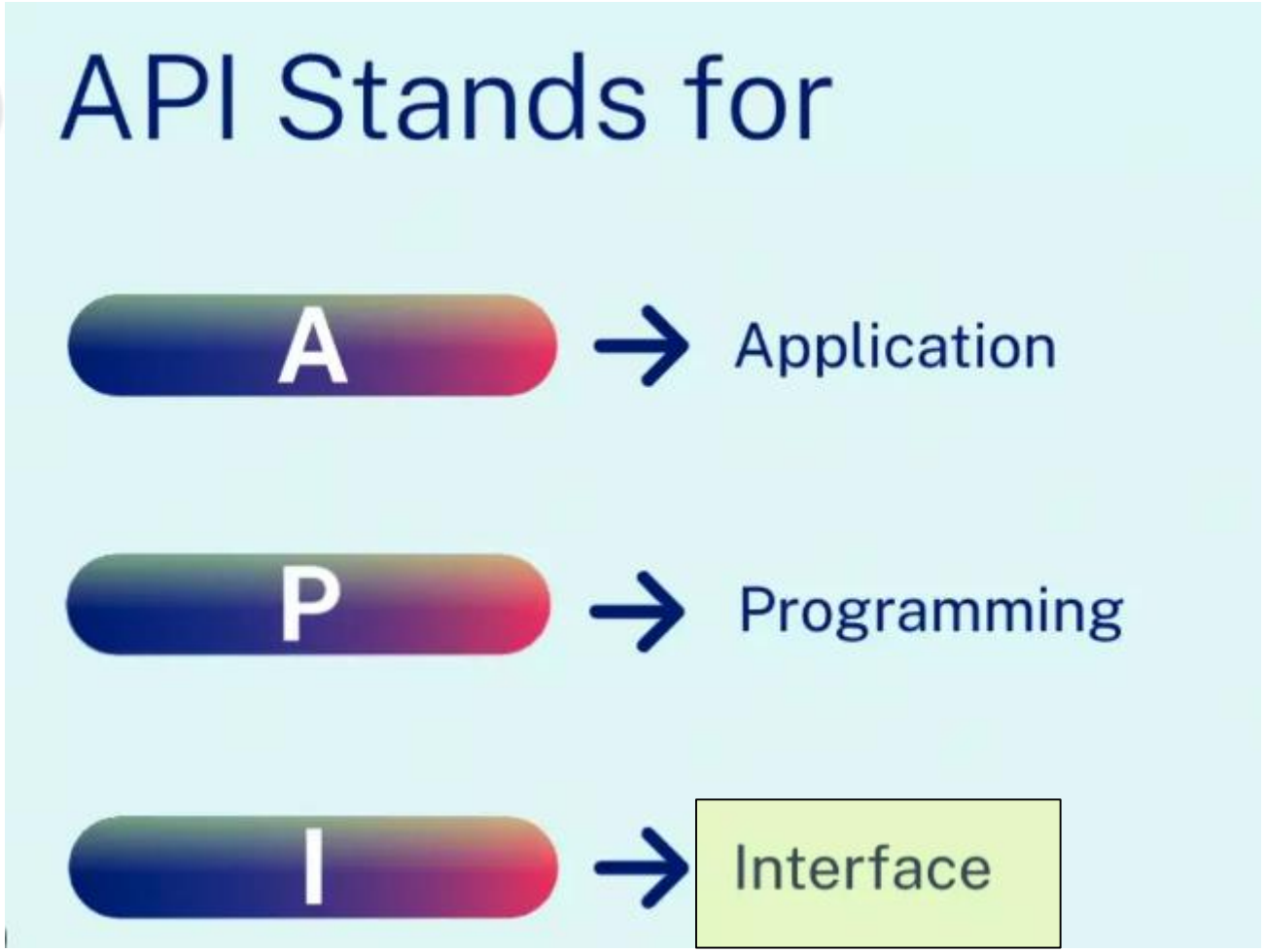
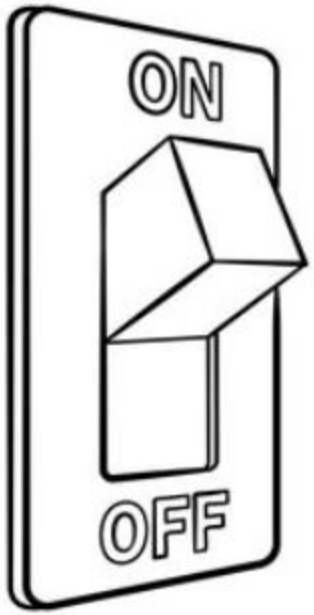
בדיקות ממשקים
Web API Testing
Using Postman

What is an API



מה זה API ?

What is an Interface

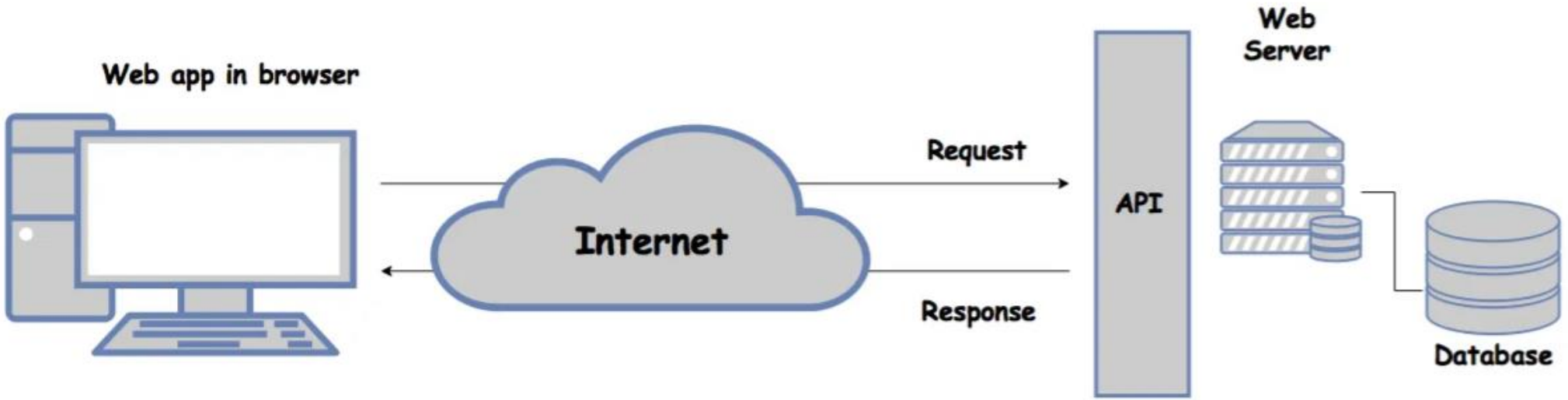


מה זה ממשק ?

What is an Interface

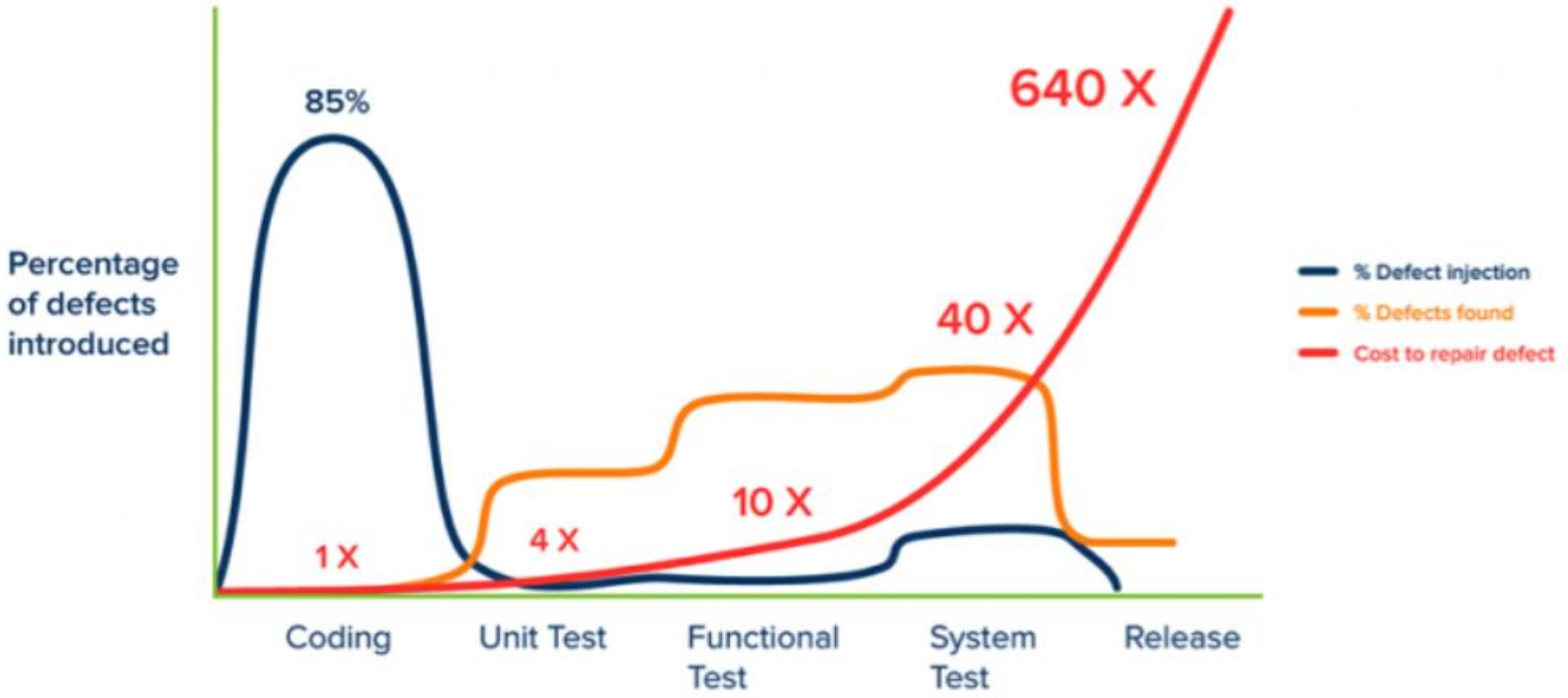
מה זה ממשק ?

What is Web API



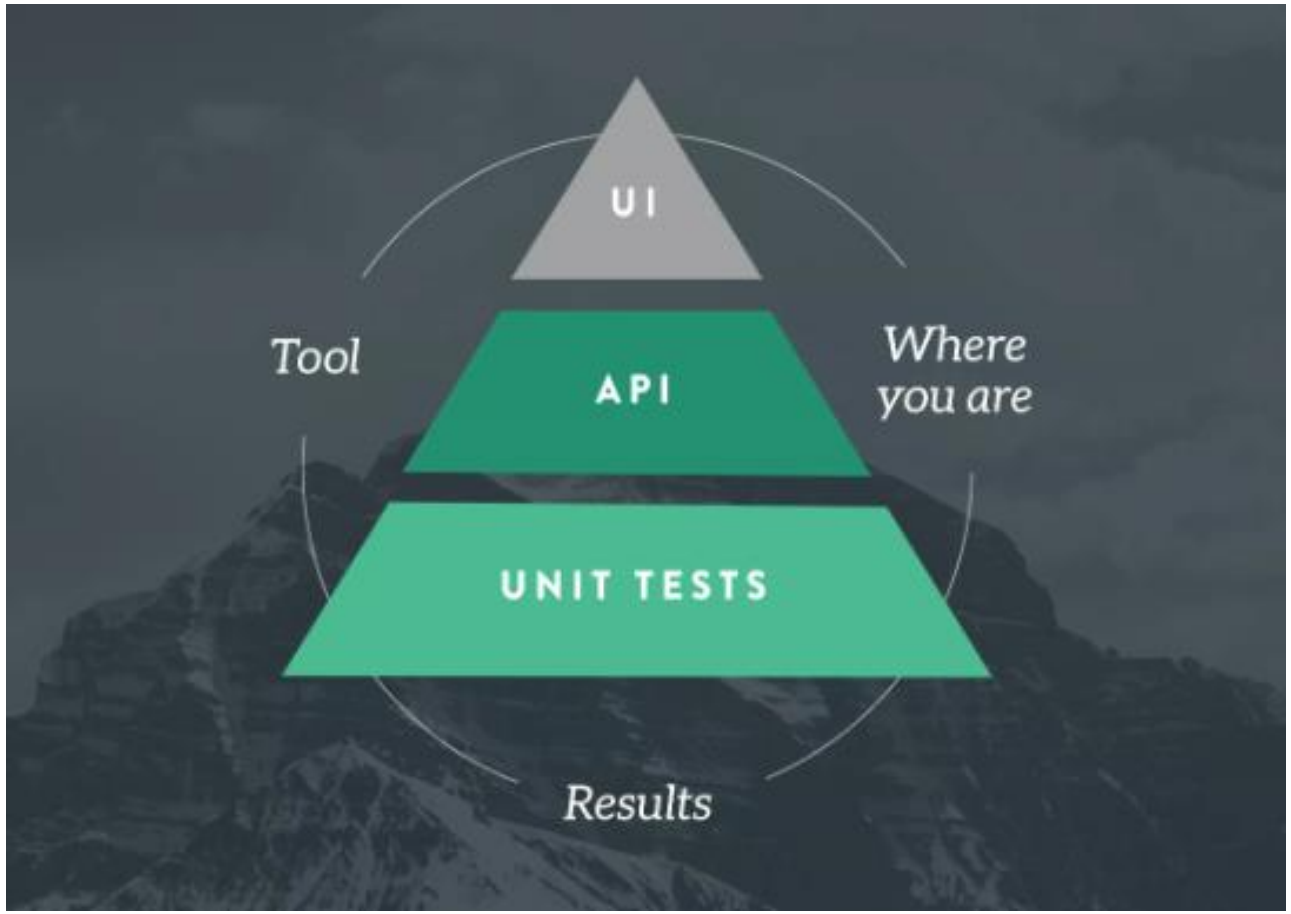
מה זה ממשק אינטרנט ?

