GET:

1. Validating list of available users.
2. Validating GET request returns a 200 Response code.
3. Validating GET request to a specific resource returns the correct data.
4. Validating when enter wrong id, name.

POST:

1. Validating by creating or updating a resource with POST request
2. Validating GET request for created resource, does not change any data
3. Validating create a resource with a POST request and ensure 200 Response code is returned.
4. Validating make a GET request for created resource, and ensure the data was saved correctly.
5. Validating add tests that ensure POST requests fail with incorrect data
6. Validating add tests that ensure POST requests fail with ill-formatted data.

DELETE:

1. Validating the user will be completely remove (New user is created with a POST request, and it can be retrieved with a GET request, then making DELETE)
2. Validating create a new user with a POST request
3. Validating the user id returned form POST request, make a DELETE request to
4. Validating GET request to user id and should return a 404 not found Response code
5. Validating after deleting a file with DELETE request, a GET request for that file should not return deleted file.
6. Validating DELETE request to an unknown user resource, should return a non 200 Response code

GET (ID):

1. Validating check HEAD request, returned a user
2. Validating make assertions against the Response code of HEAD request
3. Validating test HEAD request with various valid parameters.
4. Validating test HEAD request with various invalid parameters.

PUT:

1. Validating PUT request modifies existing file in the server will return
2. Validating make a GET request for modified existing file, and ensure the data was saved correctly.
3. Validating repeatedly calling PUT request always returns the same results
4. Validating correct Response code is returned when creating and updating a file

(Ex: 200, 201 response code is “OK”, 204 - response code is “No content”)

1. Validating after updating a file with PUT request, a GET request for that file should return the correct data
2. Validating PUT request should fail if invalid data is entered (nothing should be updated in the existing file)

## Test Scenarios: Blog/posts

### GET: [GET](http://localhost:8888/api/#!/blog%2Fposts/get_posts_collection) [/blog/posts/](http://localhost:8888/api/#!/blog%2Fposts/get_posts_collection)

1. Validating GET request returns a 200-Response code when Parameter = page and Value = integer.
2. Validating GET request returns a 400-Response code when Parameter = page and Value = non integer.
3. Validating GET request returns a 200-Response code when Parameter = bool and Value = true.
4. Validating GET request returns a 200-Response code when Parameter = bool and Value = false
5. Validating GET request returns a 200-Response code when Parameter = per\_page and Value = 10 (default)
6. Validating GET request returns a 200-Response code when Parameter = per\_page and Value = 2

### POST: [POST](http://localhost:8888/api/#!/blog%2Fposts/post_posts_collection) [/blog/posts/](http://localhost:8888/api/#!/blog%2Fposts/post_posts_collection)

1. Validating POST request do not return any Response code when Parameter = payload and Value = blank
2. Validating POST request returns a 200-Response code when Parameter = payload and Value = valid data
3. Validating POST request returns a 404-Response code when Parameter = payload and Value = invalid data

### GET: [GET](http://localhost:8888/api/#!/blog%2Fposts/get_posts_archive_collection) [/blog/posts/archive/{year}/](http://localhost:8888/api/#!/blog%2Fposts/get_posts_archive_collection)

1. Validating GET request returns a 200-Response code when Parameter = year and Value = integer (2010)
2. Validating GET request returns a 404-Response code when Parameter = year and Value = non integer (201e)
3. Validating GET request returns a 200-Response code when Parameter = page and Value = integer (2)
4. Validating GET request returns a 400-Response code when Parameter = page and Value = non integer (w)
5. Validating GET request returns a 200-Response code when Parameter = bool and Value = true.
6. Validating GET request returns a 200-Response code when Parameter = bool and Value = false
7. Validating GET request returns a 200-Response code when Parameter = per\_page and Value = 10 (default)
8. Validating GET request returns a 200-Response code when Parameter = per\_page and Value = 2

### GET: [GET](http://localhost:8888/api/#!/blog%2Fposts/get_posts_archive_collection_0) [/blog/posts/archive/{year}/{month}/](http://localhost:8888/api/#!/blog%2Fposts/get_posts_archive_collection_0)

