

Postman

Version History



S. No	Revision Date	Version	Modifications	Modified By
1	10/03/2023	V1.0	Initially Created the Postman Document	Ayerathammal

Table of Contents

- What is Postman
- Why should we use purpose postman
- How to create a collection in the postman
- How to SSL settings in the postman
- How to setup Env Variables
- How to create test scripts
- How to import a collection in the postman
- How to export a collection in the postman

What is Postman

- Postman in the context of software, It is used for referring to an API (Application Programming Interface) development tool.
- Postman allows developers to design, test, and document APIs.
- It provides a user-friendly interface for making HTTP requests to test the functionality of an API.
- It simplifies the process of designing, testing, and documenting APIs, making it an essential tool for developers and testers involved in the API lifecycle.

Why Should we use Postman

- We use Postman to perform crud (Create, Read, Update and Delete) operation in the DataBase.
- **API Testing:** Postman allows developers to create and send HTTP requests to API endpoints, making it easy to test different HTTP methods (GET, POST, PUT, DELETE) and observe the responses.
- **API Documentation:** You can document your APIs within Postman, making it easier for other team members to understand how to use them. Postman can generate documentation from your API requests.

Why Should we use Postman

- We use Postman to perform crud (Create, Read, Update and Delete) operation in the DataBase.
- We used for API Testing to test Authorization, Headers, RequestBody, Methods,URL, Status Code and Response.
- In our Organization mostly we used 4 Methods for Performing CRUD Operation.

I.Create and Read Multiple – Post

II. Read Single – Get

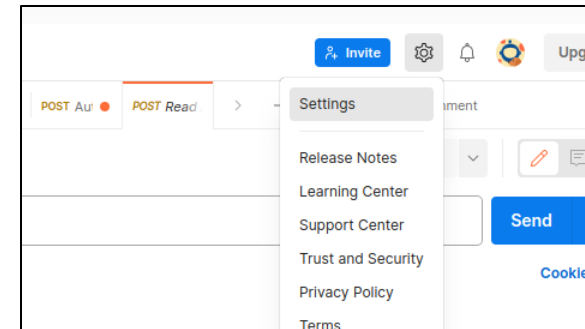
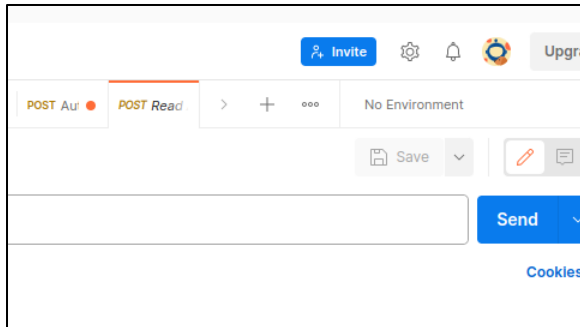
III. Update – Patch

IV. Delete Single - Del

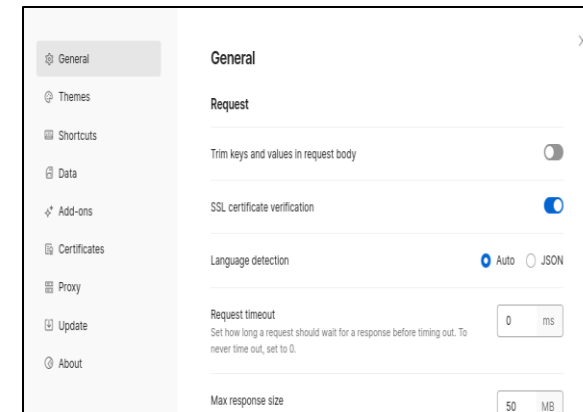
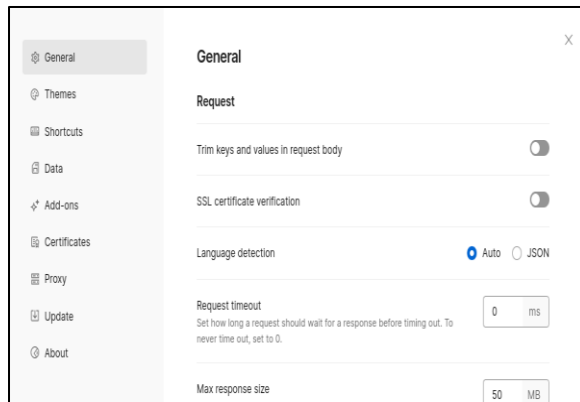
- **Collection Sharing:** You can create collections of API requests and share them with team members or the broader community. This is helpful for collaboration and ensuring that everyone uses the same API requests.
- **Environment Variables:** Postman supports environment variables, making it easy to switch between different environments (e.g., development, staging, production) without modifying each API request individually.
- **Authentication:** Postman supports various authentication methods, making it easy to test APIs that require authentication, such as OAuth or API keys.

