",
    "build_date" : "2021-02-15T13:44:09.394032Z",
    "build_snapshot" : false,
    "lucene_version" : "8.7.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

If you'd like to know more about Elasticsearch, feel free to see the file "Quick elasticsearch summary" that we made as a basic introduction to the technology. Otherwise forge ahead.

14. Load/Seed all the data into Elasticsearch. To do this, just run:

```
$ python elasticsearch-config/es-setup.py
```

which will index all the food data from the food.json file into ES.

Note that this takes ~30 minutes for us, so it is normal for it to take a while.

Part 4: Running the Flask web server

15. Create a user on the server to run the webserver. This user should have the same name as the user created to administer the Utrition database. This is due to the peer authentication requirements of the PostgreSQL database.
16. Enter the database connection details in the db.ini file found in the Utrition folder. This is imperative if the parser is going to successfully upload DNA files to the database.
17. Install psycopg2 (used to connect/interact with the database) by running


```
$ pip install psycopg2
```
18. Install configparser (used for reading the db.ini file) by running

```
$ pip install configparser
```

19. Install regex (used for parsing the DNA file) by running

```
$ pip install regex
```

20. Still inside the virtual environment, run

```
$ python app.py
```

to start the webserver.

21. By default the server listens on port 5000; so simply enter localhost:5000 into your web browser and you should see the running application. [If port 5000 doesn't work, choose another port that is not in use and update the port number on the last line of app.py.] Like this:



Welcome to Utrition!

Impacts every system of the body. From a holistic health and cancer survivorship perspective, nutrition data collection enables the evaluation of the overall dietary status, but cautions when functional nutritionists, clinical informaticians, and geneticists go beyond micronutrients.

A mobile-friendly webapp which takes a user through the 24-hour Food recall links microbiome methylation patterns, providing both patients and clinicians with personalized "Eat This" recommendations.

Enter your email, then click the submit button to begin! If you do not have an account, the system will redirect you to a registration page.

APPENDIX:

Information for the 'disease', 'rsid', and 'mutated_rsid' tables will need to be webscraped. Gathering this information is a rather slow process. Moreover, NCBI's servers aren't always able to return all of the information. In running the web scraping script, you may often run into 'HTTP Error 500: Internal Server Error'. This is an error on NCBI's side and is something that we cannot control. We recommend running this script outside of U.S working hours.

Also note that successfully running this script requires psql to be installed and correctly functioning.

To be able to run the webscript successfully, you will need to install the BioPython module using:

```
$ pip install biopython
```

Ensure that all four web scraping scripts (ncbi_webscraping.sh, queryCount.py, esearch_webscraping.py, and efetch_webscraping.py) are all in the same directory.

Navigate to the directory containing these threfoure files. In your terminal/shell, run the following commands:

```
$ chmod +x ncbi_webscraping.sh
$ chmod +x queryCount.py
$ chmod +x efetch_webscraping.py
$ chmod +x esearch_webscraping.py
```

The commands above ensure that you're able to execute all three files. To begin web scraping, run the command:

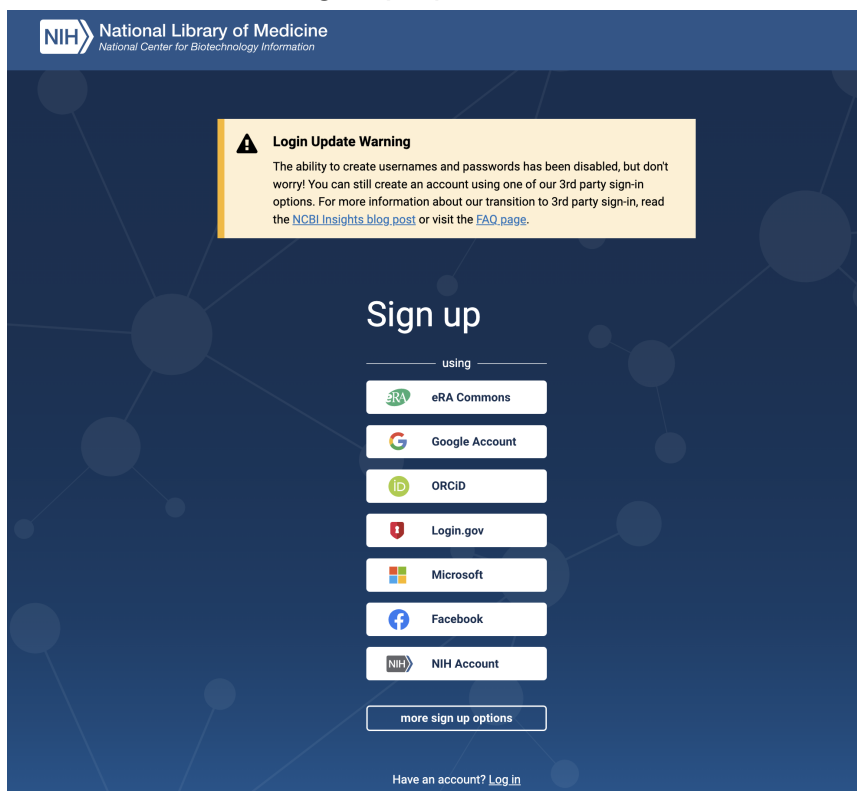
```
$ ./ncbi_webscraping.sh
```

As mentioned before, gathering data from NCBI can take quite a long time. Do not be surprised if gathering even 50 records takes up to a minute.