1. Validating GET request do not return any Response code when Parameter = month and year and Value = blank (when both /any of the required parameter = blank)
2. Validating GET request returns a 200-Response code when Parameter = month and Value = integer (2)
3. Validating GET request returns a 404-Response code when Parameter = month and Value = non integer (e)
4. Validating GET request returns a 200-Response code when Parameter = year and Value = integer (2010)
5. Validating GET request returns a 404-Response code when Parameter = year and Value = non integer (201e)
6. Validating GET request returns a 200-Response code when Parameter = page and Value = integer (2)
7. Validating GET request returns a 400-Response code when Parameter = page and Value = non integer (w)
8. Validating GET request returns a 200-Response code when Parameter = bool and Value = true.
9. Validating GET request returns a 200-Response code when Parameter = bool and Value = false
10. Validating GET request returns a 200-Response code when Parameter = per\_page and Value = 10 (default)
11. Validating GET request returns a 200-Response code when Parameter = per\_page and Value = 2

### GET: [GET](http://localhost:8888/api/#!/blog%2Fposts/get_posts_archive_collection_0_1) [/blog/posts/archive/{year}/{month}/{day}/](http://localhost:8888/api/#!/blog%2Fposts/get_posts_archive_collection_0_1)

1. Validating GET request do not return any Response code when Parameter = day, month and year and Value = blank (when all /any of the required parameter = blank)
2. Validating GET request returns a 200-Response code when Parameter = day and Value = integer (2)
3. Validating GET request returns a 404-Response code when Parameter = day and Value = non integer (w)
4. Validating GET request returns a 200-Response code when Parameter = month and Value = integer (2)
5. Validating GET request returns a 404-Response code when Parameter = month and Value = non integer (e)
6. Validating GET request returns a 200-Response code when Parameter = year and Value = integer (2010)
7. Validating GET request returns a 404-Response code when Parameter = year and Value = non integer (201e)
8. Validating GET request returns a 200-Response code when Parameter = page and Value = integer (2)
9. Validating GET request returns a 400-Response code when Parameter = page and Value = non integer (w)
10. Validating GET request returns a 200-Response code when Parameter = bool and Value = true.
11. Validating GET request returns a 200-Response code when Parameter = bool and Value = false
12. Validating GET request returns a 200-Response code when Parameter = per\_page and Value = 10 (default)
13. Validating GET request returns a 200-Response code when Parameter = per\_page and Value = 2

### DELETE: [DELETE](http://localhost:8888/api/#!/blog%2Fposts/delete_post_item) [/blog/posts/{id}](http://localhost:8888/api/#!/blog%2Fposts/delete_post_item)

1. Validating DELETE request do not return any Response code when Parameter = id and Value = blank
2. Validating DELETE request returns a 204-Response code when Parameter = id and Value = integer (3)
3. Validating DELETE request returns a 404-Response code when Parameter = id and Value = non integer (w)
4. Validating DELETE request returns a 404-Response code when Parameter = id and Value = integer (3 – deleted previously )

### GET: [GET](http://localhost:8888/api/#!/blog%2Fposts/get_post_item) [/blog/posts/{id}](http://localhost:8888/api/#!/blog%2Fposts/get_post_item)

1. Validating GET request do not return any Response code when Parameter = id and Value = blank
2. Validating GET request returns a 200-Response code when Parameter = id and Value = integer (2)
3. Validating GET request returns a 404-Response code when Parameter = day and Value = non integer (w)

### PUT: [PUT](http://localhost:8888/api/#!/blog%2Fposts/put_post_item) [/blog/posts/{id}](http://localhost:8888/api/#!/blog%2Fposts/put_post_item)

1. Validating PUT request do not return any Response code when Parameter = id , payload and Value = blank (when both /any of the required parameter = blank)
2. Validating PUT request returns a 204-Response code when Parameter = valid id and Value = valid data
3. Validating PUT request returns a 404-Response code when Parameter = valid id and Value = invalid data
4. Validating PUT request returns a 404-Response code when Parameter = invalid id and Value = valid data