REST API Testing Procedure

API testing is a type of software testing that involves testing application programming interfaces (APIs) directly and as part of integration testing. API testing is used to determine if they meet expectations for functionality, reliability, performance, and security.

Web API testing

1. sending requests to the server and analyzing responses to ensure correct behavior
2. Verifying functionality, reliability, performance and security of APIs
3. Enhancing security by checking authentication, authorization, and data protection.

Prerequisites

- **Postman** - Postman is a popular API client that allows to design, test, and document APIs. It provides a user-friendly interface for creating requests, assertions, and collections of API tests.
- **Node.js** - Postman's runtime is based on Node.js. All validations are written using node.js.
- **newman** - newman is a command-line collection runner for Postman. It allows to run and automate Postman collections directly from the command line or integrate them into your CI/CD pipelines.

    a. install Newman globally using npm: ***npm install -g newman***
    b. Check version of newman: ***newman -v***
    c. Install newman html report: ***npm install -g newman-htmlextra***

Ensure that newman version installed is 6.2.1 or more than this.

```
C:\>newman --version
6.2.1
```

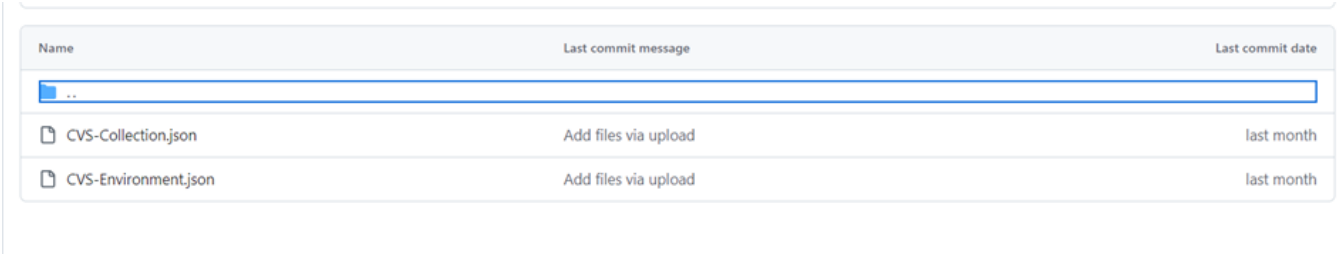
Step-by-step guide

Step 1: Download Collection and Environment Files

1. **Access the Repository**
   Navigate to the repository on Github.

2. **Locate the Collection and Environment Files**

    a. Look for files with extensions like .json for collections and environments.
    b. Common naming conventions might include collection.json and environment.json.



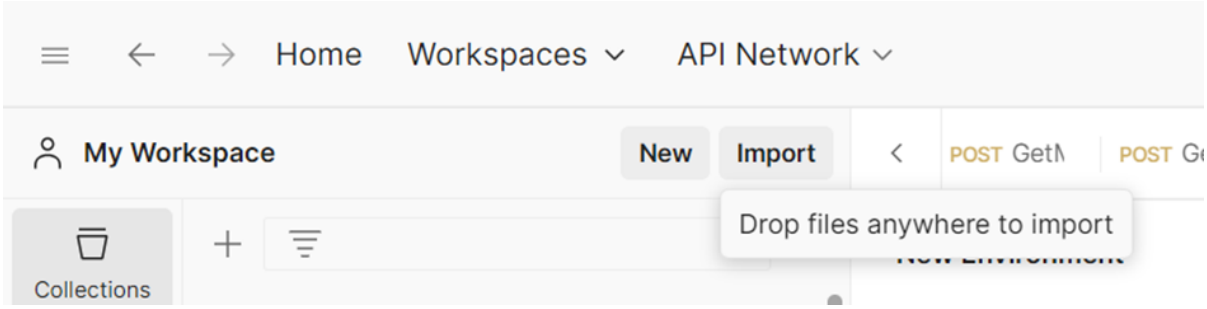
3. **Download the Files.**

   Click on each file and use the "Download" option or right-click and select "Save link as..." to save them to your local machine.