Why starting QA early in SDLC



פערים בעלויות ובהשפעה של תקלות בשלבים שונים של המוצר ככל שנמצא את הבאגים רחוק יותר משלב כתיבת הקוד המחיר של התיקון יהיה גבוה יותר בצורה אקספוננציאלית

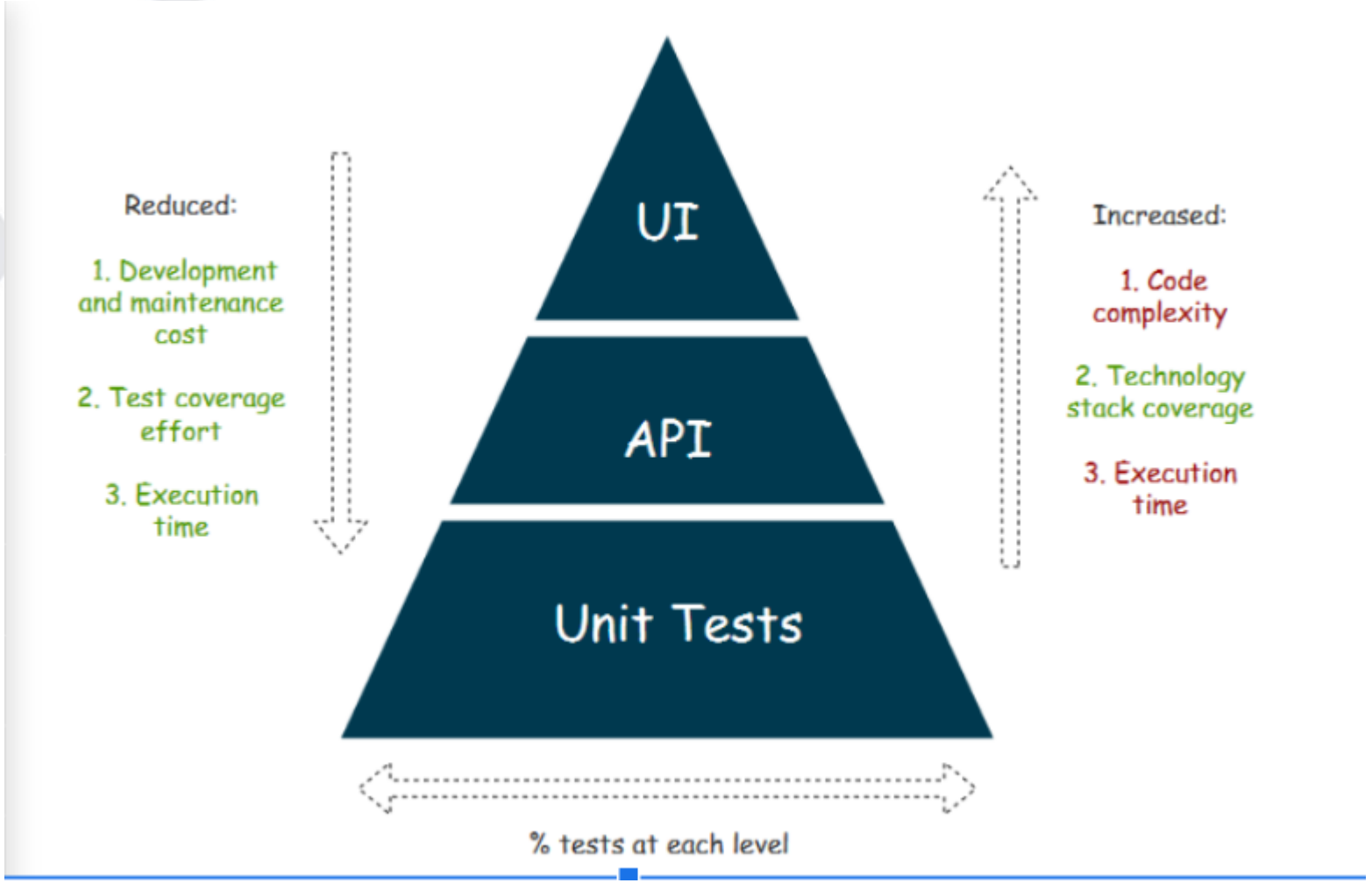
Pyramid of testing



נכיר את מודל פירמידת הבדיקות שכל איש בדיקות צריך להכיר

Why use API in Testing is better

פירמידת הבדיקות :בקצה העליון בדיקות קצה לקצה (E2E)בדיקות יקרות בזמן הרצה ובזמן יצירה ,לעומת סוגי הבדיקות האחרות: בדיקות ממשקים מהירות וזולות יותר ,כלי מסייע לכתיבת הבדיקות ממשקים ,הוא הפוסטמן בדיקות יחידה הם הבדיקות הקצרות והיעילות ביותר , אך הן מוגבלות מבחינת היקף הבדיקה.

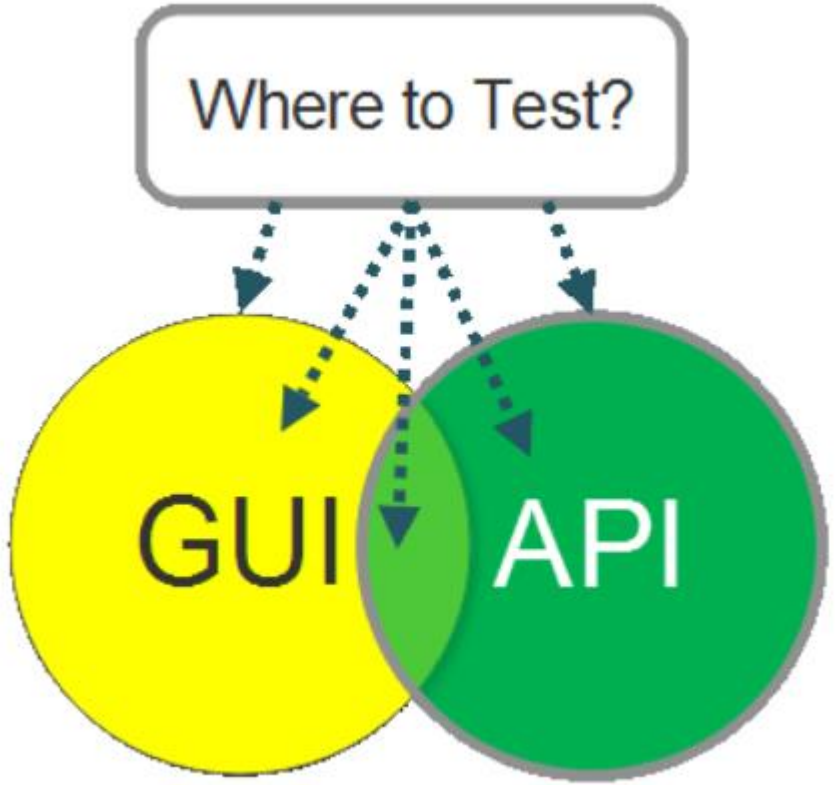


Why get into automation

Manual Testing	Automation Testing
The simplest low-level type, during each a QA runs all tests manually, without using any helpful software.	A QA uses a special program/tool for running tests, already existing or written specially for the project.
A time-consuming process.	Time-saving, particularly, because a QA can re-run the same tests numerous times
Can be repetitive and boring.	Helps to avoid repetitive tasks as a QA "delegates" them to a computer.
Is suitable for almost any software product.	Is suitable only for stable systems and used mainly for regression. Some types of testing (research, ad-hoc, etc.) cannot be automated.
Helps to define whether automation testing is possible and necessary.	100% automation is not possible.

שימת דגש על היתרונות של בדיקות אוטומטיות מול בדיקות ידניות

API vs. GUI
Backend vs. Frontend



כאיש בדיקות נרצה להבין איך לכתוב ולהריץ בדיקות קצה וגם בדיקות ממשקים

POSTMAN

API testing

Compared by	API testing	UI testing
Functionality	The functionality of the business logic	The functionality of the UI and the business logic
Fragility	Somewhat fragile	Fragile
Speed of execution	Fast	Slow
Maintenance cost	Medium	High
Who writes them	Developers and testers	Testers

יתרונות וחסרונות של בדיקת ממשקים מול בדיקת ממשק גרפי, בדיקה של צד שרת מול בדיקת צד שרת וצד לקוח.

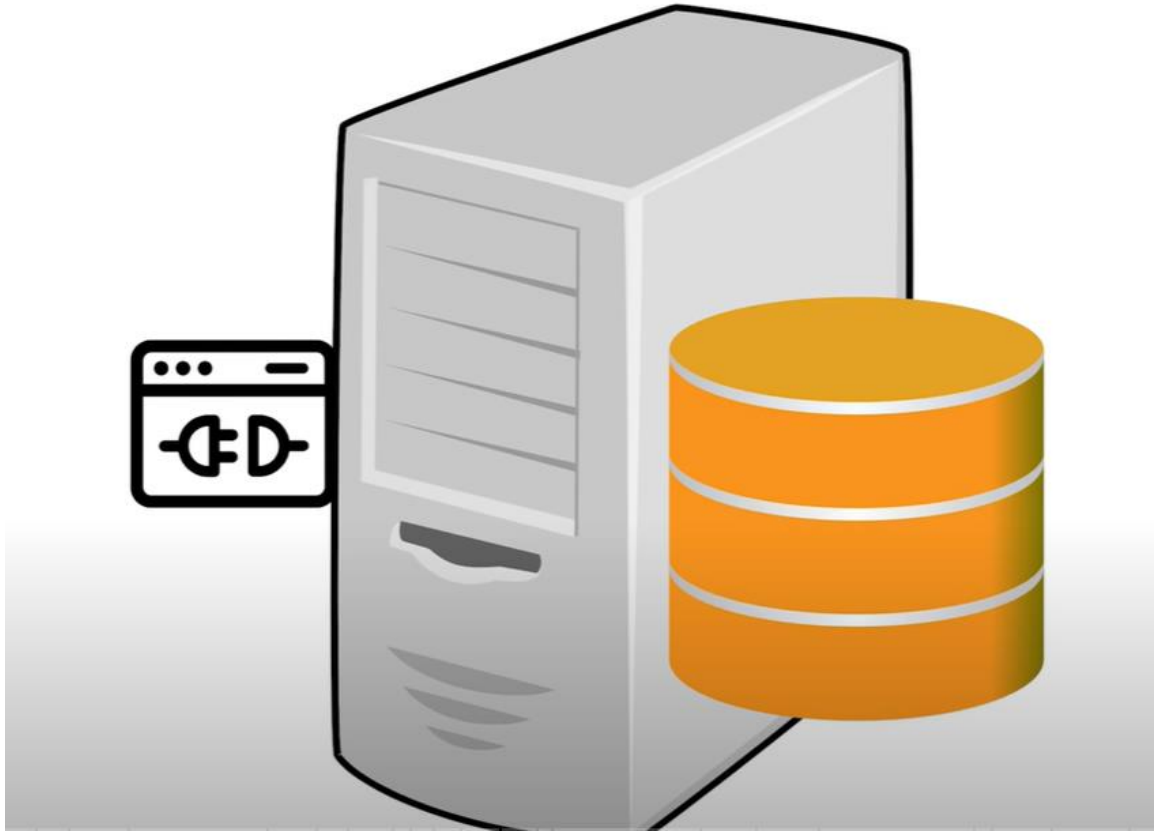
Concept of API

API = an interface to data

- they are designed to be used from programs
- no friendly interface

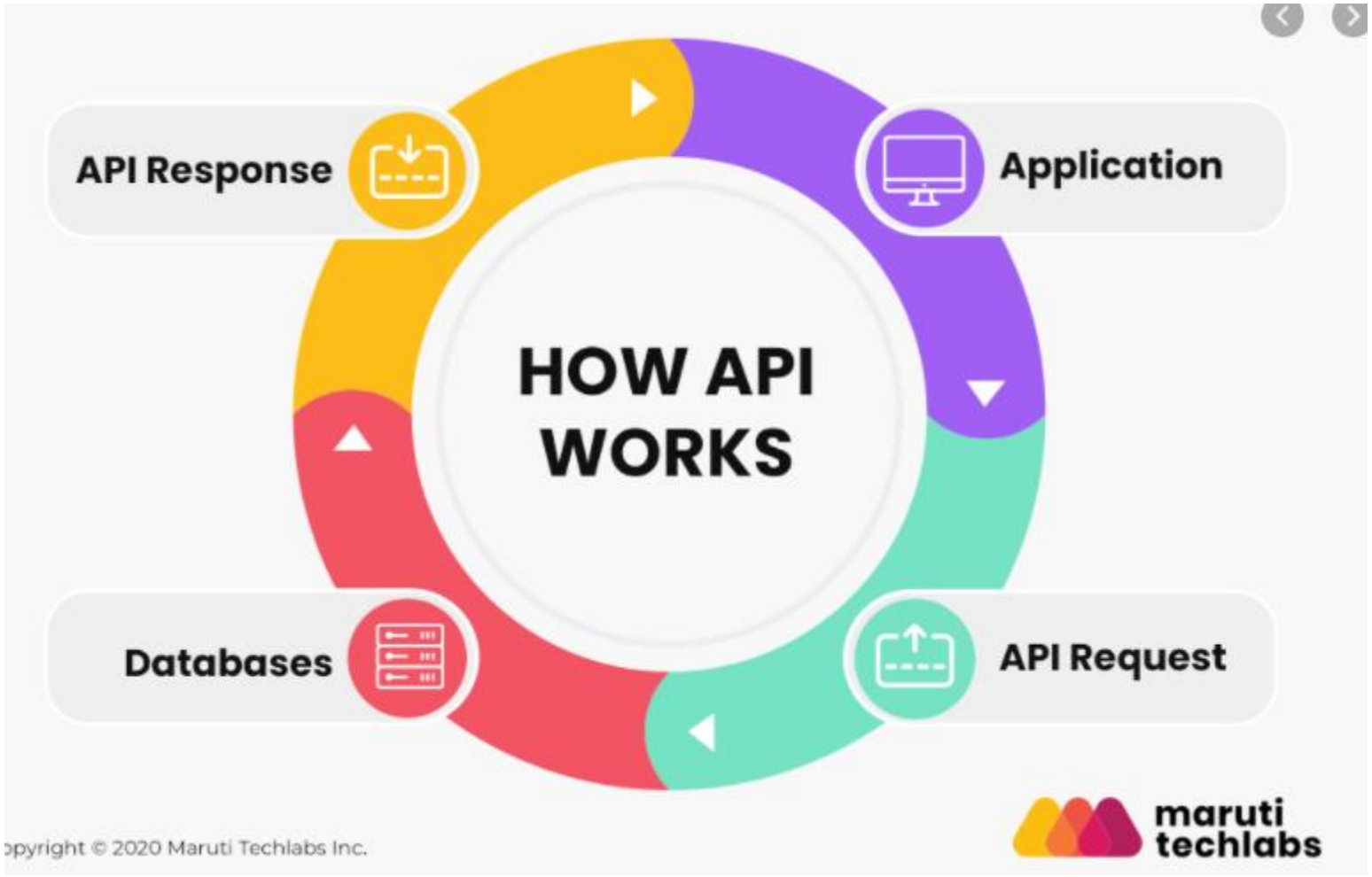
בדיקות ממשקים הוא בדיקות של הממשק האינטרנטי (Web)עם הנתונים שיושבים בצד השרת (Backend)

