API Testing Flow in REAL Companies

1. **Requirement**
    1. API Documentation (If they are Good Company)
        1. Google Doc
        2. Google Excel
        3. Confluence
        4. Swagger UI - httpbin.org/
        5. Postman Documentation - apidocs.imgur.com/#c85c9dfc-7487-4de2-9ecd-66f727cf3139
        6. HTML page (URL)
            1. docs.github.com/en/rest/repos?apiVersion=2022-11-28
            2. restful-booker.herokuapp.com/apidoc/index.html#api-Booking-DeleteBooking
            3. apidocs.imgur.com/
2. **No Documentation**
    1. Create your Own Documentation
        1. UI is ready ->
            1. docs.google.com/document/d/1OVxw2b9XHutE_trecdlP1lvWrn5u_yw3/edit#heading=h.gjdgxs
            2.
        2. No UI - **Curl** Way (Mech) - Create your Own Req. -
            1. Ask the **Developer for the Curl Request**
            2. via AI Generation - chatgpt.com/share/67c29482-9290-8009-a8f5-e551ebc2874d
            3. you.com/search?q=Generate+a+API+Documentation+in+this+format+for+this+request+%0A%0AExample+%3A++%22Objective+-+This+API+is...&cid=c1_a9f29fa2-3c63-4878-9826-340d6e1c72c4&tbm=youchat

3. **Test Plan (Get it reviewed by the QA Lead/Product Manager)**
    1. Google Doc
    2. JIRA - Zephyr
4. **Test Cases**
    1. docs.google.com/spreadsheets/d/1EH1UJ9Qezgx_aZ0xim3KcVJUCEeR7A-7/edit?usp=sharing&ouid=104755920778477387077&rtpof=true&sd=true
    2. LIVE Zephyr - > prnt.sc/1LO-8X5Bq0EV
5. **Test Environment - Setup**
    1. Max - **DevOps (Person) -> SRE, Deployment Application)**
    2. Dev
    3. QA (Browser Stack, Tekion)
6. **Test Execution**
    1. Manual
        1. **Excel Sheet** -> Execute Your Testcase where? API (No UI) - Tools
            1. Postman - Big Daddy
            2. Swagger, SOAP UI, Advanced Rest Client, Soapy, Katalon Studio, JMeter...
    2. Automation -> (Test Case Stable)
        1. Postman - Automation -> We don't use it. (Maintenance is Very High, Code Reusability Issues, Duplicate Issues).
        2. **Rest Assured** - Lib Java API Automation

What to Test in an API?

- Verify the Parse of JSON or XML response (i.e. Body and Response)
- HTTP Status Code
- Verify the Headers response with Negative or Invalid headers
- Max/ Min Key values
- Missing Keys
- Response Time
- Cookies
- JSON Schema - Parameters
- Authorization
- CRUD Operations
- Empty Updates or POST call handling
- Chaining request verification
- Verify how the APIs' error codes are handled
- Duplicate entries, Error message
- Testing Tech - Boundary Values

Extra:

- Test nulls and empty strings; these have a habit of causing issues.
- Test with Unicode characters, ensure they are correctly saved to the database.
- Test mandatory and optional parameters, in particular, test with just mandatory values.
- Ensure correct status codes are returned; you will find false positives, e.g. a 500 being returned as a 200.
- Functionality Bugs: The test looks for missing functionality bugs.
- Reliability Bugs: API testing helps identify bugs in integration across different systems.

- Performance Bugs: The tests help determine how much traffic the system can handle before being overloaded, and how to expand infrastructure to meet rising demand at the core, but also are very effective at pinpointing weak points in the API.
- Security Bugs.

Common Bugs in API?

- Missing Keys
- Duplicate Keys
- Incorrect HTTP Code
- HTTP Methods not Handled Properly
- Performance Issues
- Security Issues
- Validation Errors
- API Monitoring Issues

Uses of POSTMAN Tabs

1) **Overview**: Information about the request.

2) **Query Params**: If you have the QP, then you can write it here.

3) **Authorization:** It means permission (Do you need any permission to access this API or not?).

