Task - Jira API Automation

https://developer.atlassian.com/cloud/jira/platform/rest/v3/api-group-issues/#api-rest-api-3-issue-issueidorkey-get

1. Create the defect in Jira.

2. Update the Defect using defect ID

3. Search the Defect created in the step 1

4. Add an attachment to the issue

5. Delete the defect created at step number 1

Not able to retrieve the below details

1. **Jira API Token**: need an API token to authenticate your requests.
2. **Jira Base URL**: The base URL for Jira instance (e.g., https://your-domain.atlassian.com).
3. **Authentication**: Basic Authentication with email and API token.

Observation: Written the code with only concepts and skeleton part to drive through the requirement steps.

**Code Snippet:**

public class JiraAPIAutomation {

static String baseURL = "https://your-domain.atlassian.com";

static String apiToken = "api\_token"; // Replace with Jira API token

static String email = "email@example.com"; // Replace with Jira email

static String issueID; // To store the issue ID created

public static void main(String[] args) {

// Set Base URI

RestAssured.baseURI = baseURL;

// Step 1: Create the defect in Jira

String createIssuePayload = "{\n" +

" \"fields\": {\n" +

" \"project\": {\n" +

" \"key\": \"PROJ\"\n" + // Replace with your project key

" },\n" +

" \"summary\": \"Issue created via API\",\n" +

" \"description\": \"Creating an issue via API automation\",\n" +

" \"issuetype\": {\n" +

" \"name\": \"Bug\"\n" +

" }\n" +

" }\n" +

"}";

Response createResponse = given()

.auth().preemptive().basic(email, apiToken)

.header("Content-Type", "application/json")

.body(createIssuePayload)

.when()

.post("/rest/api/3/issue")

.then()

.statusCode(201)

.extract()

.response();

issueID = createResponse.path("id");

System.out.println("Issue created with ID: " + issueID);

// Step 2: Update the Defect using defect ID

String updateIssuePayload = "{\n" +

" \"fields\": {\n" +

" \"summary\": \"Updated issue summary via API\",\n" +

" \"description\": \"Updated issue description via API\"\n" +

" }\n" +

"}";

given()

.auth().preemptive().basic(email, apiToken)

.header("Content-Type", "application/json")

.body(updateIssuePayload)

.when()

.put("/rest/api/3/issue/" + issueID)

.then()

.statusCode(204); // 204 No Content indicates success

System.out.println("Issue updated.");

// Step 3: Search the Defect created in step 1

Response searchResponse = given()

.auth().preemptive().basic(email, apiToken)

.when()

.get("/rest/api/3/issue/" + issueID)

.then()

.statusCode(200)

.extract()

.response();

System.out.println("Issue details: " + searchResponse.prettyPrint());

// Step 4: Add an attachment to the issue

given()

.auth().preemptive().basic(email, apiToken)

.header("X-Atlassian-Token", "no-check")

.multiPart("file", new File("path\_to\_your\_file.txt")) // Replace with your file path

.when()

.post("/rest/api/3/issue/" + issueID + "/attachments")

.then()

.statusCode(200);

System.out.println("Attachment added.");

// Step 5: Delete the defect created in step 1

given()

.auth().preemptive().basic(email, apiToken)

.when()

.delete("/rest/api/3/issue/" + issueID)

.then()

.statusCode(204); // 204 No Content indicates success

System.out.println("Issue deleted.");

}

}