API



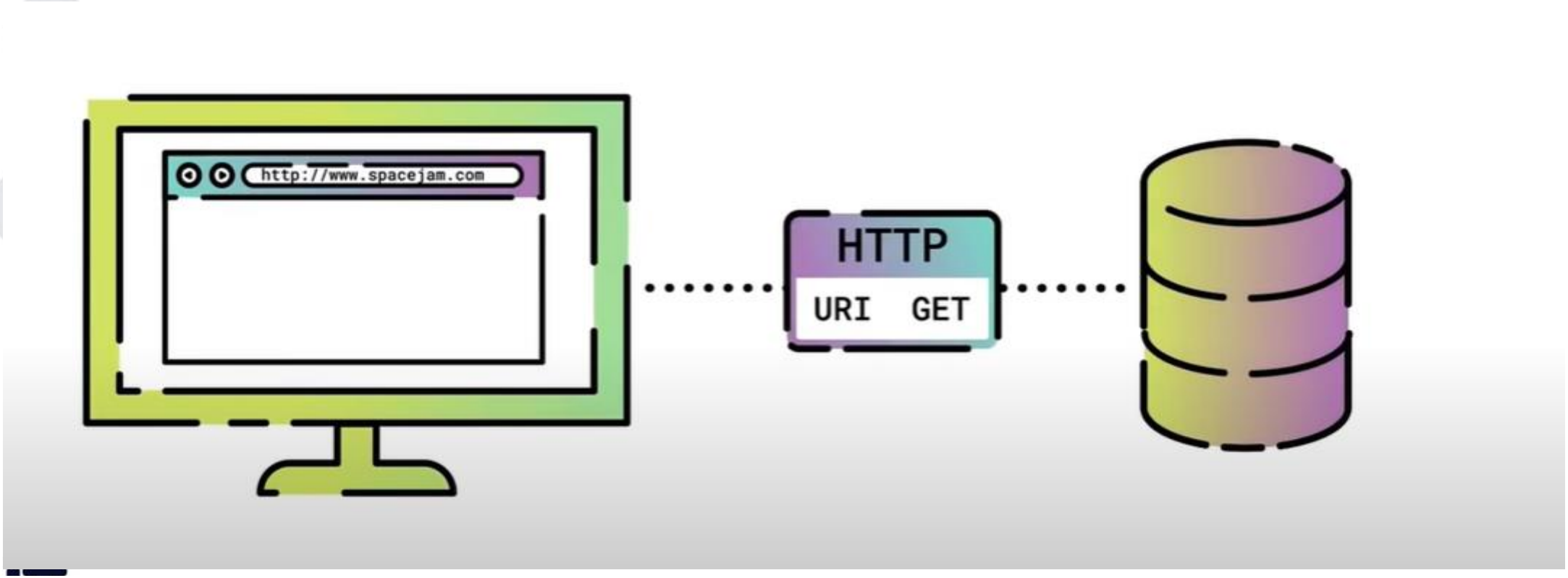
הממשק האינטרנטי מאפשר לצד הלקוח לגשת למסדי הנתונים בצד השרת ,הממשק האינטרנטי הוא מערכת המפותחת על ידי מפתחי צד שרת.
חשוב להבין את המושג ממשק ,לפני שמסבירים את המושג Web API

POSTMAN



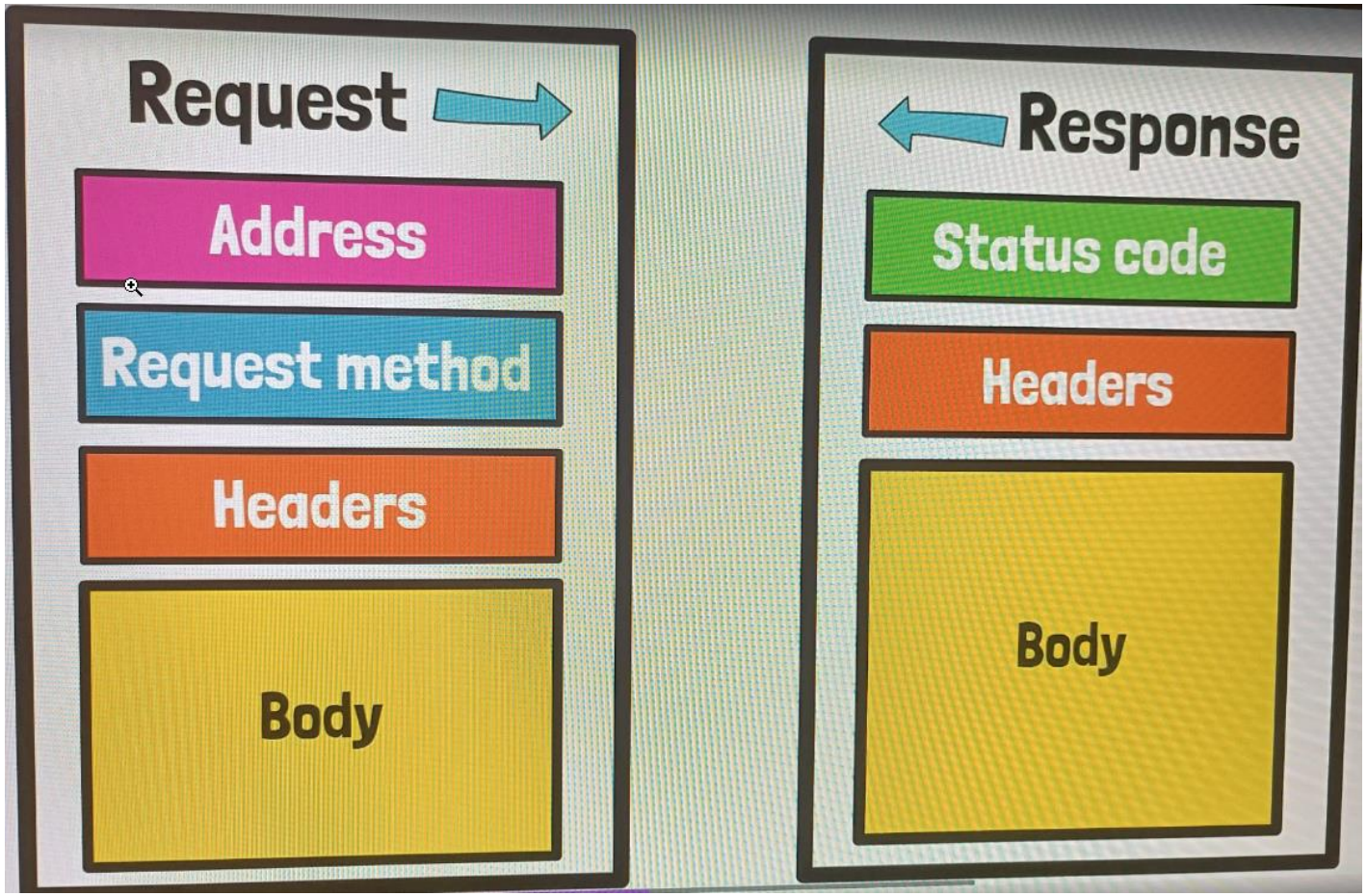
איך ממשק Webעובד :בקשה מצד השרת תענה בתשובת נתונים וקבצים מצד השרת :
לדוגמה יצירת משתמש חדש במערכת של אתר e-commerce

API



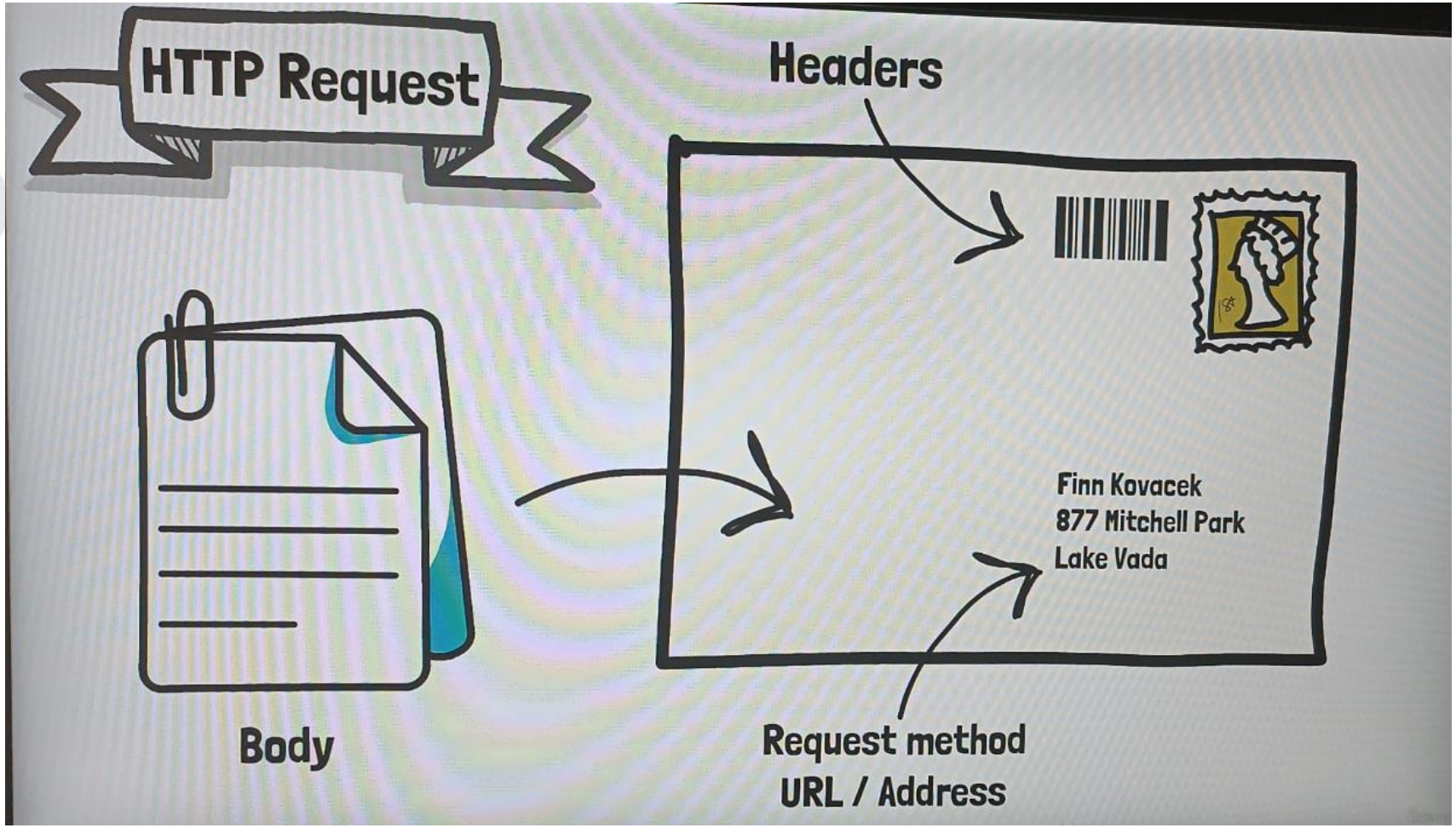
הלקוח שולח בקשה מסוג כלשהו לכיוון השרת ,כאשר רשת האינטרנט משמשת כצינור ובצד השרת נכתבה תשתית התומכת בבקשה זו.

REST HTTP Requests/Response



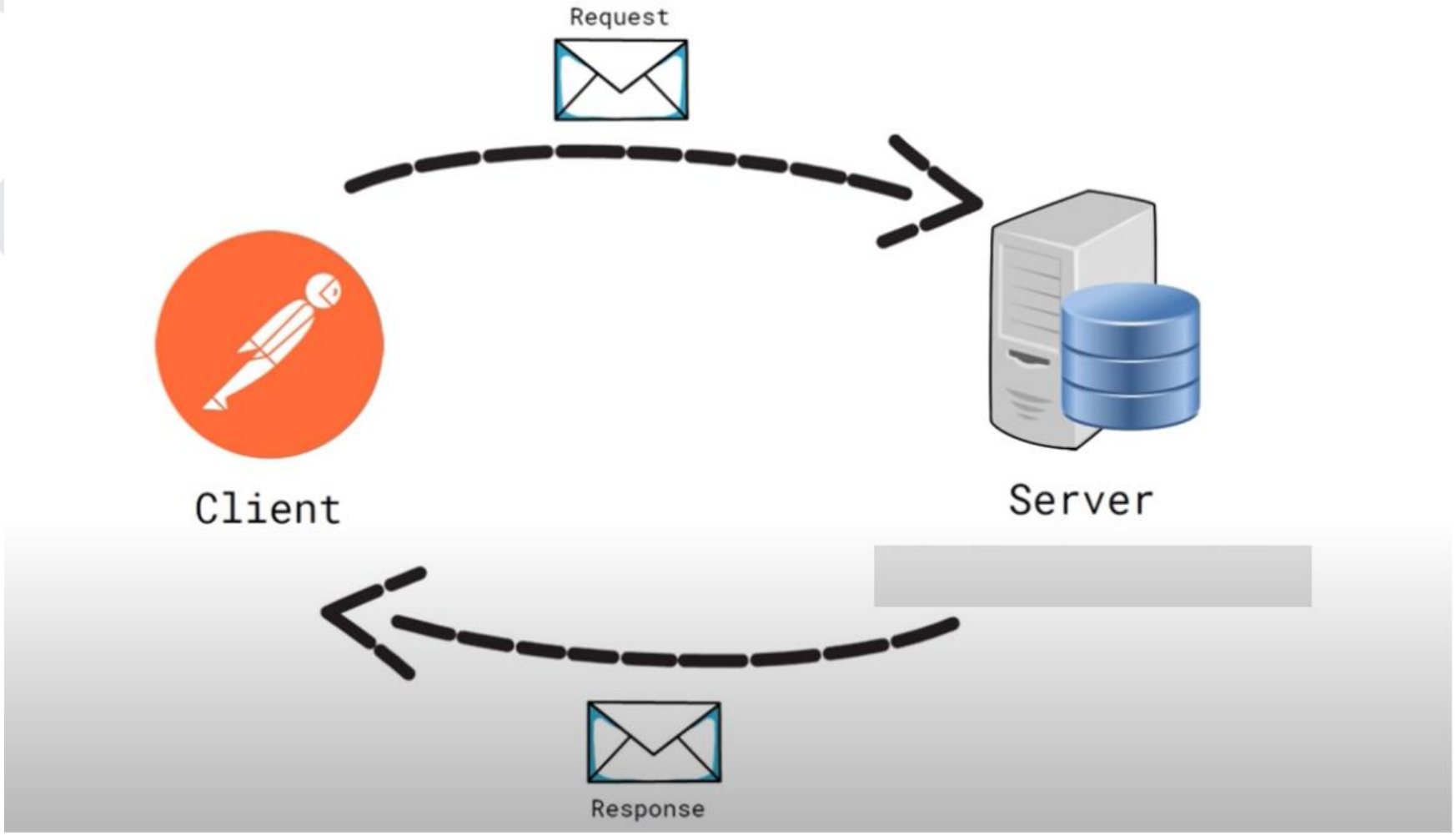
פרוטוקול REST שלרוב משמש אותנו, יש כללים ומבנה
קבועה גם לבקשות מצד הלקוח וגם לתשובות מצד השרת

HTTP Requests/Response



מה המשמעות של Headers: הגדרות מאפייני הבקשה :פורמט הבקשה ולאן לפנות בצד השרת

Postman Practice



לפי סוג הבקשה תענה תשובה מצד השרת :דוגמאות :פורמט המידע החוזרJSON / XML :

API

אפשר להקביל את סוגי הבקשות של פרוטוקול REST לסוגי הפעולות שהכרנו מעולם מושגי מסדי נתונים:SQL :
Create / Update / Read / Delete

HTTP request methods

1. GET:
   - **Description**: The **GET** method requests a representation of a specific resource.
   - **Purpose**: It is used to retrieve data from the server.
   - **Example**: Fetching details of a user profile or retrieving a list of products.
   - **Safety**: Safe (no side effects on the server).
   - **Idempotent**: Yes (multiple identical requests yield the same result).
2. POST:
   - **Description**: The **POST** method submits an entity (data) to a specified resource.
   - **Purpose**: It often causes a change in state on the server (e.g., creating a new resource).
   - **Example**: Creating a new user account or adding an item to a shopping cart.
   - **Safety**: Not safe (may modify server state).
   - **Idempotent**: No (multiple identical requests may create different resources).