How to SSL Settings in Postman

- **SSL Settings:** In the request configuration, there might be an option to specify SSL settings. Look for a dropdown or a tab related to security or SSL/TLS.

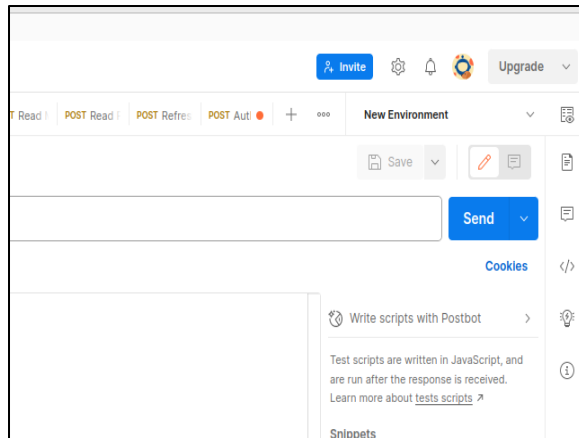


- **Enable SSL:** There should be an option to enable SSL. This might be labeled as "SSL" or "HTTPS." Enable this option.

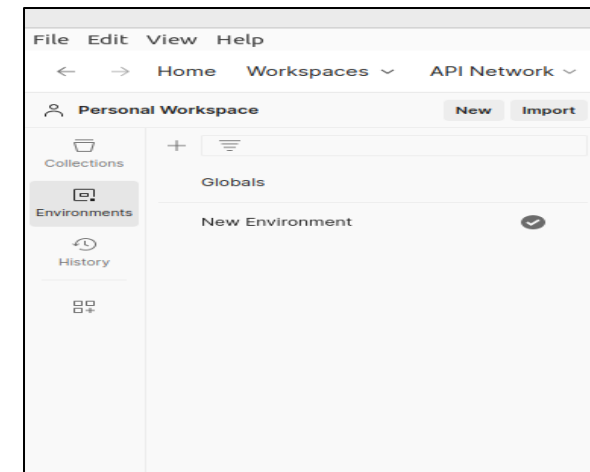


How to setup Env Variables

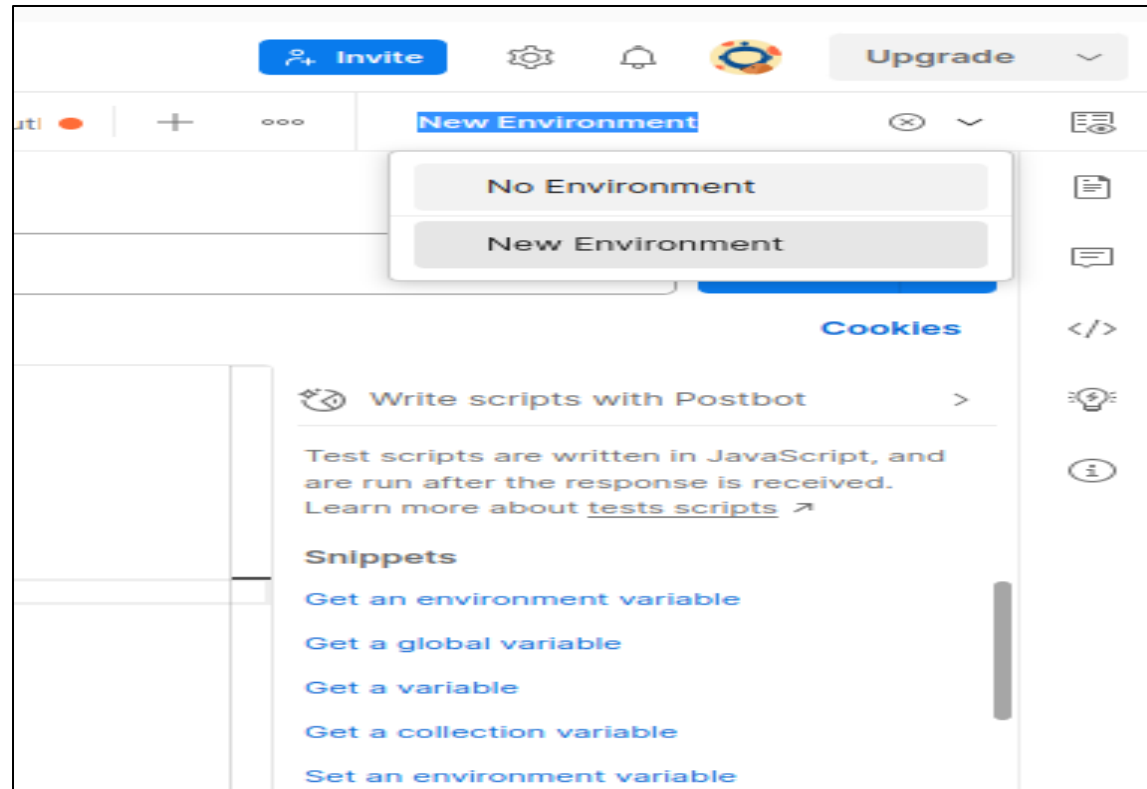
- Open Postman
- Create a New Request or Open an Existing request
- Click on the "Manage Environment icon" - It shows on the Upper right corner.



