# January 19, 2023

<https://hc-sc.api.canada.ca/en/detail?api=cvp>

A picture containing shape

Description automatically generated

Graphical user interface, text, application

Description automatically generated

A picture containing application

Description automatically generated

Timed out, 11 times in a row

Graphical user interface

Description automatically generated with low confidence

Graphical user interface, application, Word

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated with low confidence

Timed out, 4 times in a row

A picture containing application

Description automatically generated

A picture containing graphical user interface

Description automatically generated

A picture containing application

Description automatically generated

* Since most of the API calls are not responding, we decided to reject the API.
* Kenny suggested using DPD.

# **Drug Product Database (DPD)**

<https://hc-sc.api.canada.ca/en/detail?api=dpd>

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated with low confidence

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated with low confidence

Graphical user interface, text, application

Description automatically generated

Suggestions made for wrapping this API:

Graphical user interface, text, application

Description automatically generated

# Jan 21, 2023

* Was successfully able to get response from the API in Python code.
* Initially tried to get the forms and route of administration of the drug using drug code (id) as input.
* Group decision was taken to filter the features using the exact name of the drug instead of the drug id.
* Was successfully able to narrow down the drugs that were being searched by the search query since the exact name was being used as input.

# January 24, 2023

* Created a “function” which takes the **exact** name of the drug as input and converted it to a .py file.
* Driver code file made to check if the py file is working
* VimFunction\_code updated to match the contents in py file

# January 25, 2023

* Converted the code to a function that can be called rather than just code for which user inout was being hard-coded.
* Data validation equivalent for the user input. Checked to see if the user input matches any records returned by the API, if not, stop the program with a message.
* Try and except block implemented where the get request is made.

# January 26, 2023

* Kenny pointed out that the code was not working for certain brand\_names like "ISOPTO CARBACHOL 3%" and "PLACIDYL CAP 200MG" and recommended having a look at the issue.
* It was decided that this code will be referred/referenced by all groupmates to have uniformity in the code and save some development effort and make it easier to merge code later.

# January 27, 2023

* The issue that Kenny pointed out was looked into and turns out Python only reads a maximum of approximately 5000 rows of responses from a API call and that was the issue.
* Tried some fixes that were found on the net but none seemed to work. The biggest hunch was processing response in chunks and getting the complete response from there. The problem with this was that the string that had all the chunks together was very big and soon as it was decoded using json.loads(), it would again limit the responses.
* The solution found thus far is to read the response from the API website and save it as a json file locally.
* This way we already have the entire database of drugs with multiple details like:

Text

Description automatically generated

* This json file can periodically be updated and it will work fine. This will enable us to find the drug\_code (id) from the name of the brand that the user will enter and we can search for the route and form of that drug.
* Added new py file till we decide which one we will use
* Added code for new py file in .ipynb file if we need any changes this can be used for it

# January 28, 2023

* Created test file and wrote code to test my wrapper function. The test cleared all checks.
* Created a skeleton for other partners to write their test code into.
* Fixed the code to get exact matches of the brand name and not all possible matches.

Graphical user interface, text, application

Description automatically generated

Checked results to see it works.

* Created a version of the code which splits the entire functionality into two functions rather than one. Was done as backup just in case the coverage percentage does not come out to be well, then this code can be implemented.

# January 30, 2023

* Modified the py file to take token/user-key as input
* Checked the coverage of the test file
* Setup the git ignore file to not track unwanted files.
* Added docstring
* Fixed folder structure