   Types of Auths are:

   a) **Cookie-Based**: Token (which is created by the POST REQUEST) - It is shared with you by another request.

   b) **No Auth**: The Simplest form of authentication, where no credentials are required.

   c) **Basic Auth:** Credentials, we need to access the API, you need a username and password in base64 encoding.

      Example: Username: admin, Password: password123 by the base64(decode) YWRtaW46cGFzc3dvcmQxMjM=]

      Limitations of Basic Auth:

      It is not very secure (E.g. PUT Request - Restful Booker)

      i) **First Option**: Add them via the Authorization Tab - Basic Auth and enter your Username and password

      ii) **Second Option:** Headers -> Add Authorization as “Key” and Basic YWRtaW46cGFzc3dvcmQxMjM=] as “Value”

   d) **Digest Auth:**

      - Digest Authentication is more secure than Basic Auth.

      - Having strong encryption like SHA256, MD5.

      - Username and Password are in encrypted form as a Unique Key.

         Example:

            pramod -> bb16fa6160fa1d8a73eba6217844a08b

e) **Bearer Token:**

- It has no information, a random string.

  Example:

  eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c.

- It is in the form of a Unique Key, which is created to access the APIs.

  1. Get this Token from Dev, PM, API documentation, or you can also get it via another API.
  2. Expiry - It depends on the Dev?

f) **JWT Bearer (JSON Web Token):**

- It has some information, like a Unique Token

Example:

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c

{

"sub": "1234567890",

"name": "John Doe",

"iat": 1516239022

}

g) **OAuth 1.0 and OAuth 2.0:** Token -> Imgur APIs example. (LIVE Project)

4) **Headers**: It is a Key and Value pair that you can send.

   Example:

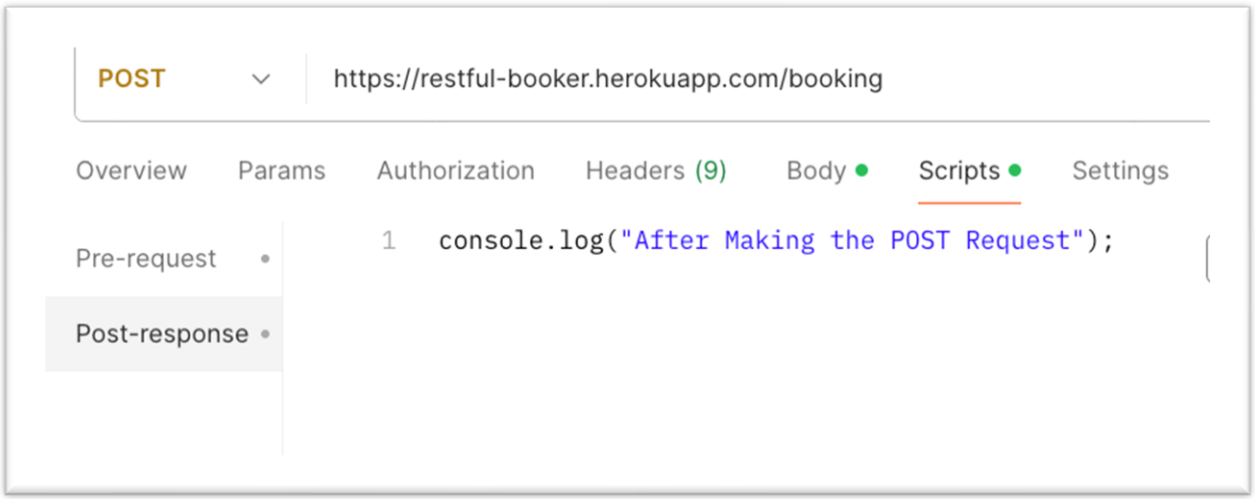
   - content-type: application/json
   - Authorization, Basic, Digest, Token, Cookie, etc

5) **Body:** It is the Payload of the API

   - none – It is Nothing (Just Blank)
   - form-data (HTML, Key and Value pair that you want to send to the Server)
   - raw – Text, JavaScript, JSON, HTML, XML
   - binary – Send file like pdf, jpg, jpeg, png, docx, etc
   - GraphQL - Query language. To ask the API, give me this