HTTP request methods

3. PUT:
    - **Description**: The **PUT** method replaces all current representations of the target resource with the request payload.
    - **Purpose**: Used for updating an existing resource.
    - **Example**: Updating user profile information or modifying an article.
    - **Safety**: Not safe (modifies server state).
    - **Idempotent**: Yes (multiple identical requests yield the same result).
4. DELETE:
    - **Description**: The **DELETE** method deletes the specified resource.
    - **Purpose**: Removes a resource from the server.
    - **Example**: Deleting a user account or removing a comment.
    - **Safety**: Not safe (modifies server state).
    - **Idempotent**: Yes (multiple identical requests yield the same result).
5. PATCH:
    - **Description**: The **PATCH** method applies partial modifications to a resource.
    - **Purpose**: Used for making specific updates to a resource.
    - **Example**: Changing the status of an order or updating specific fields.
    - **Safety**: Not safe (modifies server state).
    - **Idempotent**: No (multiple identical requests may have different effects).

Postman

Copilot

Your personal and company data are protected

Postman is a powerful tool for testing and automating **API requests**. Let's dive into what it offers:

1. **Automated API Testing with Postman**:
   - **Purpose**: Postman allows you to create test suites that can run repeatedly. These tests can include unit tests, functional tests, integration tests, end-to-end tests, regression tests, and more.
   - **Benefits**:
     - **Streamline Development and QA**: Integrate automated testing into your CI/CD pipeline. Test every code push seamlessly.
     - **Scale Easily**: As your programs grow, automated testing helps maintain robustness and reduces the risk of breakage.
     - **Cost-Effective**: Spend less on manual QA, reduce lag time between development and QA, and minimize debugging in production.
   - **How It Works**:
     - **Test Suites**: Organize your requests into Postman Collections. These collections allow you to run and automate a series of requests.
     - **Collaboration**: Postman Workspaces facilitate collaboration among teammates.

פוסטמן ברמת ההגדרה :כלי ניהול של ממשקי ווב REST API :ועוד
הוא כלי המאפשר כתיבה של ממשקים ,ובנוסף להוסיף בדיקות נלוות כדי לוודא את ההתנהגות המצופה.

Postman

- **JavaScript Snippets**: Use built-in JavaScript snippets to test APIs without writing code.

- **Tools**:
  - **Postman App**: Manage and run your test workflows.
  - **Postman Monitoring**: Monitor APIs.
  - **Newman**: Command-line tool for running Postman collections.

2. Writing Tests in Postman:

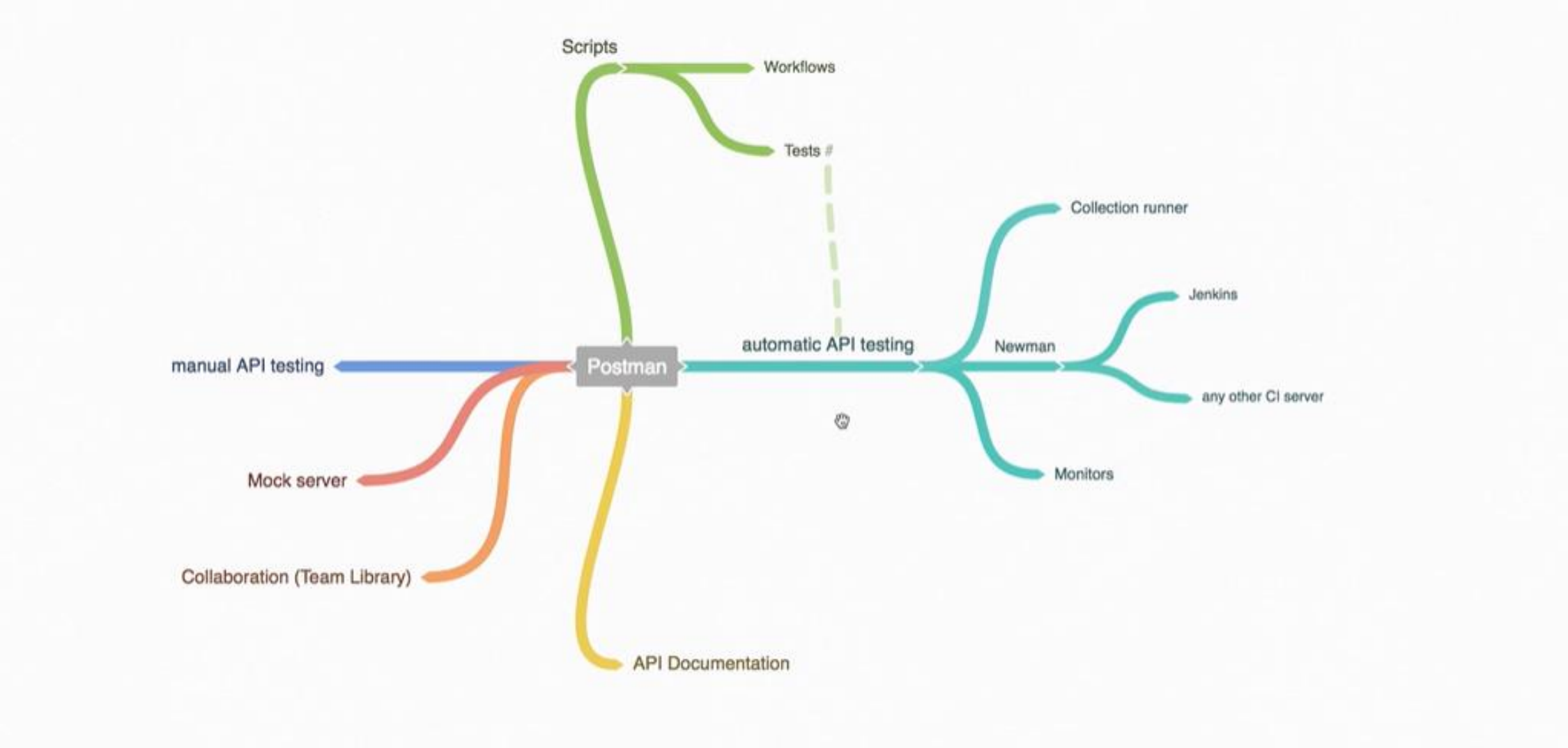
- **Tests Tab**: Add tests to individual requests, collections, or folders. Tests execute after the request runs.
- **JavaScript**: Write test scripts in JavaScript. Validate responses, status codes, and more.
- **Example Test Script**:

JavaScript

```
pm.test("Status code is 200", function () {
    pm.response.to.have.status(200);
});
```

AI-generated code. Review and use carefully. More info on FAQ.

Postman - structure



What is Postman?

What is Postman?

Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster.

```
~/workspace curl --location --request POST 'https://simple-tool-rental-api.glitch.me/orders' --header 'Authorization: Bearer 204df14ba6ad5b12483bfc9acc947dd2fe5e9047c5bb2bf06face06d630b3690' --header 'Content-Type: application/json' --data-raw '{"toolId": 4643, "customerName": "John Doe"}
quote>
```