Step 2: Import Files into Postman

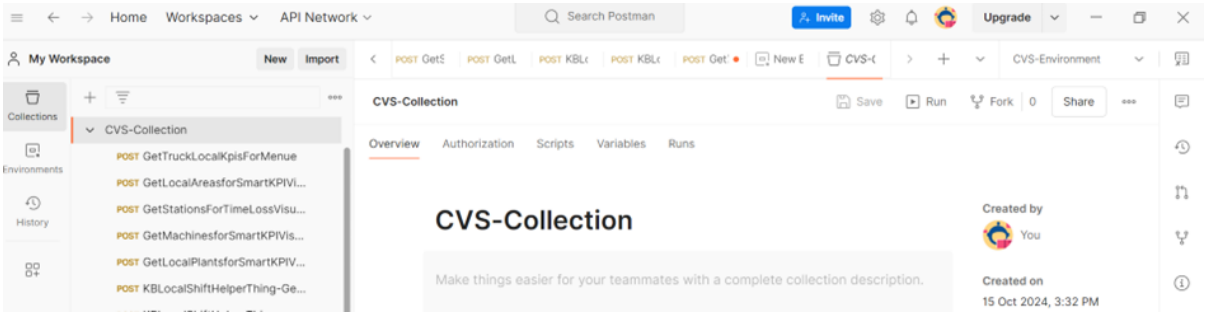
1. **Open Postman**
   Launch the Postman application on your device.

2. **Import Collection**
   Click on the "Import" button located in the top-left corner of the Postman interface. In the import dialog, select the Files tab. Click on "Choose Files" and navigate to where you saved the collection file (collection.json). Select the file and click "Open."



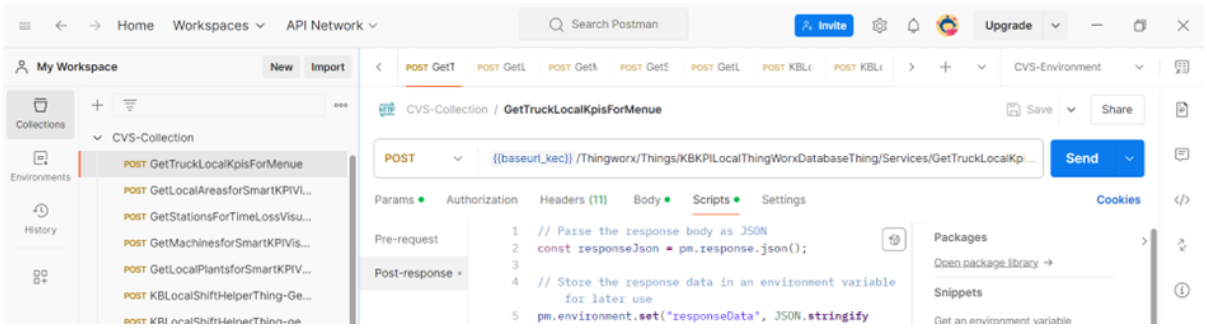
Step 3: Modify Collection and Environment

1. **Open the Imported Collection**
   a. In the left sidebar, find and click on the newly imported collection to expand its details.



2. **Add/Modify Requests**

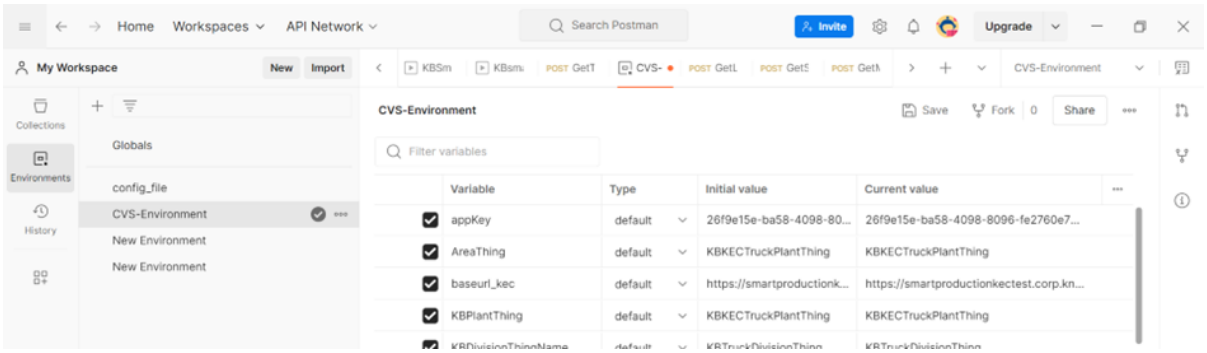
   a. Click on individual requests within the collection to view their details.



   b. You can change request methods (GET, POST, etc.), modify URLs, add headers, or change the body of the requests as needed.

3. **Add/Modify the Environment**

   a. Click on the "Environments" tab in the left sidebar.
   b. Select the imported environment to view its variables.
   c. Edit variable names, values, or add new variables as required.