OR



- Click "Add" or "Create" a New Environment.



- Enter Environment Name and add Variables

Postman

Search Postman

Invite

ST Refre GET Bas POST Setup POST Read POST Setup POST Read POST Ge POST Setup POST Setup POST Read POST Read POST Read POST Refre POST Aut

New Environment

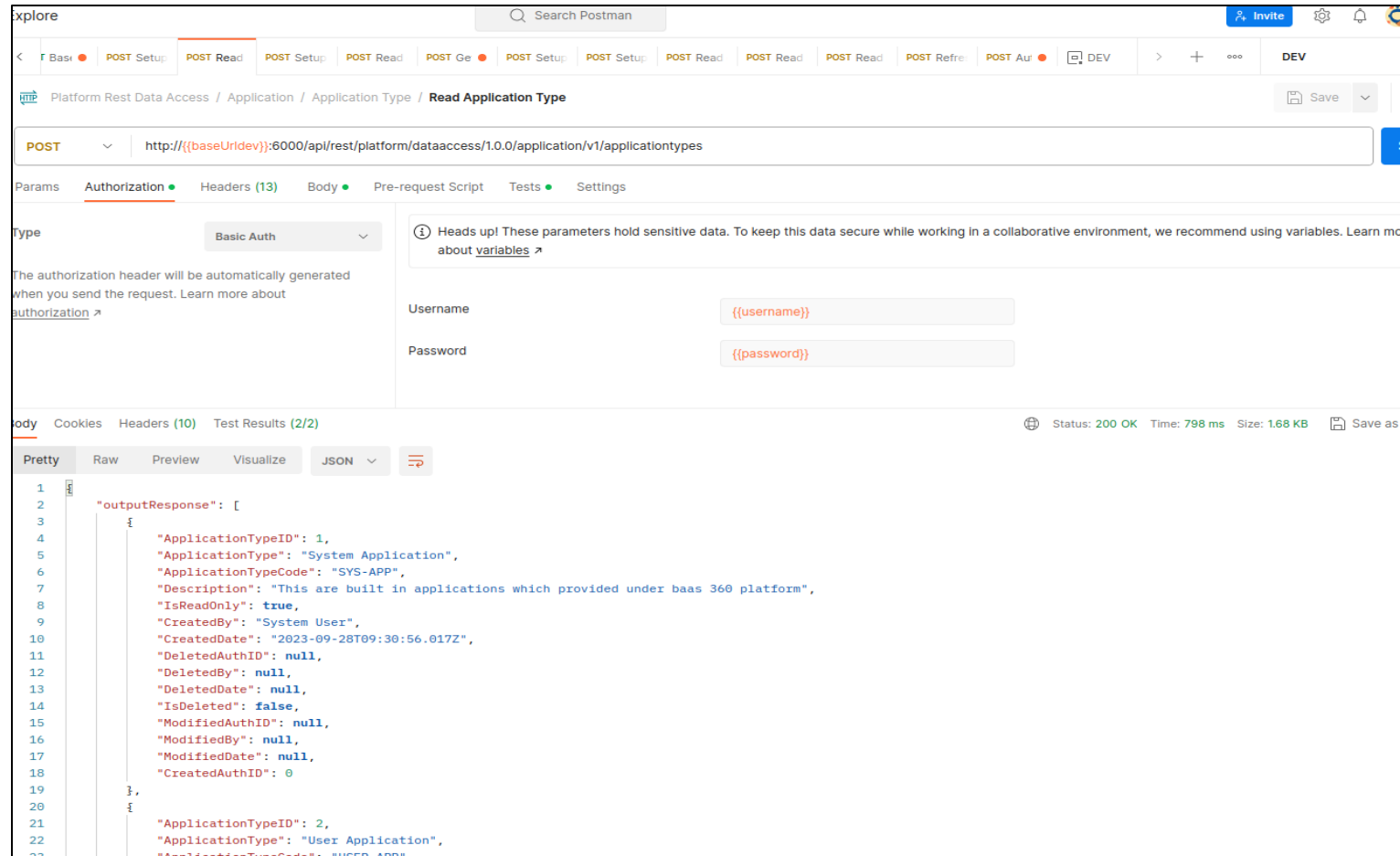
Fork 0 Save

Filter variables

	Variable	Type	Initial value	Current value
<input checked="" type="checkbox"/>	baseUrl	default	dev1.baas360.alitasys.com	dev1.baas360.alitasys.com
<input checked="" type="checkbox"/>	username	default		QP0192923232LD
<input checked="" type="checkbox"/>	password	default		927NBGJJ0283HKA74782
	Add new variable			

- Click "Add" to save the Environment.
- Select the Active Environment

- Use the Environment Variables in Request.
- Send the Request



The screenshot shows the Postman interface for a POST request. The URL is `http://{{baseUriDev}}:6000/api/rest/platform/dataaccess/1.0.0/application/v1/applicationtypes`. The request is configured with Basic Auth, where the Username is `{{username}}` and the Password is `{{password}}`. A warning message states: "Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables".

The response is displayed in the bottom pane, showing a JSON array of application types. The first object in the array is:

