

Assignment - Week2

Assignment 1: PostMan

- 1) Create an issue in jira
- 2) Update issue in jira
- 3) Delete the issue

Note : Implement the above with chaining process in Postman

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists collections like 'ChangeRequest' and 'Jira'. The main area is titled 'Runner' and displays a 'RUN ORDER' section. It contains five requests: 'AllIssues' (GET), '24hourbefore' (GET), 'Create Request' (POST), 'Update Request' (PUT), and 'Delete Request' (DEL). To the right of the requests are configuration options: 'Iterations' set to 1, 'Delay' set to 3000 ms, and several checkboxes for response saving, variable values, and cookie management. A large blue 'Run Jira' button is at the bottom right.

This screenshot shows the Postman Runner interface after the requests have been run. The 'Jira' collection is selected in the sidebar. The 'Runner' tab shows the results of the run. At the top, it says 'Jira Dev_Service_Now, just now'. Below that, it shows 'All Tests Passed (3) Failed (0)'. The 'Iteration 1' section details three successful requests: a POST 'Create Request' to https://restapijira.atlassian.net/rest/api/2/issue/ with a status of 201 Created, a PUT 'Update Request' to https://restapijira.atlassian.net/rest/api/2/issue/{{issue_id}} with a status of 204 No Content, and a DELETE 'Delete Request' to https://restapijira.atlassian.net/rest/api/2/issue/{{issue_id}} with a status of 204 No Content. Each request has a green 'Pass' status and a validation message like 'Validate the status code as 201' or 'Validate the status code as 204'.

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists collections, APIs, environments, mock servers, monitors, flows, and history. A search bar at the top right contains the text 'Jira'. The main area displays a collection named 'Jira' with the following requests:

- ChangeRequest**
 - GET GetCRJSON
 - GET GetCRXMLCategory
 - GET GetCRJSONFields
 - POST POST with Request Body JSON
 - POST POST with Request Body XML
 - PATCH New Request
 - DEL New Request
- CollectionChain**
 - GET GetCRJSON
 - POST POST with Request Body JSON
 - PUT New Request
 - DEL New Request
- Jira**
 - GET AllIssues
 - GET 24hourbefore
 - POST Create Request
 - PUT Update Request
 - DEL Delete Request

In the 'RUN SUMMARY' section, there is one successful POST request. At the bottom, there are buttons for 'View Results', 'Run Again', 'New', and 'Export Results'.

2. RestAssured:

1) Get All the issues in jira

The screenshot shows the Eclipse IDE interface with the 'eclipse-workspace - MavenProject/src/test/java/week2/assignment2/SimpleGetRequest.java - Eclipse IDE' window open. The 'Package Explorer' view shows the project structure with files like 'SimpleGetRequest.java', 'createRequestWithReqBody.java', and 'SimpleGetRequest.java'. The 'SimpleGetRequest.java' file contains the following code:

```

1 package week2.assignment2;
2
3
4 import io.restassured.RestAssured;
5
6 public class SimpleGetRequest {
7
8     public static void main(String[] args) {
9         // TODO Auto-generated method stub
10        SimpleGetRequest.sendGetRequest();
11    }
12
13    public static void sendGetRequest() {
14
15        //End point
16        RestAssured.baseURI = "https://restapijira.atlassian.net/rest/api/2/search";
17
18        //Add Request
19        RestAssured.authentication=RestAssured.preemptive().basic("RestAPIjira@gmail.com", "LQHwrjixuCa4eXz5mVKh7489");
20        RequestSpecification inputRequest = RestAssured.given()
21                    .queryParam("jql", "project=AP")
22                    .accept(MediaType.JSON);
23
24        //send Request
25        Response response = inputRequest.get();
26
27        //Validate response
28        int statusCode = response.statusCode();
29        System.out.println(statusCode);
30        response.prettyPrint();
31
32    }
33
34 }
35
36
37
38
39
40

```

The 'Console' view at the bottom shows the JSON response for a single issue:

```

{
  "self": "https://restapijira.atlassian.net/rest/api/2/project/10000",
  "id": "10000",
  "key": "AP",
  "name": "APIMAR",
  "projectTypeKey": "software",
  "simplified": false,
  "avatarUrls": {
    "48x48": "https://restapijira.atlassian.net/rest/api/2/universal_avatar/view/type/project/avatar/10405",
    "24x24": "https://restapijira.atlassian.net/rest/api/2/universal_avatar/view/type/project/avatar/10405?size=small",
    "16x16": "https://restapijira.atlassian.net/rest/api/2/universal_avatar/view/type/project/avatar/10405?size=xsmall"
  }
}

```

2) Create an issue in jira

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a Maven project named "MavenProject" with several source and test packages.
- Code Editor:** Displays the file `createRequestWithReqBody.java` containing Java code using the RestAssured library to create a Jira issue. The code includes importing `java.io.File`, defining a main method, setting the base URI to `https://restapijira.atlassian.net/rest/api/2/issue/`, adding a request with a JSON file body, and printing the status code.
- Console Output:** Shows the execution results:

```
<terminated> createRequestWithReqBody (1) [Java Application] /Volumes/Eclipse/Eclipse.app/Contents/Eclipse/plugins/org.eclipse.justj/openjdk.hotspot.jre.full.macosx.aarch64_17.0.1.v202111
Body:
./data/CreateIssue.json
{
    "id": "10311",
    "key": "AP-312",
    "self": "https://restapijira.atlassian.net/rest/api/2/issue/10311"
}
Status Code: 201
```