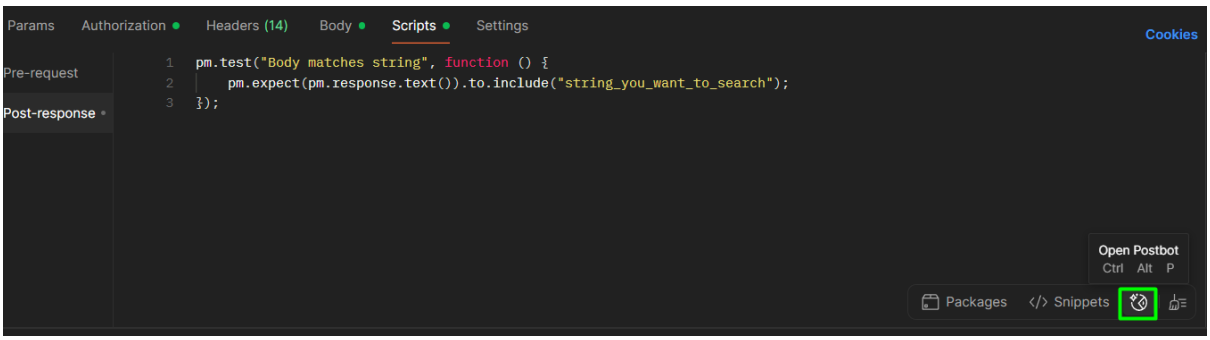
6) **Scripts**

- Pre Script - Before making the request, you can execute a JS Code.
- Post Script - After making the request, you can execute a JS Code

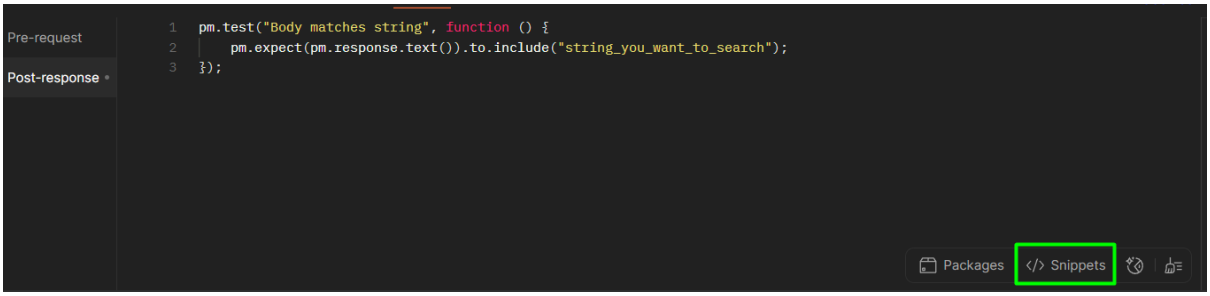


How to write test cases for all the Requests that we have imported in Postman?

