3.3.4 Using headers and parameters

You might have noticed that the headers are printed out in one line, making it difficult to read. Remember, headers are a way to transmit additional information with an HTTP request, and being able to easily read the information they store is valuable. One work around is to set parameters and headers.

Set parameters and headers

There are several API parameters that you would use to determine the type of action to take on the resource. There are four types of `REST` API parameters: Query, template or path, matrix, and plain. **Query** parameters are by far the most commonly used and form the focus of this topic. Before we explore query parameters in more detail, select an icon for brief descriptions of the three other parameters.

Among other things, query parameters are used in API requests to **extend the functionality of the API.** You might have noticed parameters were used in the code from the `GET` request earlier. Let's briefly explore what they are.

What are query parameters?

Query parameters are simply a way of **storing** data. Like cookies store data on the client-side, query parameters store data in the URL. You can see this data in your web browser address bar when you see a long and complicated URL with question marks and equal signs.

The question mark is key, as it is after the question mark in the URL that you see the parameters. Everything after the question mark is a query parameter. These parameters are `key:value` pairs. Essentially, the parameters are extensions of the URL used to specify content or actions.

More than one query parameter forms a **parameter string**, where each parameter is separated by the ampersand symbol (`&`). The query string can feature a variety of object types, such as arrays, strings, and numbers (RapidAPI 2021).

Using query parameters in HTTP requests

Query parameters are used in `GET` requests as a form of customisation by passing values through the parameter string or, in other words, parameterising the `GET` request. What's more, it helps retrieve specific data and performs actions based on user inputs.

One way to do so is to pass data to parameters or params. You can pass params as a dictionary, list, or tuple. So, for example, passing a list to the params parameter of the `GET` request modifies the results that come back. Another way to pass query parameters is to build custom URLs. The request method utilises the query string's values to complete the URL, which another function in the program can then use.

If you wanted multiple params and multiple headers, you could just expand params and headers further like this:

```
# SAMPLE CODE:
# Use dictionary format.
params = {'freeform': 'Something is HERE',
          'test': 'test'}

headers = {'content-type': 'application/json',
           'test': 'test'}
```

You can add a fair amount of specific headers and params this way.

Apply your learning

Head over to **Coding Rooms (https://app.codingrooms.com/app/org/swinburne-online-NHciF9n/course/cos60016-q1-programming-for-development-2023-piKGheK/b/3.3.3%2BHTTP%2Brequests%2Bin%2BPythonVk0G5Bj)** to learn how to set headers and parameters.