TMAN

Postman SAAS platform

Postman: Advantages and dis

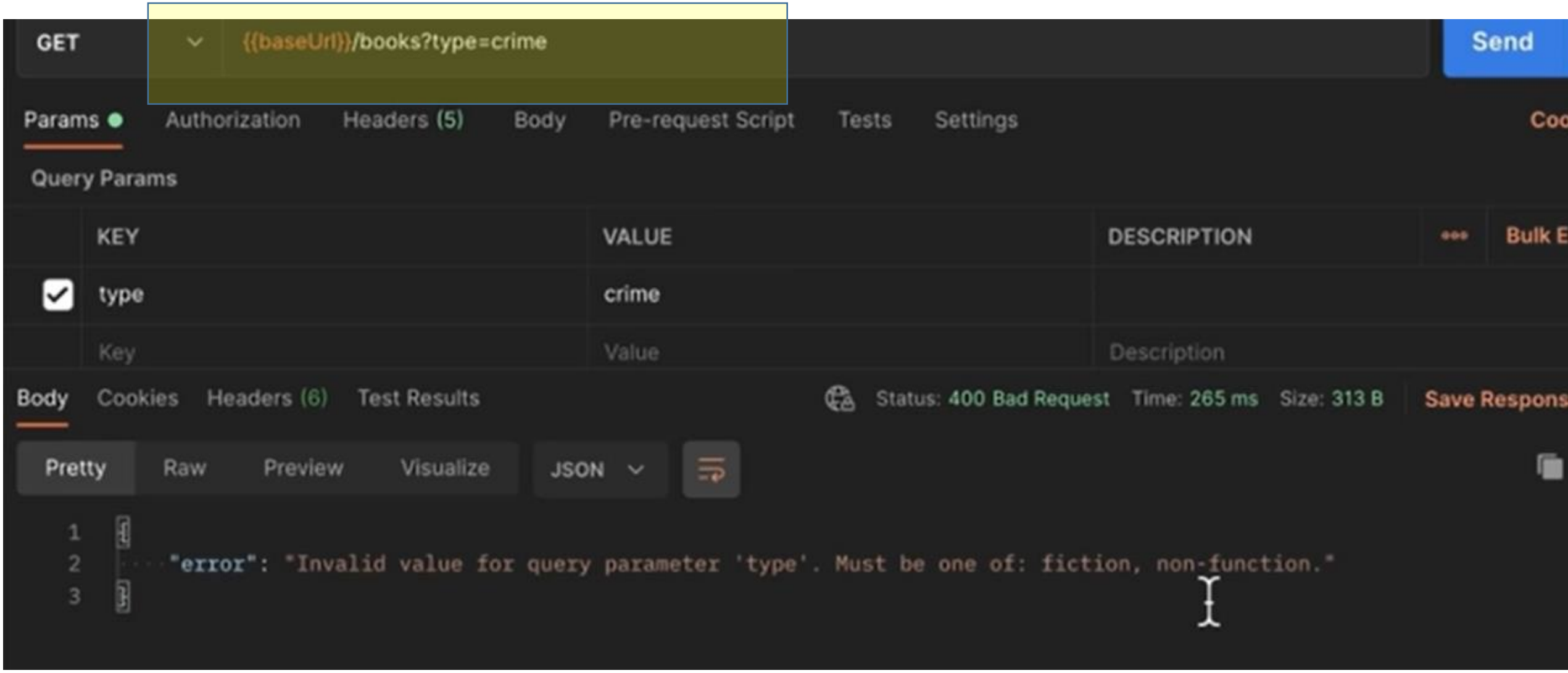
Communicate with APIs

User interaction

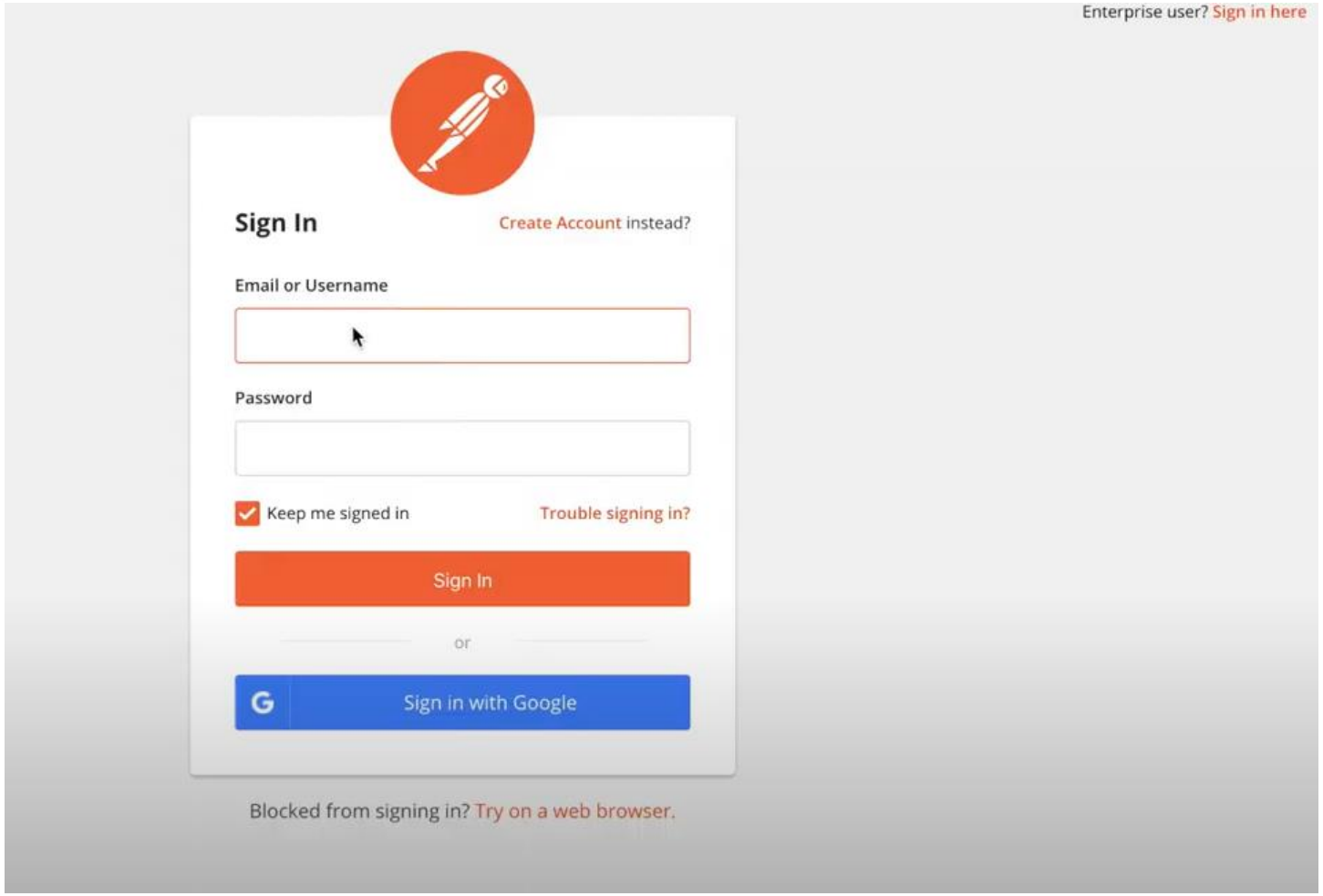
Performance testing

Security testing

Practice Postman



Create Account for Postman

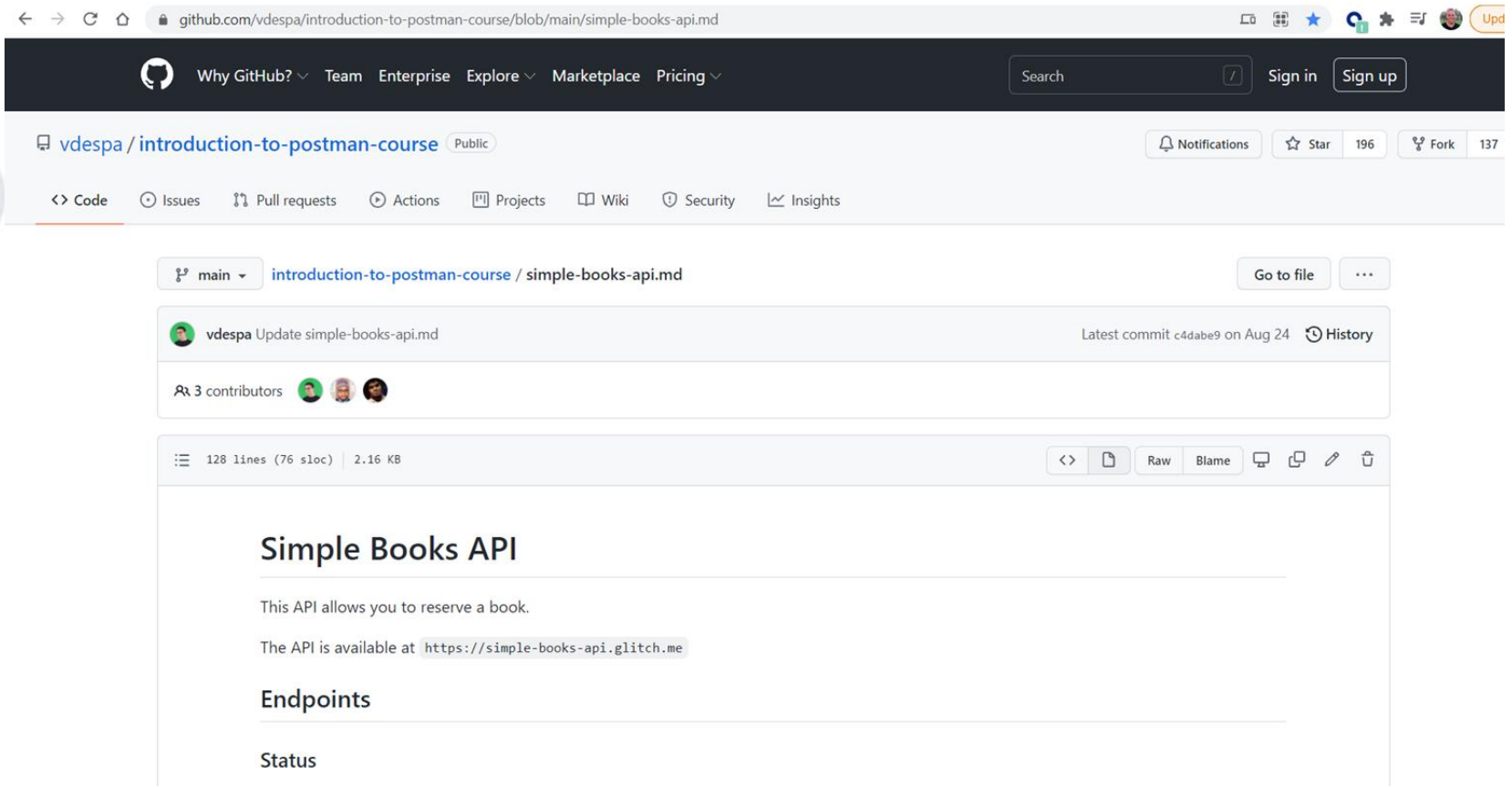


Postman


Your first
request
Let's go ahead and start using an API in Postman.

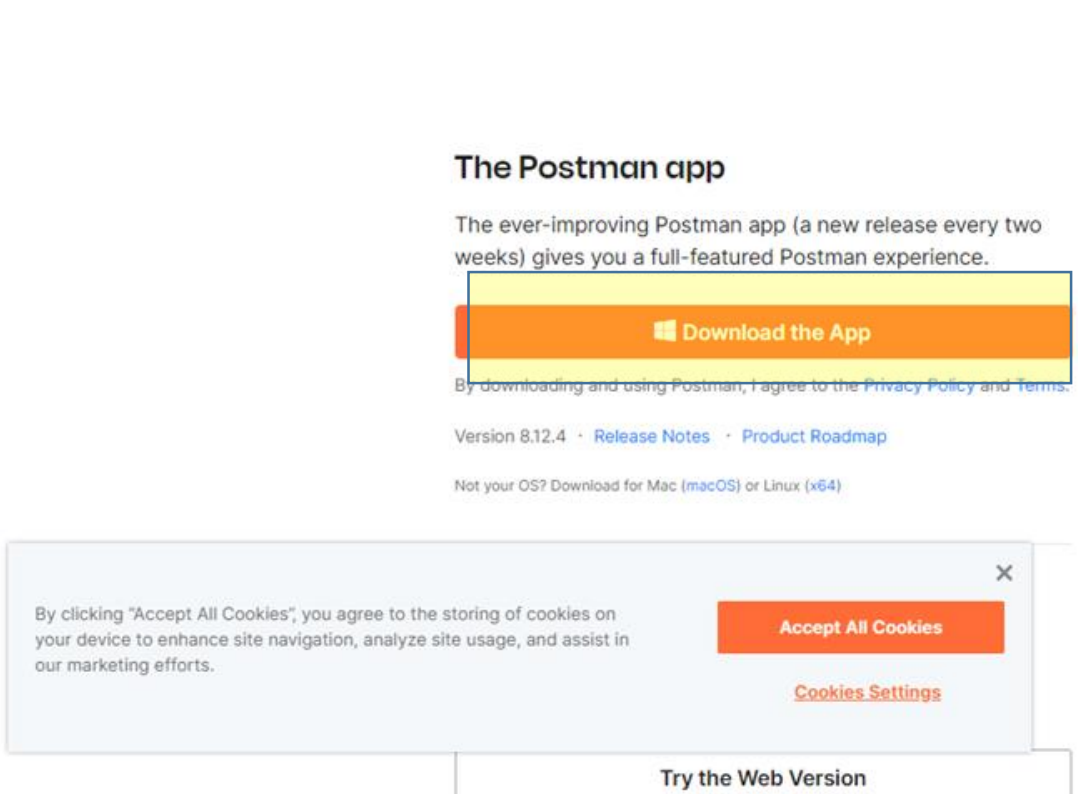
API Documentation

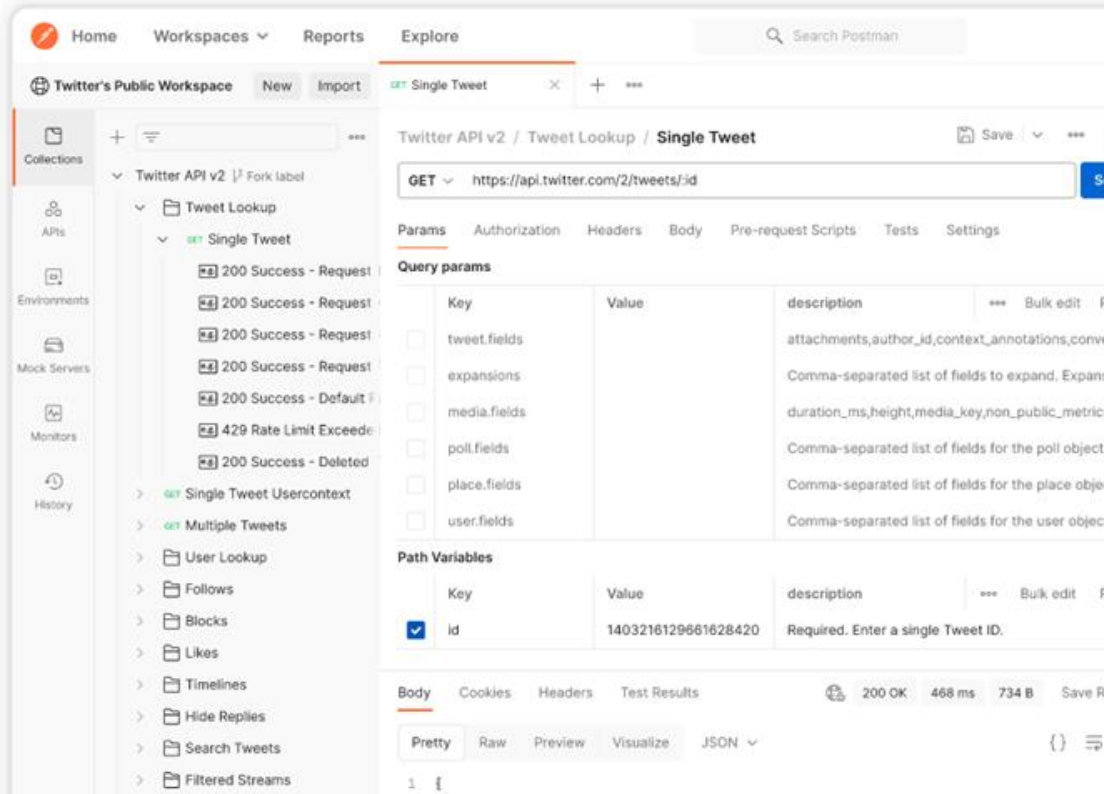
https://github.com/vdespa/introduction-to-postman-course/blob/main/simple-books-api.md



Download Postman

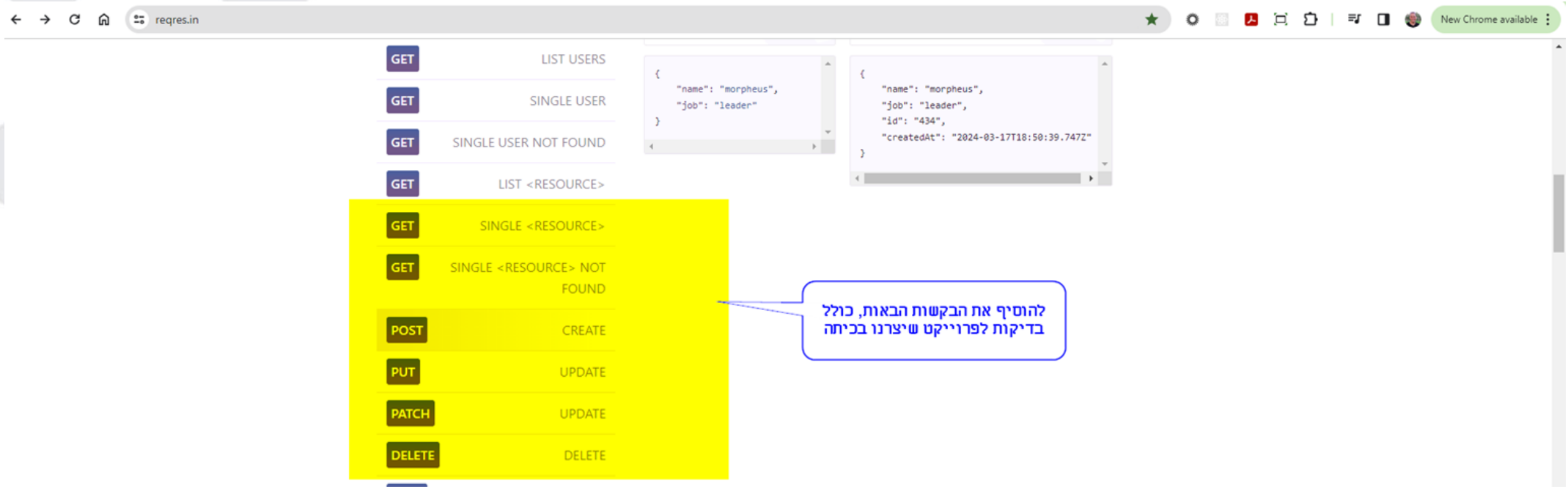
Download the app to quickly get started using the Postman API Platform. Or, if you prefer a browser experience, you can try the new web version of Postman.





Practice Postman

https://reqres.in/



API Documentation

https://github.com/vdespa/Postman-Complete-Guide-API-Testing/blob/main/simple-grocery-store-api.md

Simple Grocery Store API

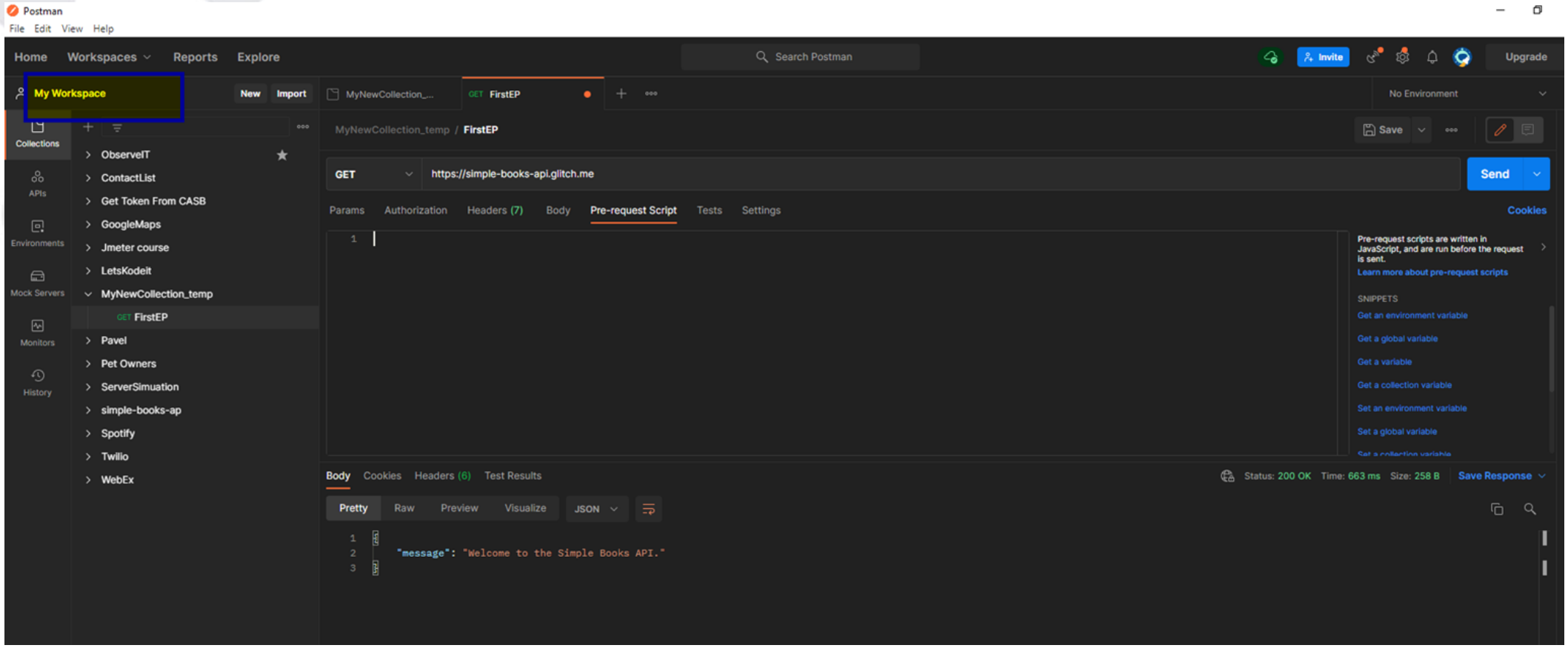
This API allows you to place a grocery order which will be ready for pick-up in the store.