- Super Easy - By using Postman's own AI or any AI Bot

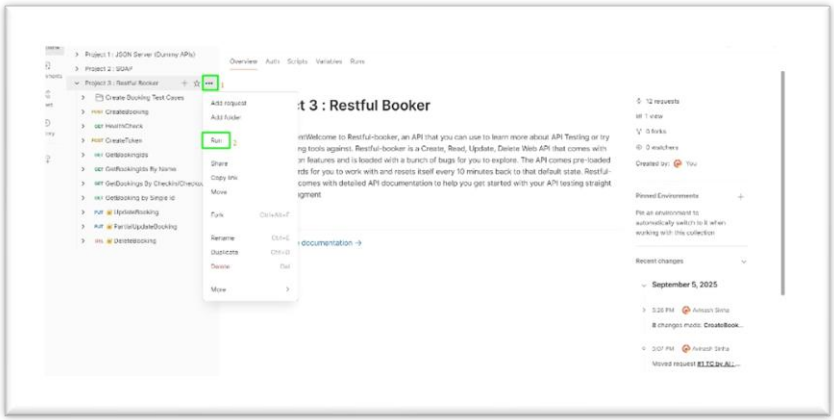


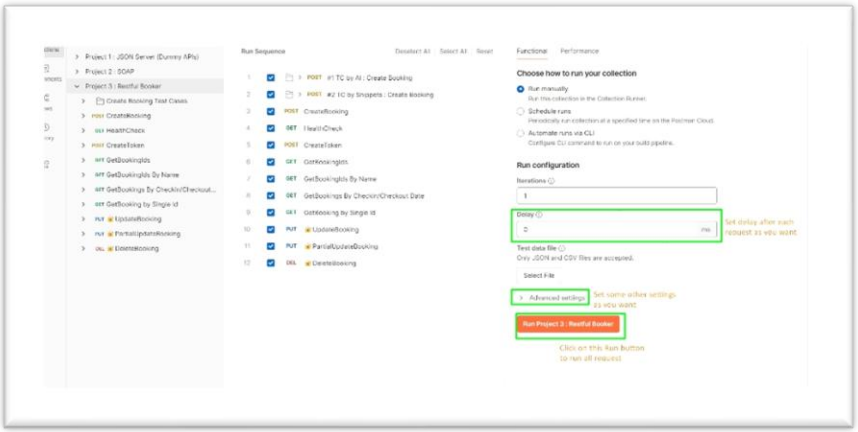
- Easy Level – Snippets

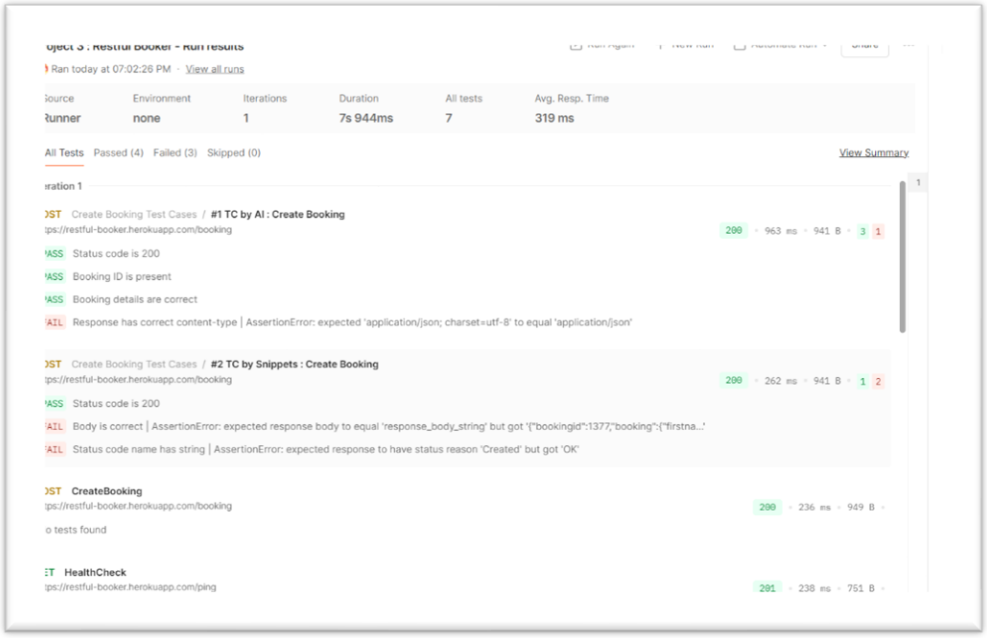


- Medium Level - With JavaScript

Run all the Collection Requests in Postman







