**Test Cases**

1. User gets created and can login with created credentials.

Steps:

1. Create user with following details {email – [eve.holt@reqres.in](mailto:eve.holt@reqres.in), password – pistol}.
2. Try to login with following details {email – [eve.holt@reqres.in](mailto:eve.holt@reqres.in), password – pistol}

Expected Result:

1. API returns 200 status and response contains token that is not null.
2. API returns 200 status and response contains token that is not null.
3. Logged-in user can view all accounts and then update one of them.

Steps:

1. Login as user with following details {email – [eve.holt@reqres.in](mailto:eve.holt@reqres.in), password – pistol}.
2. Get all users.
3. Check more than one users is displayed.
4. Perform update on first account by adding random data.

Expected Result:

1. Token gets returned.
2. API returns data.
3. There are more than 1 users returned.
4. API returns 200 status and response contains added data.
5. Logged-in user can try to delete one of accounts and then check that he did not actually delete.

Steps:

1. Login as user with following details {email – [eve.holt@reqres.in](mailto:eve.holt@reqres.in), password – pistol}.
2. Get first user’s details.
3. Perform delete on first user.
4. Get first user’s details.
5. Compare if both user details match.

Expected Result:

1. Token gets returned.
2. API returns data.
3. API returns 204 status.
4. API returns data.
5. Data matches.
6. User tries to create account when he can’t and then tries to login with that account without success.

Steps:

1. Register user with details {email – [user@reqres.in](mailto:user@reqres.in), password – passwd}.
2. Login user with details {email – [user@reqres.in](mailto:user@reqres.in), password – passwd}.

Expected Result:

1. API returns status 400 with error “Note: Only defined users succeed registration”.
2. API returns 400 status with error “user not found”.

**Tool Selection**

JAVA is used because it is most commonly used language for testing.

Maven is used for easy imports.

TestNG is used for running tests.

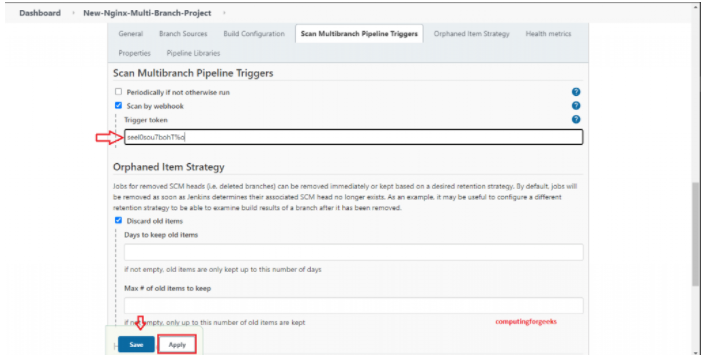
Rest-assured is used to perform API calls.

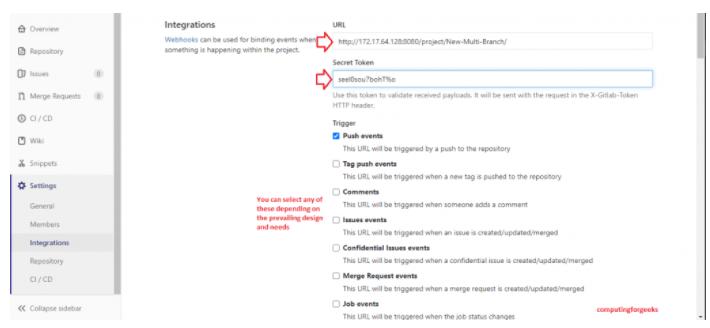
TestRail for registering results and storing test cases.

**Continues Integration**

To make pipeline work we need to create Jenkins file in the project to write configuration:

Provide which agent to use, steps and stages. In steps we provide maven commands to run TestNG suites. After that install *MultiBranch Action Triggers* plugin via Jenkins user interface. Then *Multibranch Pipeline* can be made where trigger actions can be specified and all Git credentials. To run tests after every push to master or any other branch a *webhook* trigger needs to be made (token) in the pipeline. That token later can be added to Git integrations (found in settings) where we can specify when Git is going to trigger a request to pipeline with that token. Push event needs to be selected to trigger pipeline after every push to the branch.





**Tool for result visualization**

I use TestRail, because it is most common and easy to use tool. It lets to define test cases and then perform runs where results can be filled in. Even screenshots can be added and it is easy to see steps when browsing results. It also provides pie diagram which shows how many cases failed and how many passed.

