A close-up of a logo

Description automatically generated

**Classroom Session: Chaining of Requests (Create, Update, and Delete) for Issues in Jira**

**Learning Objective:** The objective of this session is to learn how to create, update, and delete issues in Jira using a chained process in Postman, and to save and reuse the issue key from one request to the next.

**Expected Completion Time:**

* Best Case: 45 minutes
* Average Case: 60 minutes

**Session Details:**

### Setting Up Jira API in Postman

#### Preconditions:

1. **Jira API Credentials:**
   * Obtain your Jira instance URL.
   * Generate an API token from your Jira account settings.
   * Your Jira username/email.
2. **Set Up Environment Variables in Postman:**
   * **Create a New Environment:**
     + Click on the "Environment" menu.
     + Create a new environment named "Jira".
   * **Add Variables:**
     + Add a variable BaseUri with your Jira instance URL (e.g., https://your-domain.atlassian.net).
     + Add a variable ApiToken with your Jira API token.
     + Add a variable Username with your Jira username/email.

### Configuring Postman to Use Basic Auth with Jira API

#### Step-by-Step Guide:

**Part 1: Create an Issue**

**Step 1: Create collection**

1. Create a new Collection.

**Step 2: Create a New Request**

1. Click on "New" and select "HTTP Request".
2. Name your request appropriately, e.g., "Create Issue".

**Step 3: Set Up the Request URL**

1. Set the request type to POST.
2. Enter the Jira API endpoint for creating an issue:

**POST** {{BaseUri}}/rest/api/2/issue

**Step 4: Authentication**

1. Go to the "Authorization" tab in the collection level.
2. Select "Basic Auth".
3. Enter your Jira username and API token:
   * Username: {{Username}}
   * Password: {{ApiToken}}
4. Select “Inherit Auth from parent” For the create Issue request.

**Step 5: Body**

1. Switch to the "Body" tab.
2. Select raw and choose JSON from the drop-down menu.
3. Enter the JSON payload for creating an issue. Example:

{

    "fields": {

        "project": {

            "key": "<PROJECT KEY>"

        },

        "summary": "create issue in MNP project",

        "description": "Creating of an issue using project keys and issue type names using the REST API",

        "issuetype": {

            "name": "Story"

        }

    }

}

**Step 6: Save the Response as a Variable**

1. Go to the "**Scripts**" tab and add the following script to save the response as a variable:

*var* resp = pm.response.json();

*var* issueKey = resp.key;

pm.globals.set("issue\_Key", issueKey);

**Step 8: Send the Request**

1. Click "Send".
2. Observe the response in the lower pane and ensure it contains the details of the newly created issue, including the issue key.

**Part 2: Read the Issue**

**Step 1: Create a New Request**

1. Click on "New" and select "HTTP Request".
2. Name your request appropriately, e.g., "Read Issue".

**Step 2: Set Up the Request URL**

1. Set the request type to GET.
2. Enter the Jira API endpoint for reading an issue:

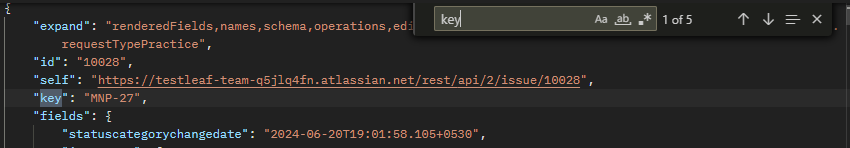
**GET** {{BaseUri}}/rest/api/2/issue/{{issueKey}}

**Step 3: Authentication**

1. Go to the "Authorization" tab.
2. Select "Inherit auth from parent" .

**Step 4: Send the Request**

1. Click "Send".
2. Observe the response in the lower pane to ensure it contains the details of the issue.



**Part 3: Update the Issue**

**Step 1: Create a New Request**

1. Click on "New" and select "HTTP Request".
2. Name your request appropriately, e.g., "Update Issue".

**Step 2: Set Up the Request URL**

1. Set the request type to PUT.
2. Enter the Jira API endpoint for updating an issue:

**PUT** {{BaseUri}}/rest/api/2/issue/{{issueKey}}

**Step 3: Authentication**

1. Go to the "Authorization" tab.
2. Select "Inherit auth from parent" .

**Step 4: Body**

1. Switch to the "Body" tab.
2. Select raw and choose JSON from the drop-down menu.
3. Enter the JSON payload for updating the issue. Example:

{

    "fields": {

        "description": "Bug creation Using REST API for testing"

    }

    }

**Step 6: Send the Request**

1. Click "Send".
2. Observe the response in the lower pane and ensure it contains the updated details of the issue. Ensure the response body with updated description.

**Part 4: Delete the Issue**

**Step 1: Create a New Request**

1. Click on "New" and select "HTTP Request".
2. Name your request appropriately, e.g., "Delete Issue".

**Step 2: Set Up the Request URL**

1. Set the request type to **DELETE**.
2. Enter the Jira API endpoint for deleting an issue:

**DELETE** {{BaseUri}}/rest/api/2/issue/{{issueKey}}

**Step 3: Authentication**

1. Go to the "Authorization" tab.
2. Select "Inherit auth from parent" .

**Step 4: Send the Request**

1. Click "Send".
2. Observe the response in the lower pane to ensure the issue has been successfully deleted.
3. Ensure the status code should **204 No content**.

### Expected Outcome:

Upon completion, you should be able to:

* Set up Basic Auth authentication in Postman for Jira.
* Chain API requests in Postman to create, read, update, and delete issues.
* Handle and troubleshoot common mistakes in API requests and Postman configuration.

Top of Form

Bottom of Form