Another way to run the Postman Collection Requests

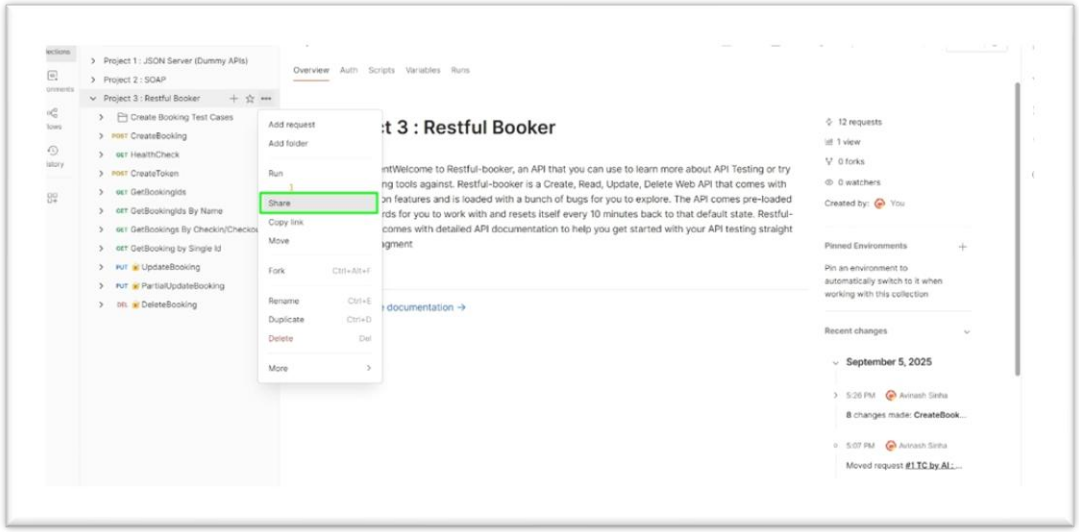
Newman

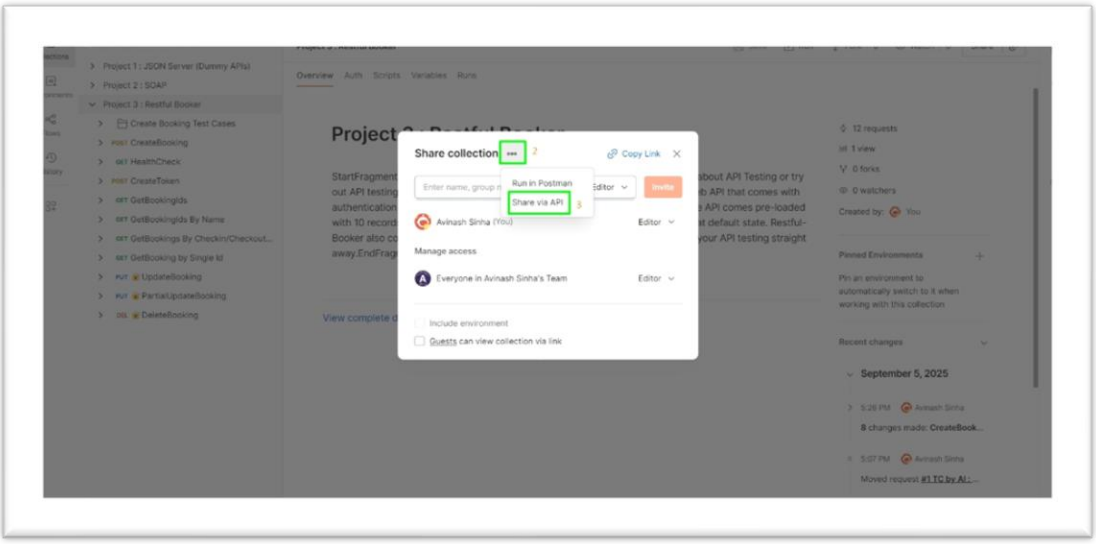
- The Newman command-line tool to run the Postman collections.
- It will generate a beautiful HTML report.