The API is available at `https://simple-grocery-store-api.glitch.me`

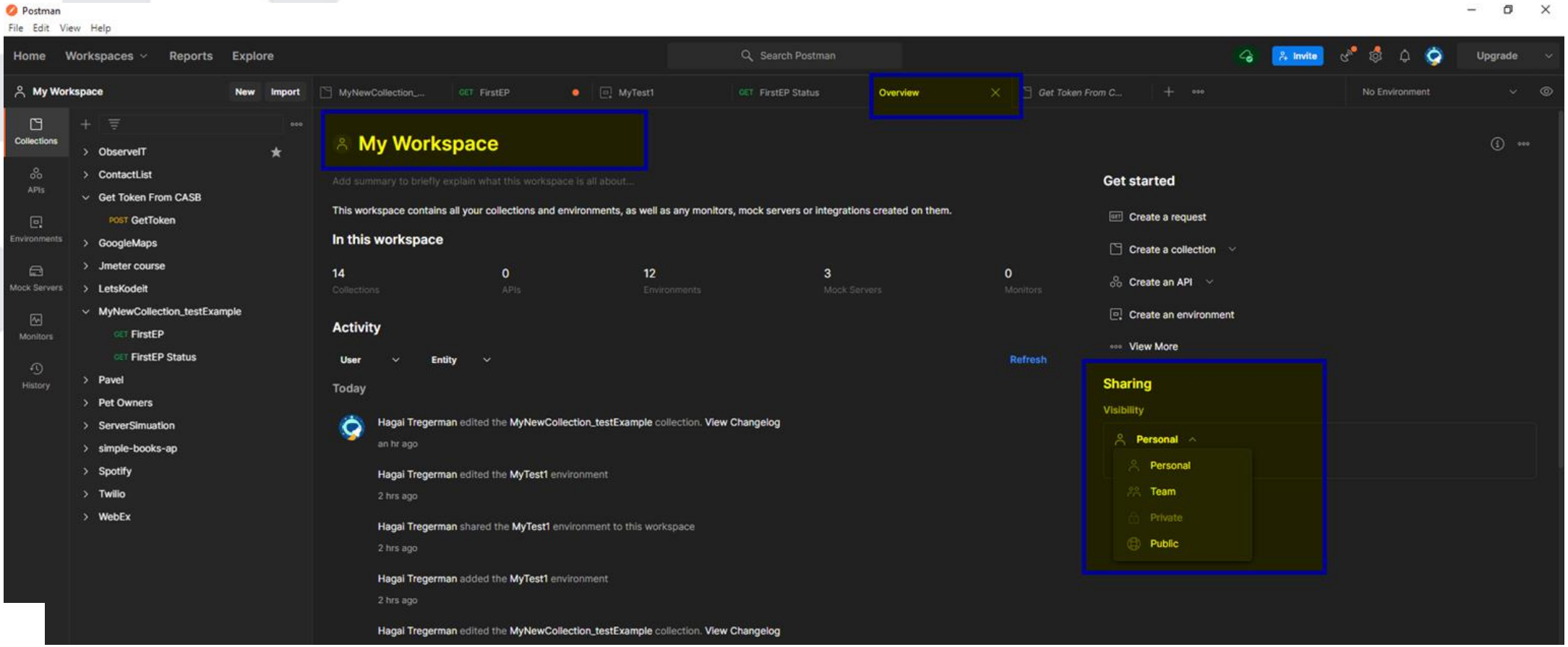
Endpoints

- Status
- Products
  - Get all products
  - Get a product
- Cart
  - Get a cart
  - Get cart items
  - Create a new cart
  - Add an item to cart
  - Modify an item in the cart
  - Replace an item in the cart