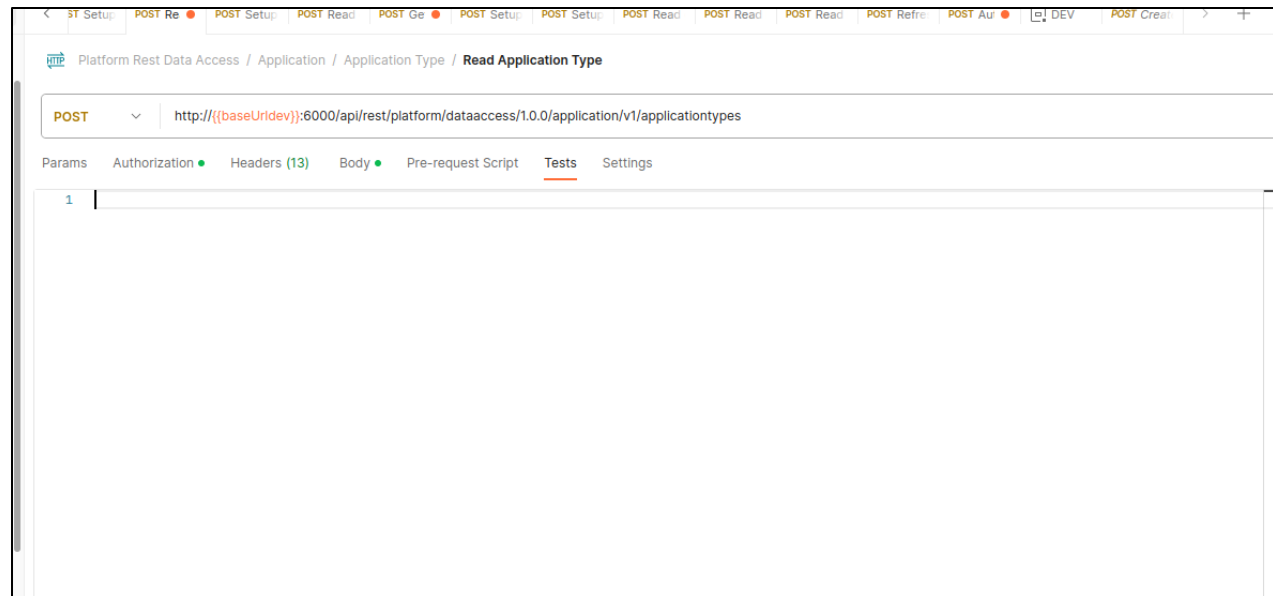
```

{
  "ApplicationTypeID": 1,
  "ApplicationType": "System Application",
  "ApplicationTypeCode": "SYS-APP",
  "Description": "This are built in applications which provided under baas 360 platform",
  "IsReadOnly": true,
  "CreatedBy": "System User",
  "CreatedDate": "2023-09-28T09:30:56.017Z",
  "DeletedAuthID": null,
  "DeletedBy": null,
  "DeletedDate": null,
  "IsDeleted": false,
  "ModifiedAuthID": null,
  "ModifiedBy": null,
  "ModifiedDate": null,
  "CreatedAuthID": 0
}

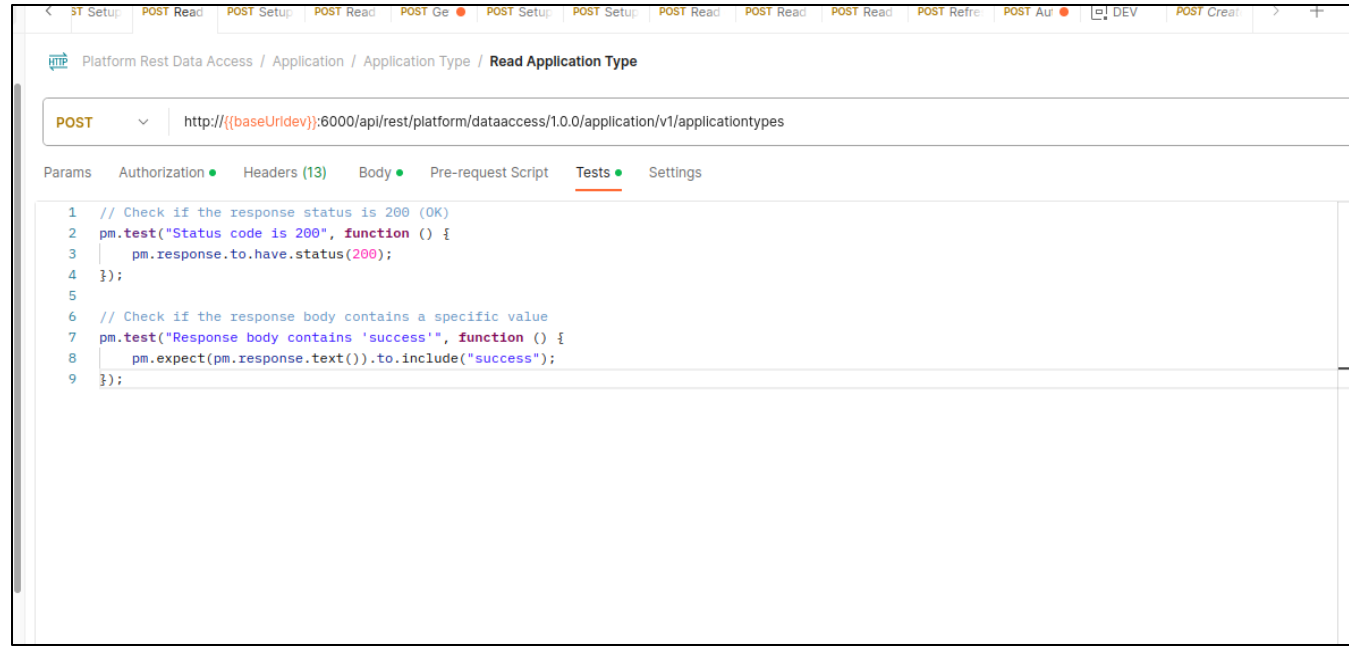
```