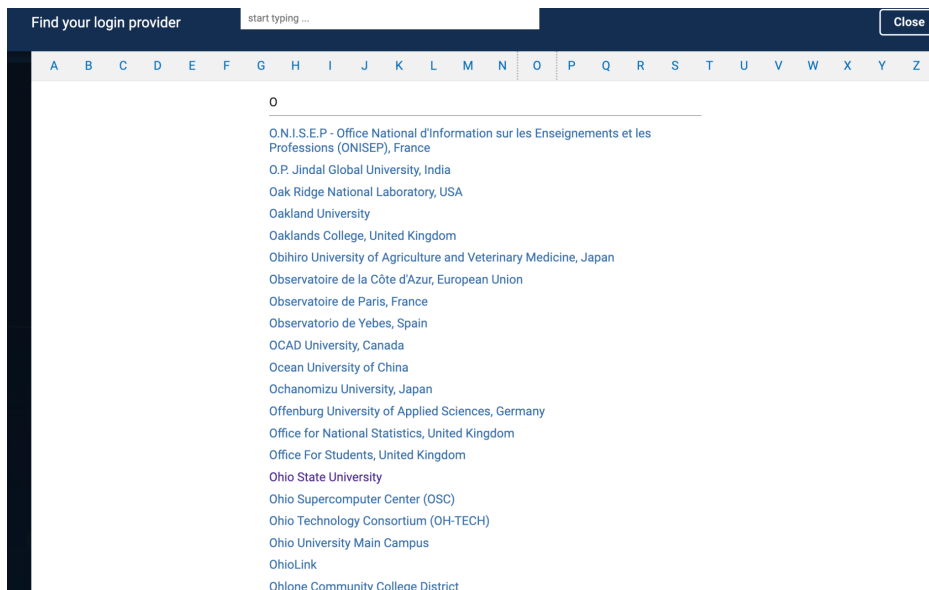
One way to speed up the retrieval process is by using an API key. We haven't noticed a significant speed up, but this is a method that NCBI has recommended.

Get an API key by following these steps:

1. Visit the [sign-up](#) page
2. Click on the 'more sign-up options' button



3. Search for 'Ohio State University' and click on it



This should redirect you to a page that looks like:

OSU.EDU

Help BuckeyeLink Map Find People Webmail Search Ohio State

Login Required

A login is required to access the requested service.

Ohio State Username

lastname.#

Password

Is this a [private](#) computer/device?

☒ Yes ☐ No

Forgot username? Forgot password?

Login

VIEW SYSTEM STATUS AND MAINTENANCE

Need Help?

Forgot your username?

Reset your password?

Need to activate your Ohio State Username?

Other questions?

About Ohio State Usernames

About OSU Web Login

Contact IT Service Desk

IMPORTANT LOGIN INFORMATION

These information systems, including all related equipment, networks, and network devices, are provided solely for use authorized by The Ohio State University. Use of these systems constitutes consent to abide by The Ohio State University's [Responsible Use of University Computing and Network Resources Policy](#). The Ohio State University may monitor use of these information systems without notice. Unauthorized disclosure of information, or evidence of unauthorized use may be subject to administrative action, civil action, and/or criminal prosecution.

For help with password or login issues, contact the IT Service Desk by calling 614-688-4357 (HELP) or online at <http://ocio.osu.edu/selfservice>.

The Ohio State University will **NEVER** ask for your password via email, phone, or any other method. If you receive such a message or have replied to one, please report it to report-phish@osu.edu. **NEVER** reply to any email asking for your account information or other personal details.

To protect your privacy, [completely clear your web browser](#) when finished.

Login will remain in effect until you completely clear your browser or several hours have elapsed.

- Log-in using your OSU log-in information
- After you log-in and are re-directed to the NCBI site, click on your email in the top-right corner.

An official website of the United States government [Here's how you know.](#)

NIH

National Library of Medicine

National Center for Biotechnology Information

shah.1440@osu.edu

My NCBI

Customize this page | NCBI Site Preferences | Video Overview | Help

Search NCBI databases

Search: PubMed

Search

Hint: clicking the "Search" button without any terms listed in the search box will transport you to that database's homepage.

My Bibliography

Your bibliography contains **no items**.