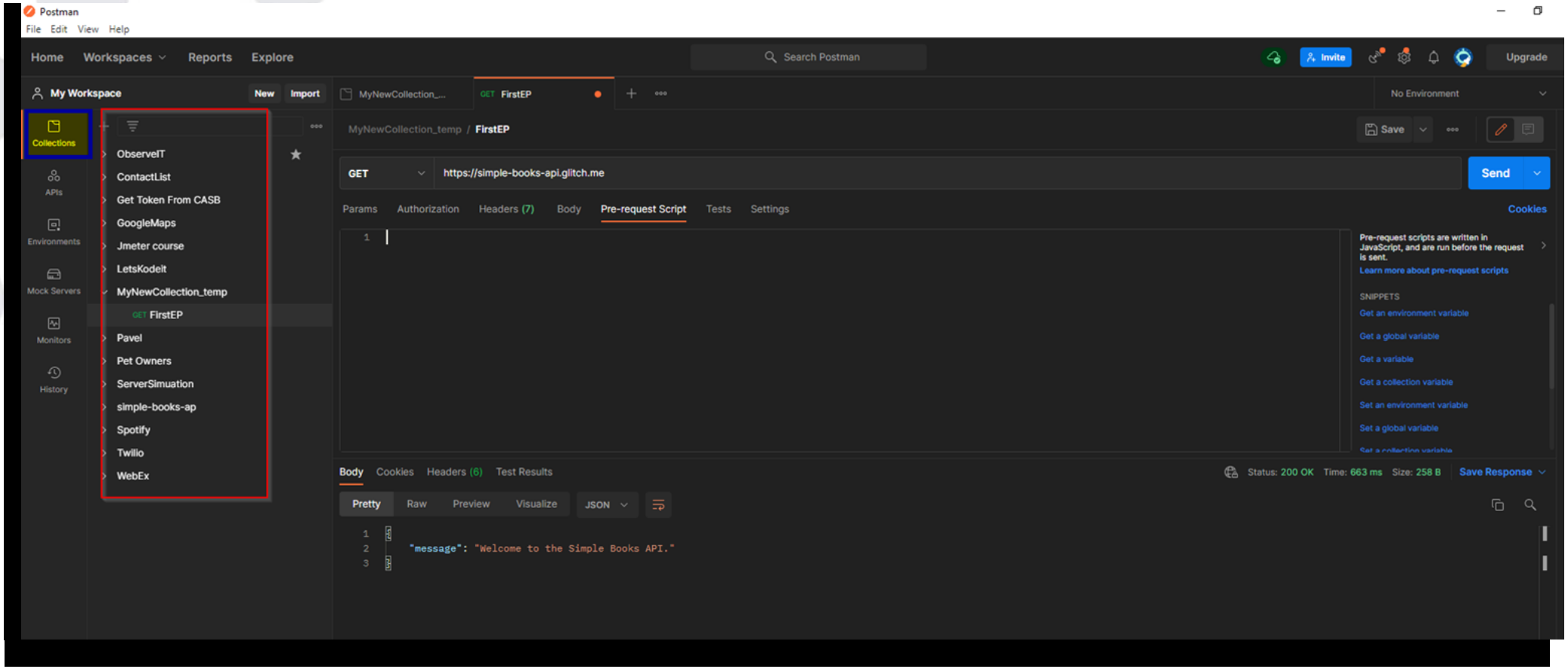
My Workspace



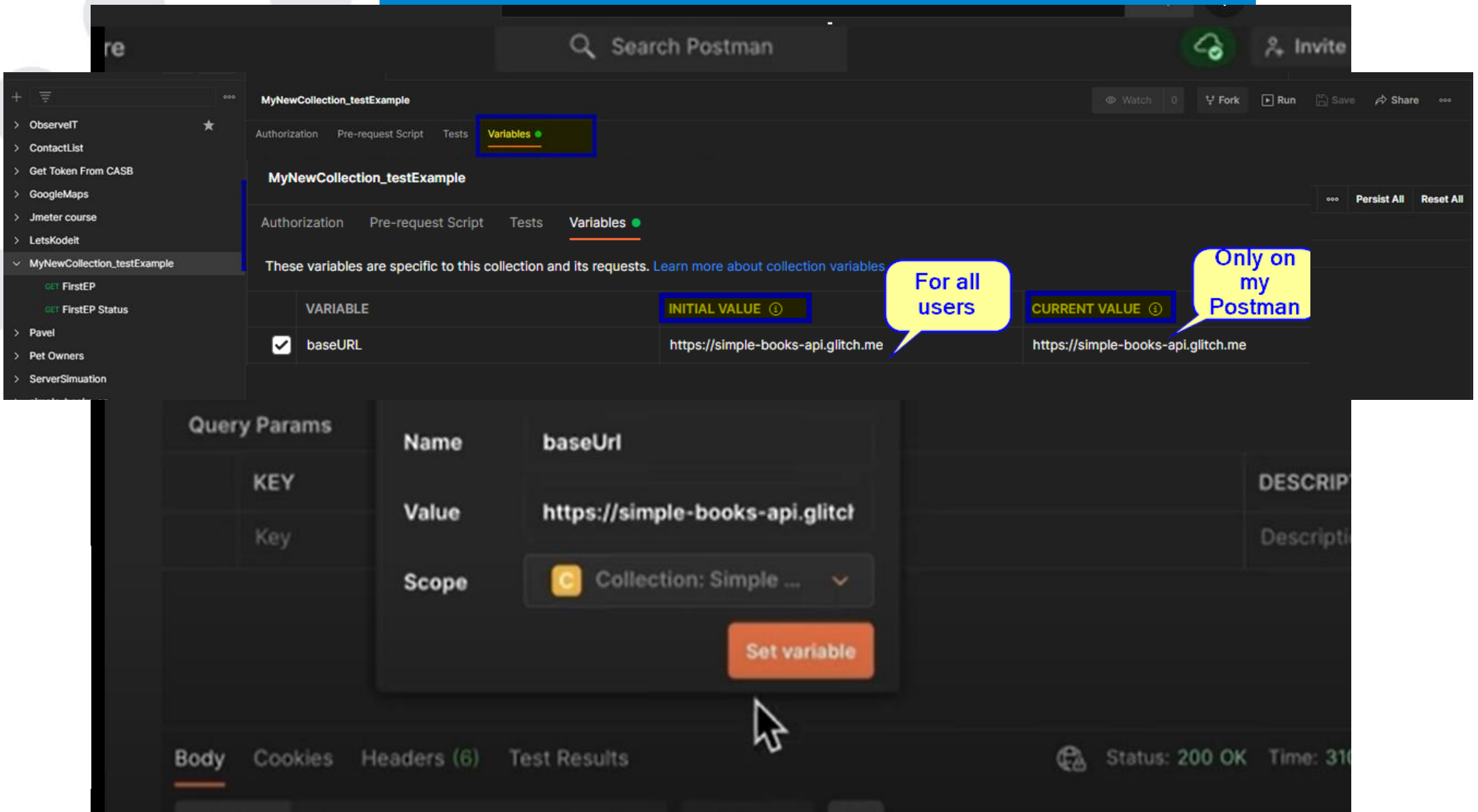
My Workspace



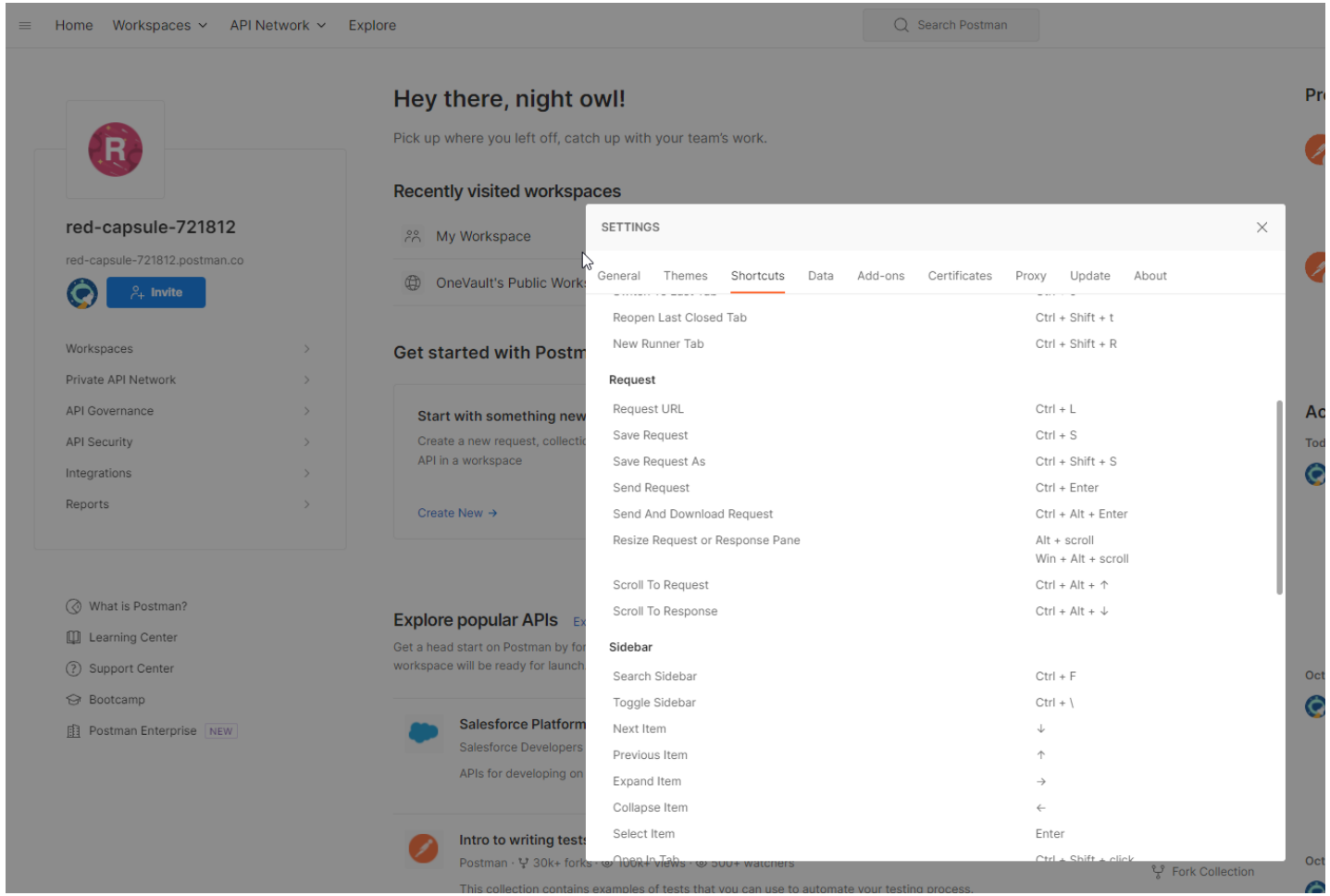
My Collections



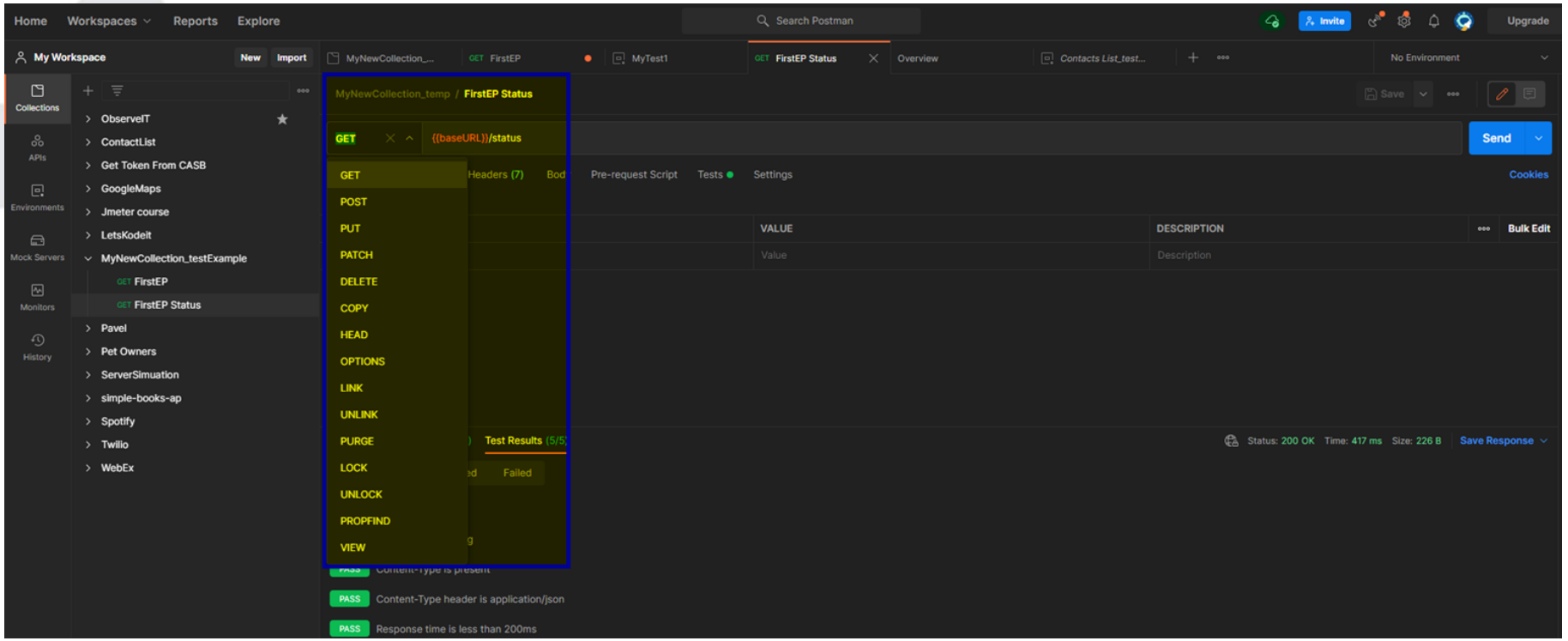
Set Variables to Collection



Shortcuts

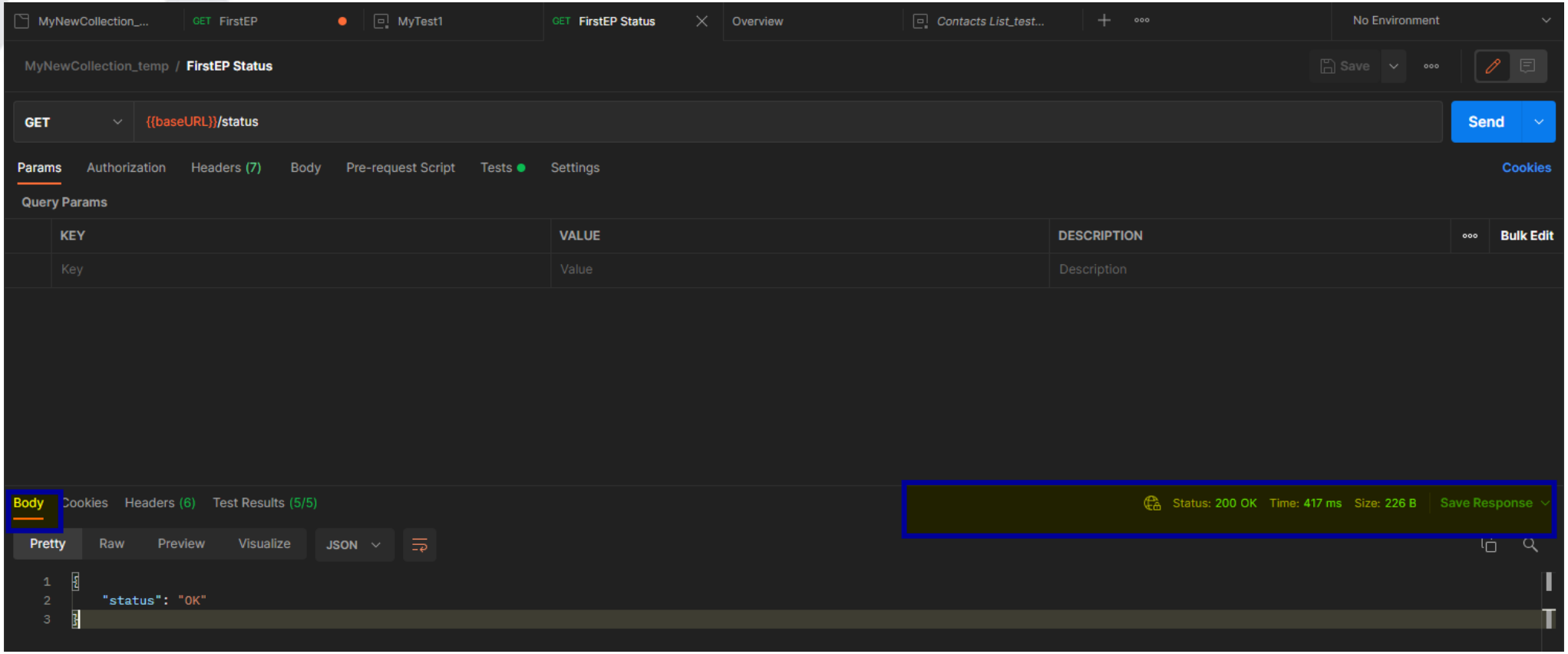

Home
Workspaces
API Network
Explore
Search Postman
Hey there, night owl!
Pick up where you left off, catch up with your team's work.
Recently visited workspaces
My Workspace
red-capsule-721812
red-capsule-721812.postman.co
Invite
Workspaces
Private API Network
API Governance
API Security
Integrations
Reports
What is Postman?
Learning Center
Support Center
Bootcamp
Postman Enterprise
NEW
Get started with Postm
Start with something new
Create a new request, collecti
API in a workspace
Create New
Explore popular APIs
Get a head start on Postman by for
workspace will be ready for launch
Salesforce Platform
Salesforce Developers
APIs for developing on
Intro to writing tests
Postman · 30k+ forks
This collection contains examples of tests that you can use to automate your testing process.
Fork Collection
SETTINGS
General
Themes
Shortcuts
Data
Add-ons
Certificates
Proxy
Update
About
Reopen Last Closed Tab
Ctrl + Shift + t
New Runner Tab
Ctrl + Shift + R
Request
Request URL
Ctrl + L
Save Request
Ctrl + S
Save Request As
Ctrl + Shift + S
Send Request
Ctrl + Enter
Send And Download Request
Ctrl + Alt + Enter
Resize Request or Response Pane
Alt + scroll
Win + Alt + scroll
Scroll To Request
Ctrl + Alt + ↑
Scroll To Response
Ctrl + Alt + ↓
Sidebar
Search Sidebar
Ctrl + F
Toggle Sidebar
Ctrl + \
Next Item
↓
Previous Item
↑
Expand Item
→
Collapse Item
←
Select Item
Enter

REST Verbs


Home
Workspaces
Reports
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
History
ObserveIT
ContactList
Get Token From CASB
GoogleMaps
Jmeter course
LetsKodeit
MyNewCollection_testExample
FirstEP
FirstEP Status
Pavel
Pet Owners
ServerSimuation
simple-books-ap
Spotify
Twilio
WebEx
MyNewCollection_...
FirstEP
MyTest1
FirstEP Status
Overview
Contacts List_test...
No Environment
MyNewCollection_temp / FirstEP Status
Save
GET
{{baseURL}}/status
Send
GET
POST
PUT
PATCH
DELETE
COPY
HEAD
OPTIONS
LINK
UNLINK
PURGE
LOCK
UNLOCK
PROPFIND
VIEW
Headers (7)
Pre-request Script
Tests
Settings
Cookies
VALUE
DESCRIPTION
Bulk Edit
Value
Description
Test Results (5/5
Failed
Status: 200 OK
Time: 417 ms
Size: 226 B
Save Response
PASS
Content-Type header is application/json
PASS
Response time is less than 200ms

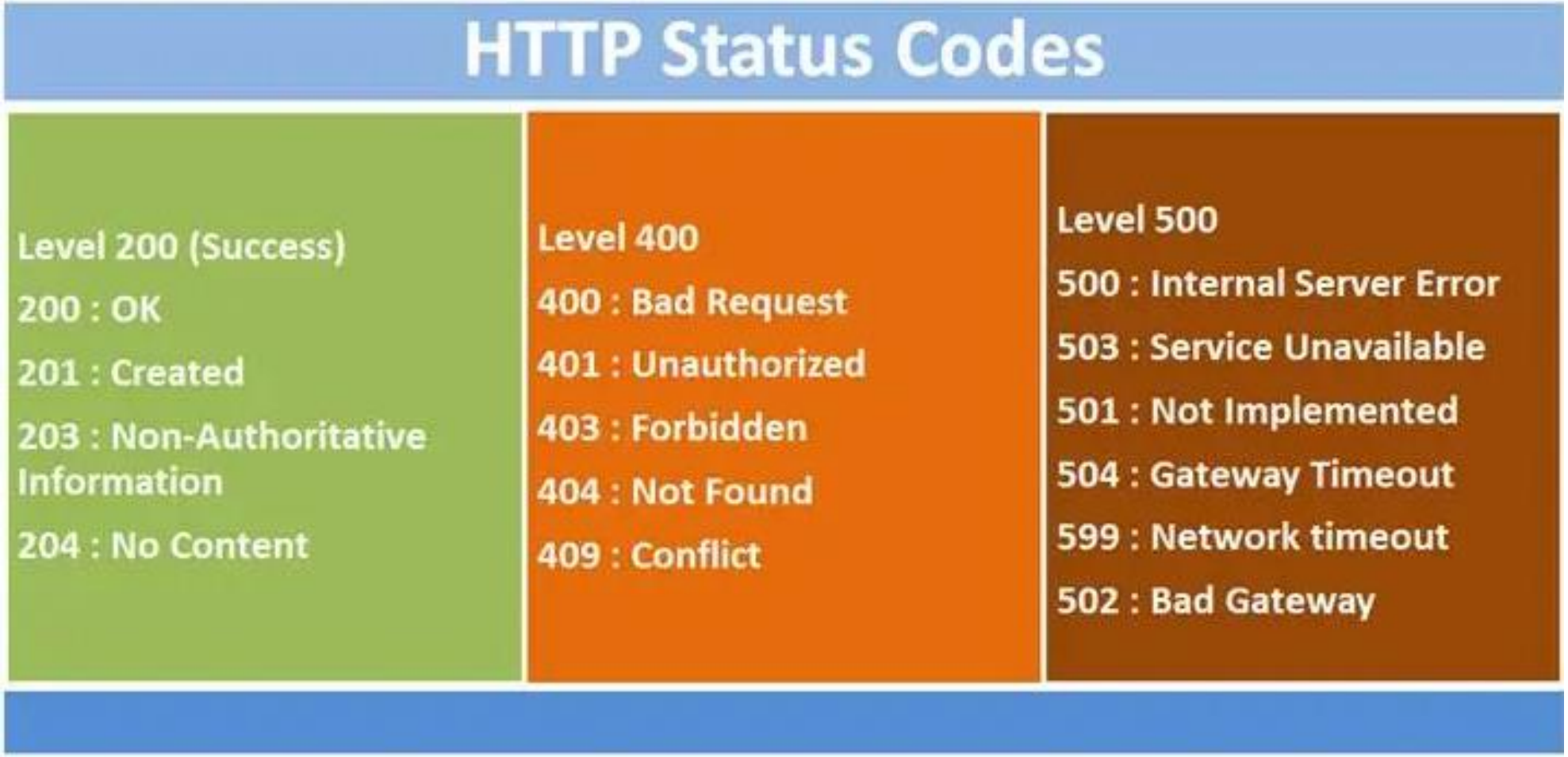
HTTP Response

https://developer.mozilla.org/en-US/docs/Web/HTTP/Status



HTTP Response

https://developer.mozilla.org/en-US/docs/Web/HTTP/Status



Response Errors

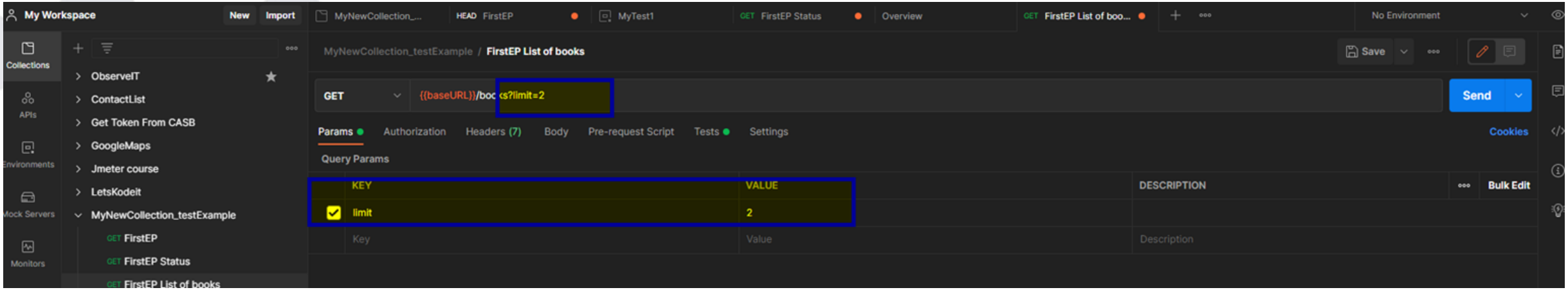
https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