How to create Test Scripts

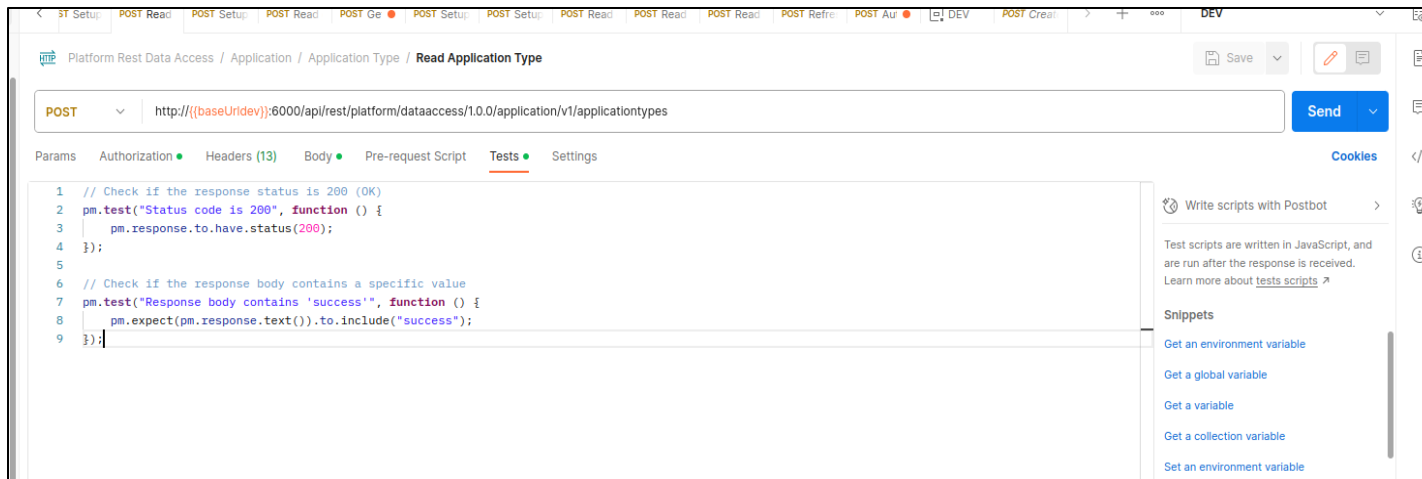
- Open Postman
- Create a New Request or Open an Existing request
- Go to the "Tests" Tab