Your bibliography is **private**.

Manage My Bibliography >

Recent Activity

Time	Database	Type	Term
Yesterday 01:55 AM	Books	record	The E-utilities In-Depth: Parameter...
Yesterday 01:40 AM	Books	record	Entrez Direct Examples - Entrez Pro...
Yesterday 01:40 AM	Books	record	Entrez Direct: E-utilities on the U...
Yesterday 01:40 AM	Books	record	Table 1. [- Valid values of &retmod...
Yesterday 01:38 AM	Books	record	A General Introduction to the E-uti...
13-Apr-2022	SNP	search	(rs1588453974[Reference SNP ID]) AN...
13-Apr-2022	SNP	search	rs1588453974
13-Apr-2022	Books	record	Sample Applications of the E-uti...

Saved Searches

You don't have any saved searches yet.

Go and [create some saved searches](#) in PubMed or our other databases.

Manage Saved Searches >

Collections

All bibliographies and Other citations are now in My Bibliography

Collection Name	Items	Settings/Sharing	Type
Favorites	edit 0	Private	Standard

Manage Collections >

Filters

Filters for: PubMed

You do not have any active filters for this database.

[Add filters for the selected database.](#)

Manage Filters >

SciENcv

[Click here to create a new CV.](#)

6. Select 'Account Settings' in the drop down

The screenshot shows the NIH My NCBI dashboard. At the top, there's a blue header with the NIH logo and text: "National Library of Medicine National Center for Biotechnology Information". Below the header, the "My NCBI" section is visible. On the right side, an "ACCOUNT" dropdown menu is open, showing the user is logged in as "shah.1440@osu.edu". The menu options are: Dashboard, Publications, Account settings (highlighted), and Log out. The dashboard itself has several panels: "Search NCBI databases" with a search bar set to "PubMed"; "Saved Searches" with a message that no searches are saved; "My Bibliography" showing it is private; "Recent Activity" with a table of recent searches; "Collections" showing a table with one collection named "Favorites"; and "Filters" for the PubMed database.

7. Scroll down to the 'API Key Management' section

8. Click on the 'Create an API Key' button.

The screenshot shows the "NCBI Account Settings" page. At the top, there's a blue header with the NIH logo and text: "National Library of Medicine National Center for Biotechnology Information". Below the header, the "My NCBI » Settings" section is visible. A green checkmark indicates that the API key has been deleted successfully. The page is divided into several sections: "Email" with the email address "shah.1440@osu.edu" (confirmed) and a "Change" button; "NCBI Account" with the username "shah.1440@osu.edu"; "Linked accounts" with "Ohio State University" and the email address "shah.1440@osu.edu" (currently signed in via this method) and a "Change" button; "Delegates" with a message that delegates can be added to manage bibliographies and/or SciENcv profiles, and a link to "Add a Delegate"; "API Key Management" with a "Create an API Key" button and a message that E-utilities users are allowed 3 requests/second without an API key, and a link to the "helpo.development" page; and "NCBI Site Preferences" with a link to "Click here to access the NCBI Site Preferences page."

9. You should now have your API key.

NCBI Account Settings

Email

shah.1440@osu.edu (confirmed)

This email is used for delivery of saved searches.

Change

NCBI Account

Your Username is the email address of the 3rd-party account you used to register your NCBI Account.

Username: shah.1440@osu.edu

Linked accounts

You can sign in via these 3rd-parties. Contact the 3rd party for sign-in related issues.

Ohio State University

shah.1440@osu.edu (currently signed in via this method)

Change

Delegates

You can add delegates to help you manage your bibliography and/or SciENcv profiles.

[Add a Delegate](#)

API Key Management

API Key

e56171b63df697be1631e21735674d888507

Replace

Delete

E-utils users are allowed 3 requests/second without an API key. Create an API key to increase your e-utils limit to 10 requests/second. Contact our [help department](#) if you need higher throughput. Only one API Key per user. Replacing or deleting will inactivate the current key. Use this key by passing it with `api_key=API_KEY` parameter. Refer to [documentation](#) for more.

How to use your API Key:

1. Open the `efetch_webscraping.py` and `esearch_webscraping.py` files.
2. Right below the import statements, in both files:
 - a. Replace the value for 'Entrez.email' with the [OSU] email used to sign up for NCBI
 - b. Replace the value for 'Entrez.api_key' with the API Key generated on your NCBI account

EXTRA NOTES:

1. Food-related data will be provided. No extra web scraping necessary.
2. Data for the 'dna', 'food_recall', 'users', and 'user_recall' tables will come from user input from the UI.
 - a. In the event that the UI is unable to be set up, a separate script will be provided to enter data for those tables into the database.
 - b. However, this data is not a full substitute for the actual data being entered from the UI. The number of sample records is small, and tables with traditionally large numbers of records like the 'dna' table have a much smaller record count.