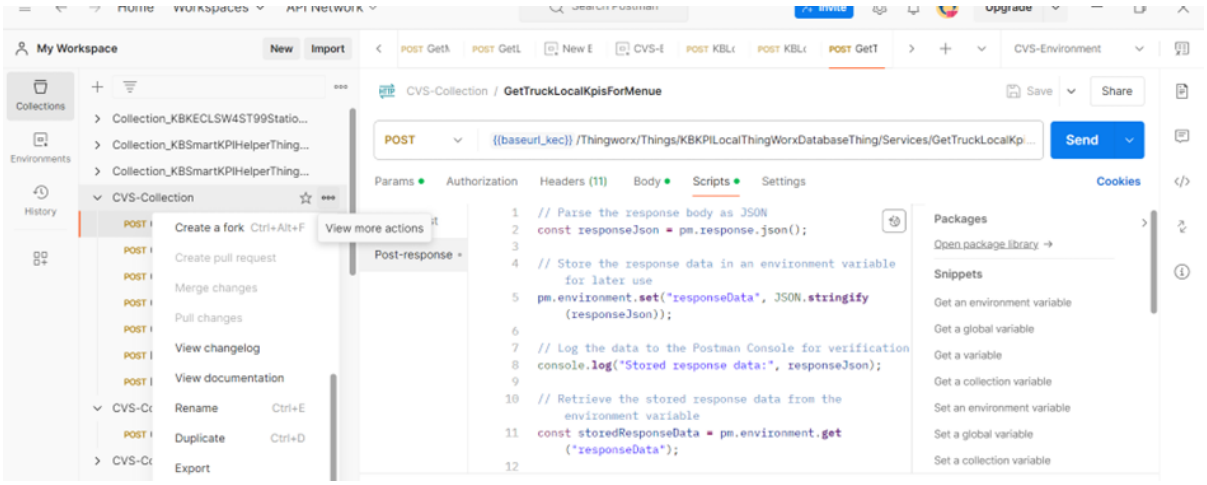
Note: 1. The variables should be named in format <varname>_<APIname>

4. **Save Changes**

    a. Ensure to save any changes made to requests or environment variables by clicking on the "Save" button.

Step 4: *Export Modified Postman Collection and Environment*

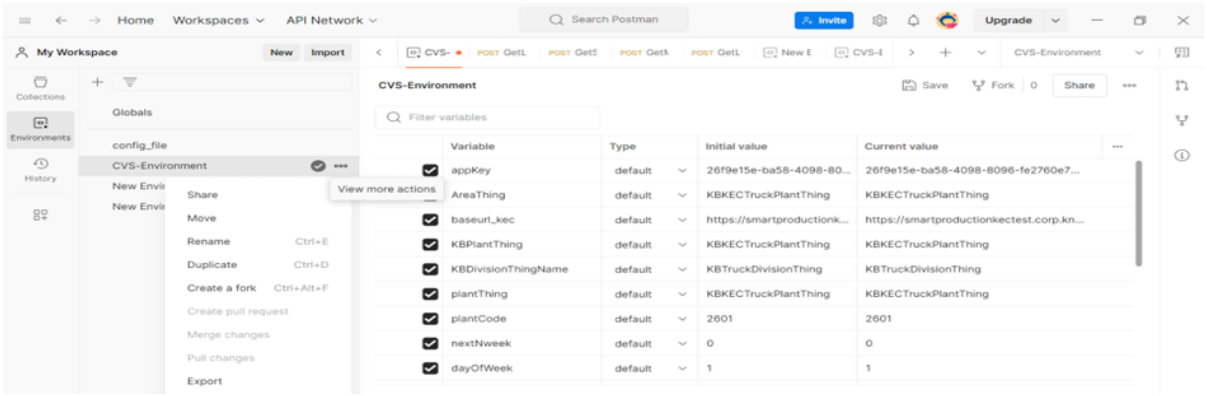
1. **Export Collection:**
    a. Click on the collection name in the left sidebar.
    b. Click on the three dots (more actions) next to the collection name.
    c. Select **Export**.



    d. Choose the format (usually Collection v2.1 is preferred) and save the file.

2. **Export Environment:**

    a. Click on the environment in the left sidebar.
    b. Click on the three dots next to the environment name.
    c. Select **Export**



    d. Save the environment file.

Step 5: *Run Collection*

Run your collection with environment

newman run your_collection.json -e your_environment.json -r htmlextra --insecure

```
C:\KBData\newman_collections>newman run CVS-Collection.json -e CVS-Environment.json -r htmlextra --insecure
```

Step 6: *Commit to Github*

Add Files: Add the exported Postman collection and environment JSON files to the repository.

Commit Changes: Provide a meaningful commit message that describes the updates

Use scripts to add logic and tests to Postman requests | Postman Learning Center

Edit and set environment variables in Postman | Postman Learning Center

Related articles

- How to resolve Github certificate issue (unable to get local issuer certificate) on DevOps server
- REST API Testing Procedure
- How to Migrate / Switch Kepware Productive Server