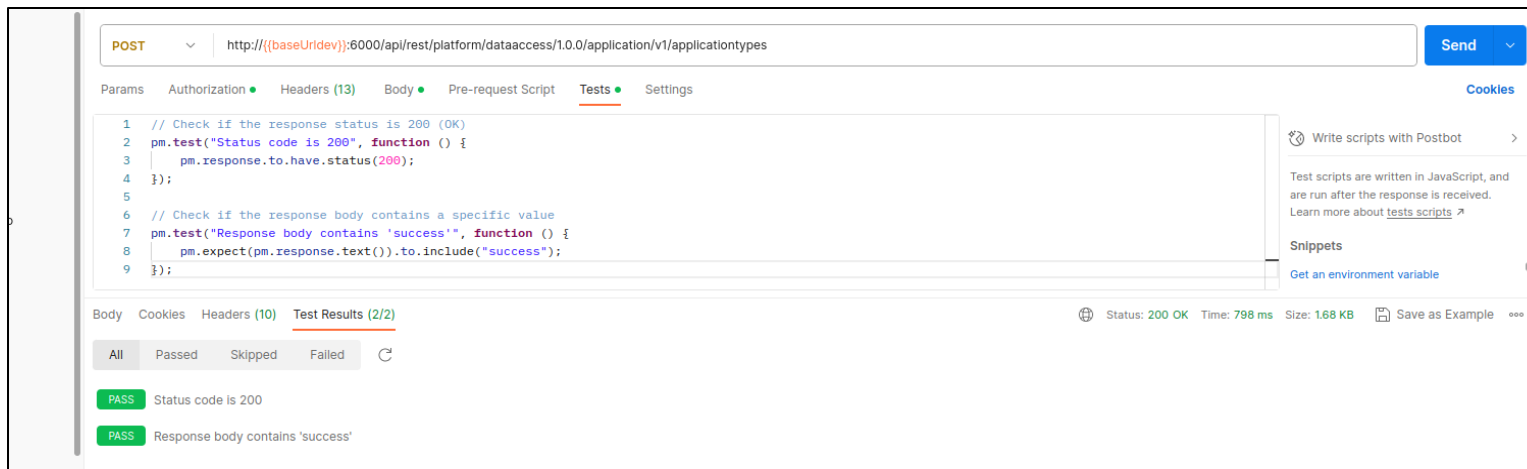
- Write a Javascript Test Scripts



- Save the Request
- Send the Request

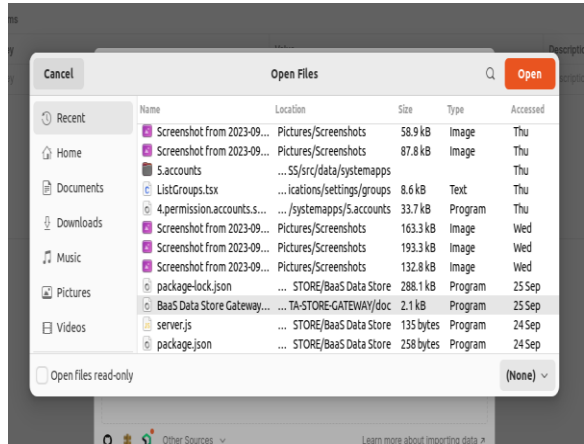
