API & WebServices Testing
Assignment 2

Question:

What is the difference between path params and query params and write the running example code with API's for both.
Note - Example API’s from https://reqres.in/"

Solution:

Path Params:
Path parameters are part of the URL path and are used to identify a specific resource. They are often used for CRUD (Create, Read, Update, Delete) operations.

Syntax: Path parameters are represented as a variable segment in the URL path, enclosed in curly braces, such as "{user_id}".

Usage: Path parameters are used to identify a specific resource, and are typically used for operations that involve retrieving, updating, or deleting a single resource.

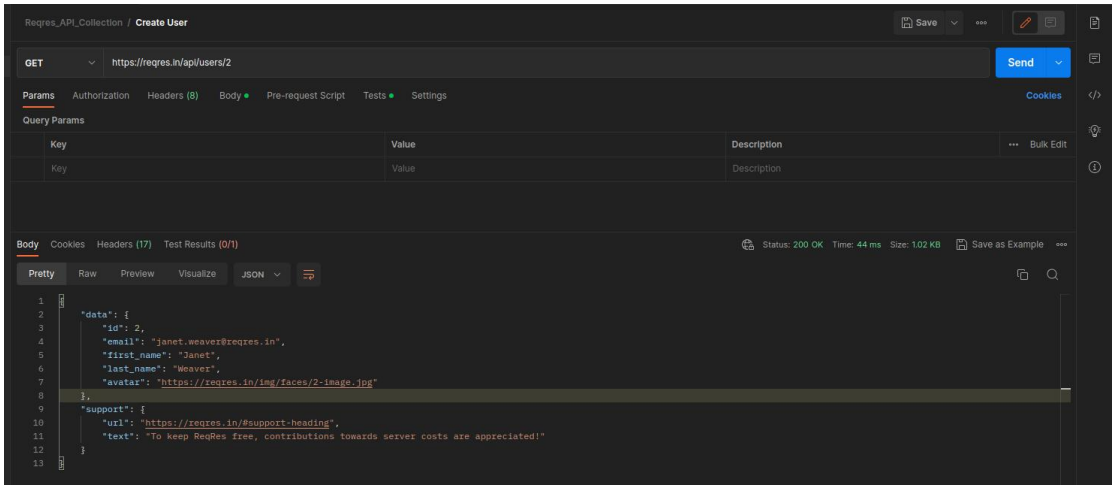
Data Type: Path parameters are limited to a specific data type, such as integer or string, and are often used to represent unique identifiers, such as user IDs or product IDs.

Positioning: Path parameters are positioned in the URL path, and are therefore more visible and less flexible than query parameters.

Security: Path parameters are generally considered more secure than query parameters, because they are part of the URL path and cannot be easily tampered with or intercepted.

For example, in the URL "https://reqres.in/api/users/:id", ":id" is a path param that can be replaced with the ID of the user to retrieve.

Such as GET **https://reqres.in/api/users/2**, where “2" is a path parameter that identifies the specific user resource.



Query Params:
Query parameters, on the other hand, are added to the end of the URL as a key-value pair separated by an ampersand symbol. They are used to filter or modify the data being returned. Query parameters are typically used when retrieving a list of resources.

Syntax: Query parameters are added to the end of the URL after a question mark, represented as a key-value pair, separated by an ampersand symbol, such as "?page=2&per_page=10".

Usage: Query parameters, on the other hand, are used to filter or modify a list of resources, and are often used for operations that involve retrieving multiple resources.

Data Type: Query parameters can support a wider range of data types, such as boolean, arrays, and objects.

Positioning: Query parameters are more flexible because they can be appended to the URL in any order, and can be added or removed without affecting the URL path.

Security: Query parameters, on the other hand, can be easily manipulated or altered by users or attackers, and may expose sensitive information or cause unintended side effects.

For example, in the request GET **https://reqres.in/api/users?page=2** is an API endpoint that returns a list of users in JSON format. Specifically, it returns the users on page 2 of the users list.

The base URL is "https://reqres.in/api/users", and the query parameter "page=2" specifies that the API should return the second page of users.