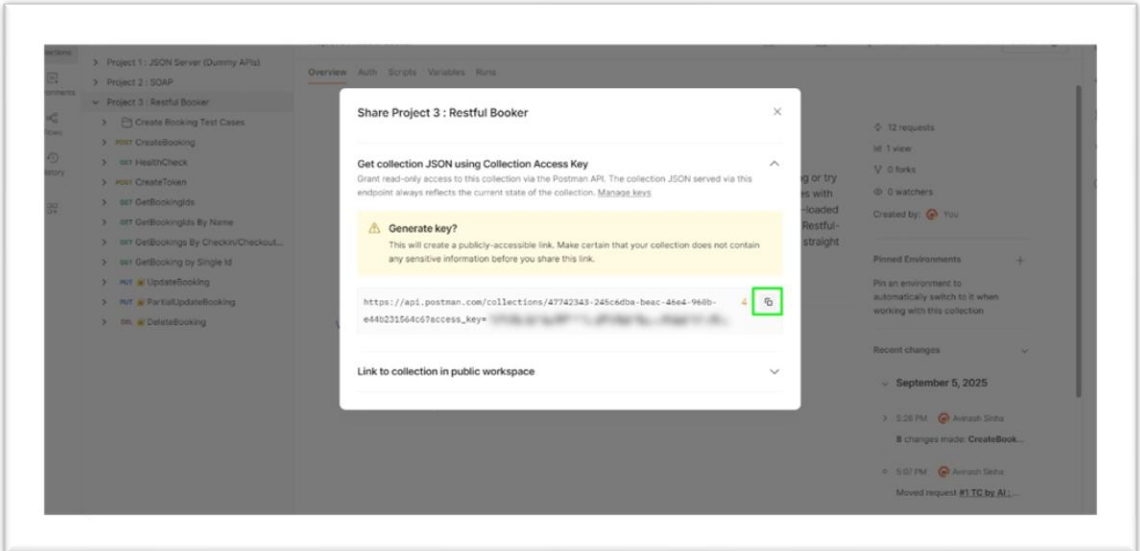
How to use the Newman?

1. Install Node.js
2. Restart your system (after install)
3. Open the CMD and verify that node and npm are installed:
    1. node --version
    2. npm --version
    3. npm -g install newman
    4. newman --version
    5. npm -g install newman-reporter-htmlextra
4. Run the collection
    1. newman run <link> -r cli,htmlextra

P.S.: IF YOU HAVE A WINDOWS SYSTEM, AND YOUR NAME OF USER IS LIKE THIS "ABC XYZ", THEN NEWMAN WILL NOT WORK.


Project 1 : JSON Server (Dummy APIs)
Project 2 : SOAP
Project 3 : Restful Booker
Create Booking Test Cases
POST CreateBooking
GET HealthCheck
POST CreateToken
GET GetBookingIds
GET GetBookingIds By Name
GET GetBookings By Checkin/Checkout...
GET GetBooking by Single Id
PUT UpdateBooking
PUT PartialUpdateBooking
DEL DeleteBooking
Overview Auth Scripts Variables Runs
Add request
Add folder
Run
Share
Copy link
Move
Fork Ctrl+Alt+F
Rename Ctrl+E
Duplicate Ctrl+D
Delete Del
More
t 3 : Restful Booker
12 requests
1 view
0 forks
0 watchers
Created by: You
Pinned Environments
Pin an environment to automatically switch to it when working with this collection
Recent changes
September 5, 2025
5:26 PM Avinash Sinha
8 changes made: CreateBook...
5:07 PM Avinash Sinha
Moved request #1 TC by AI : ...


Share collection
Copy Link
Run in Postman
Share via API
Editor
Invite
Avinash Sinha (You)
Manage access
Everyone in Avinash Sinha's Team
Include environment
Guests can view collection via link


Share Project 3 : Restful Booker
Get collection JSON using Collection Access Key
Grant read-only access to this collection via the Postman API. The collection JSON served via this endpoint always reflects the current state of the collection. Manage keys
Generate key?
This will create a publicly-accessible link. Make certain that your collection does not contain any sensitive information before you share this link.
https://api.postman.com/collections/47742343-245c6dba-beac-46e4-968b-e44b231564c6?access_key=
Link to collection in public workspace

```
C:\WINDOWS\system32\cmd.

C:\Users\HP\Documents>newman run https://api.postman.com/collections/47742343-245c6dba-beac-46e4-960b-e44b231564c6?acces
s_key=PMAT-01K4CWTC6KG4F3VA6HQS6V2WC7 -r cli,htmlextra
```