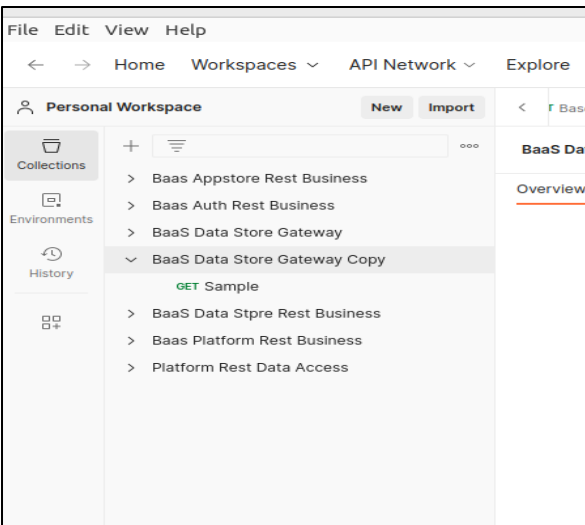
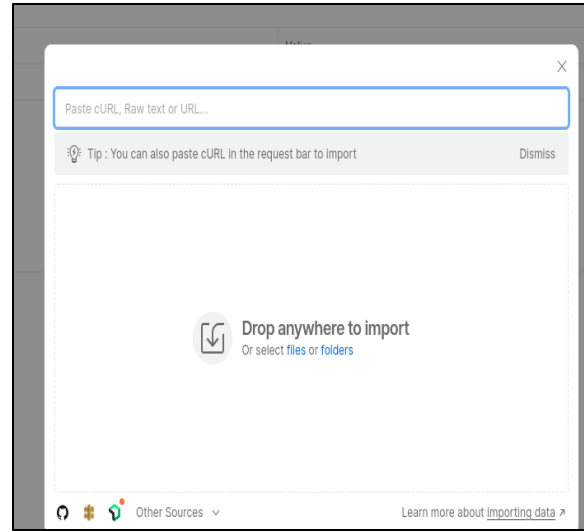
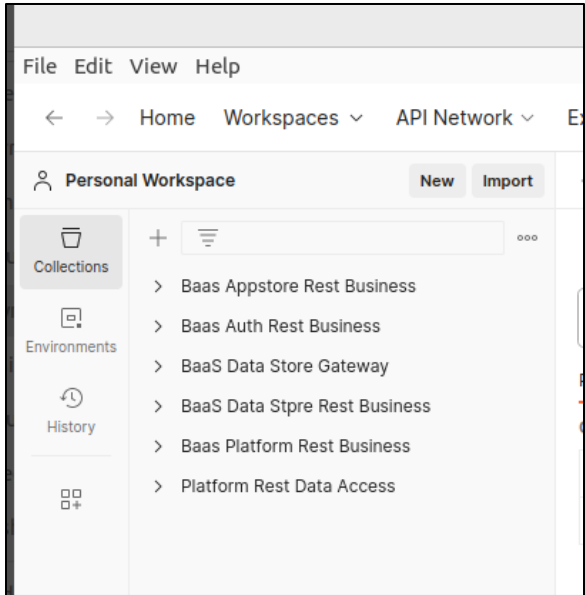


- View Test Results



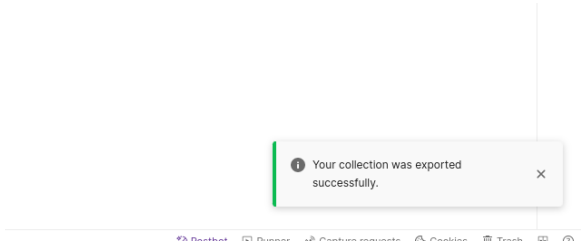
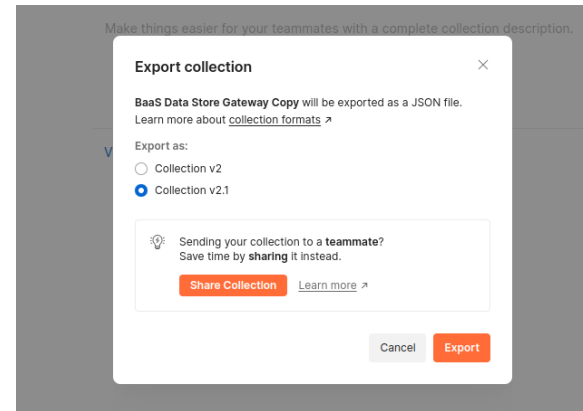
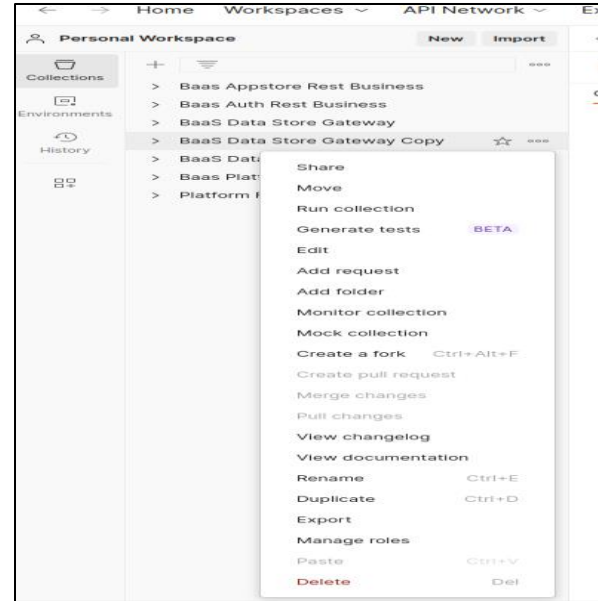
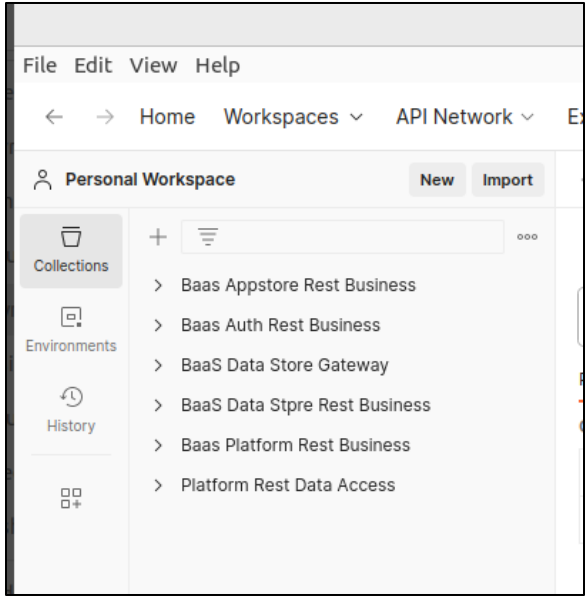
How to Import a collection in the Postman

- Open Postman
- Go to Collections
- Click "Import" Button
- Choose File or Raw Text
- Select Import Source
- Click "Import"
- Verify Imported Collection
- Open Imported Collection



How to Export a collection in the Postman

- Open Postman
- Go to Collections
- Select the Collection
- Click "Export" Button
- Choose Export Format
- Click "Export"
- Save the File
- Verify Export



Happy Coding