HTTP response status codes

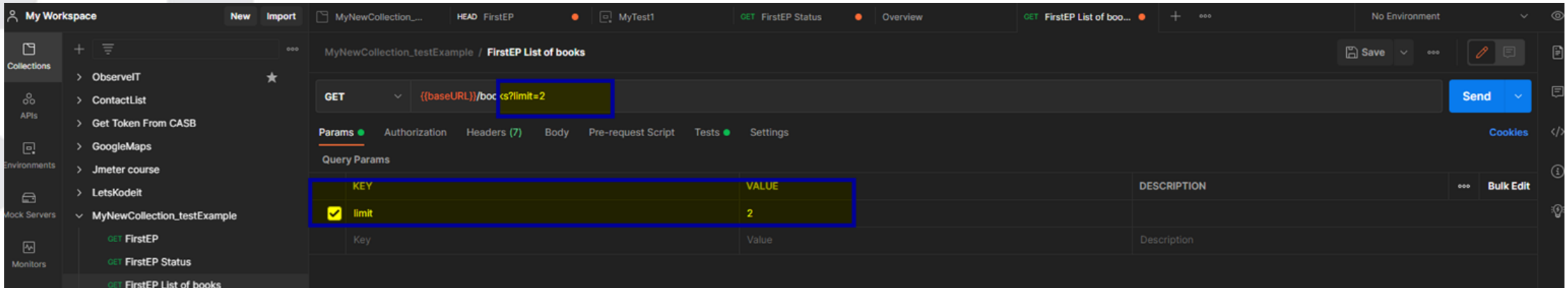
HTTP response status codes indicate whether a specific HTTP request has been successfully completed. Responses are grouped in five classes:

1. Informational responses (`100` – `199`)
2. Successful responses (`200` – `299`)
3. Redirection messages (`300` – `399`)
4. Client error responses (`400` – `499`)
5. Server error responses (`500` – `599`)

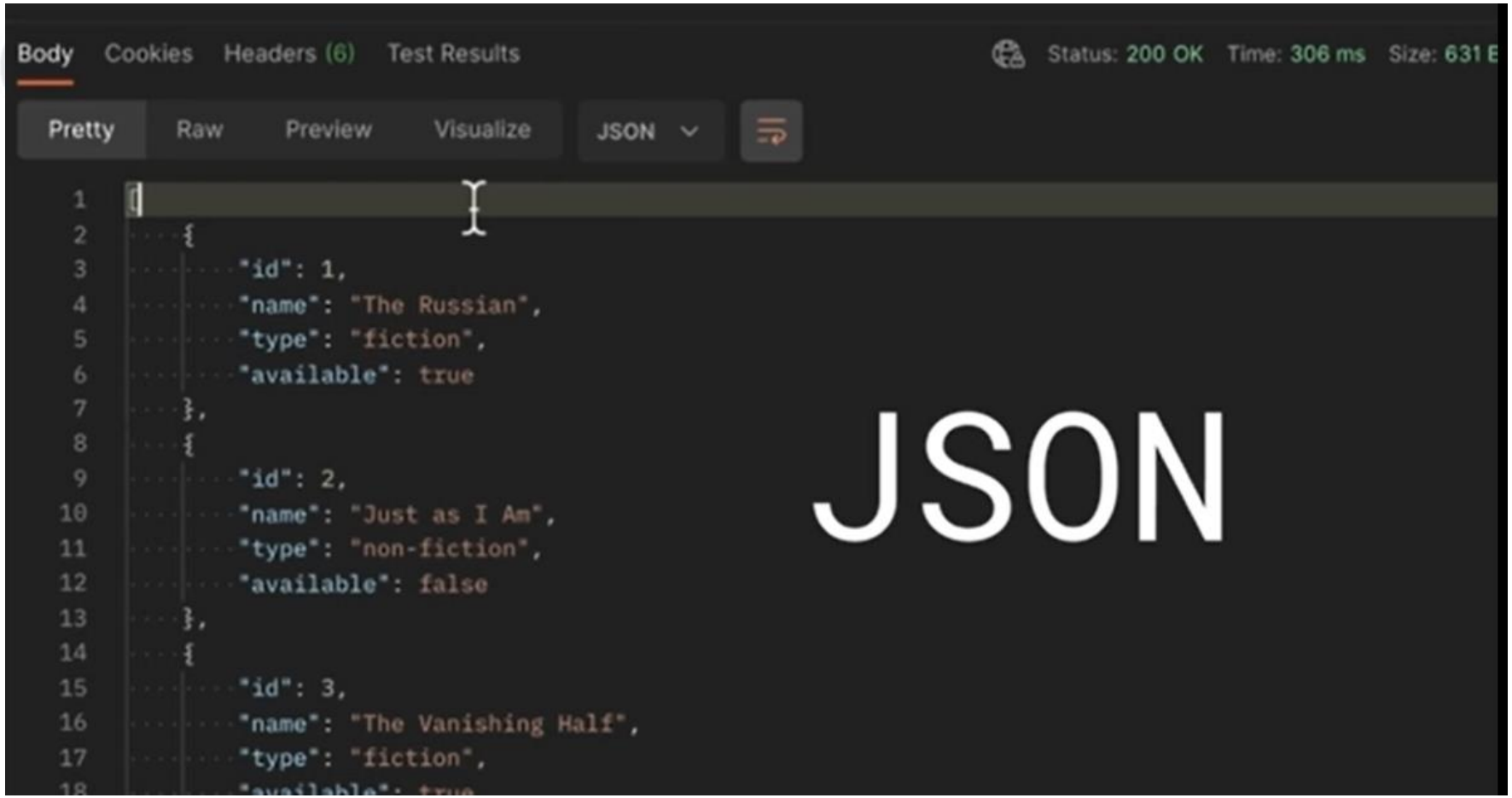
Query Param


My Workspace
New
Import
Collections
APIs
Environments
Mock Servers
Monitors
ObserveIT
ContactList
Get Token From CASB
GoogleMaps
Jmeter course
LetsKodeit
MyNewCollection_testExample
GET FirstEP
GET FirstEP Status
GET FirstEP List of books
MyNewCollection_...
HEAD FirstEP
MyTest1
GET FirstEP Status
Overview
GET FirstEP List of boo...
No Environment
MyNewCollection_testExample / FirstEP List of books
Save
GET
{{baseURL}}/boo ks?limit=2
Send
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies
Query Params
KEY
VALUE
DESCRIPTION
Bulk Edit
limit
2
Key
Value
Description

Query Param – Practice: type



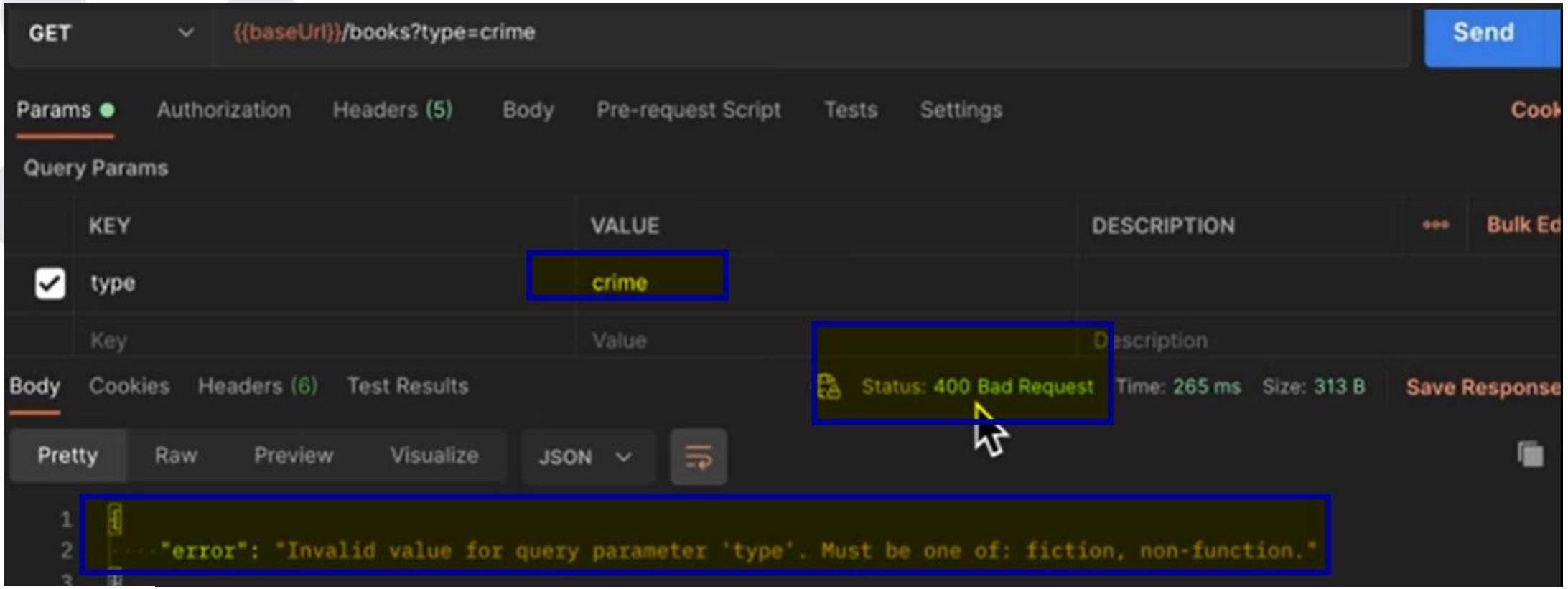
Response



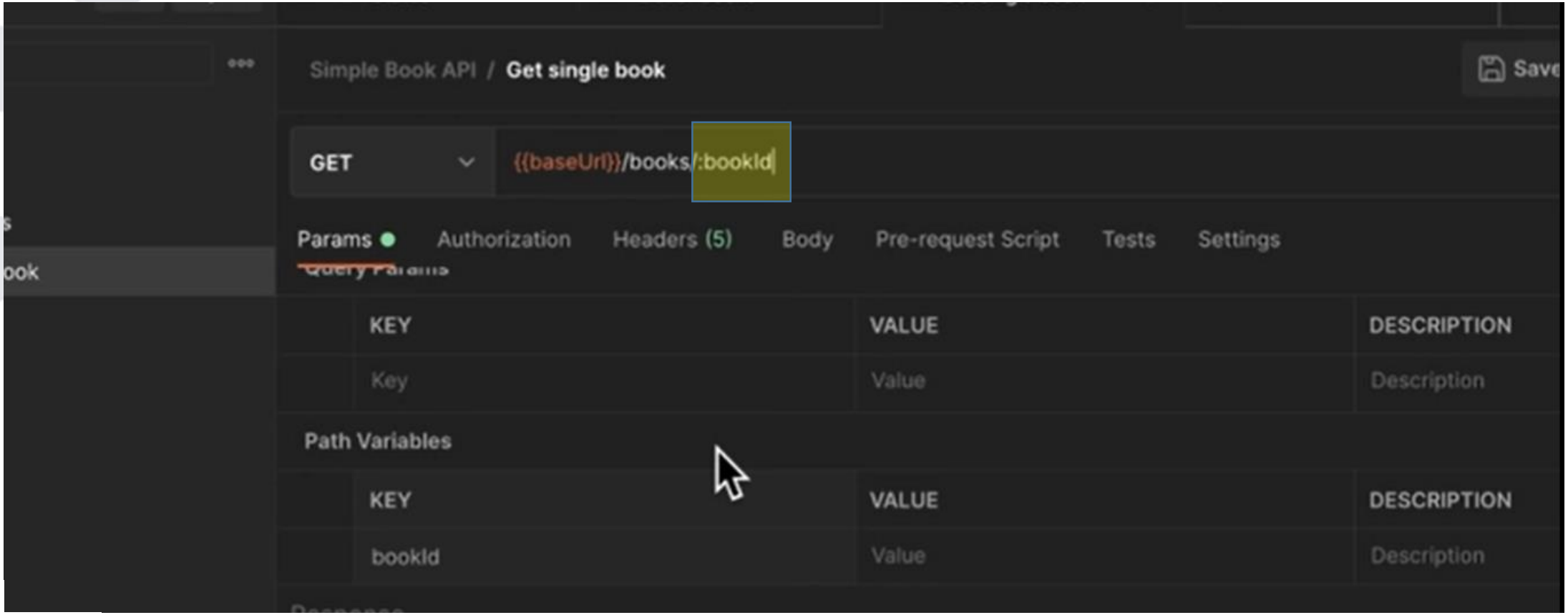
JSONהוא בסך הכל פורמט להעברת נתונים בין שרת לבין סקריפט של JavaScriptאבל אפשר להשתמש בו להעברת נתונים בין שרת לשרת JSON .זה בסך הכל ראשי תיבות של) JavaScript Object Notationלא שזה חשוב לזכור את זה (ומדובר באובייקט JavaScriptשיש לו תכונות שונות עם ערכים.

Response Errors

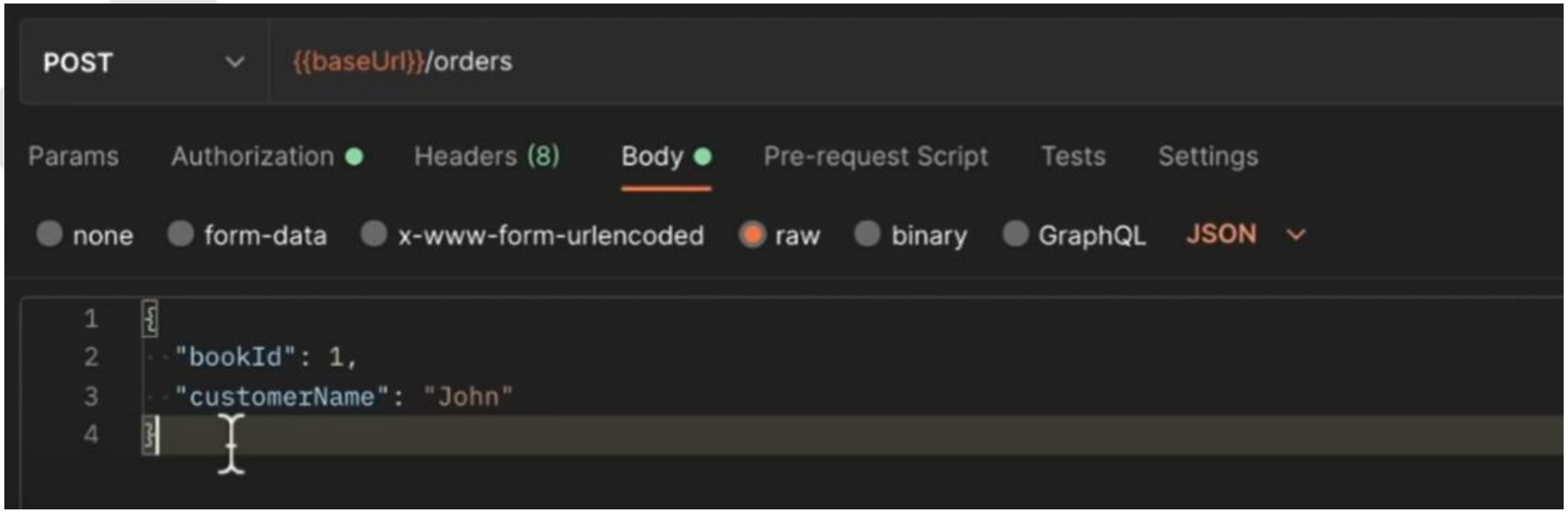
https://developer.mozilla.org/en-US/docs/Web/HTTP/Status



Path Variable

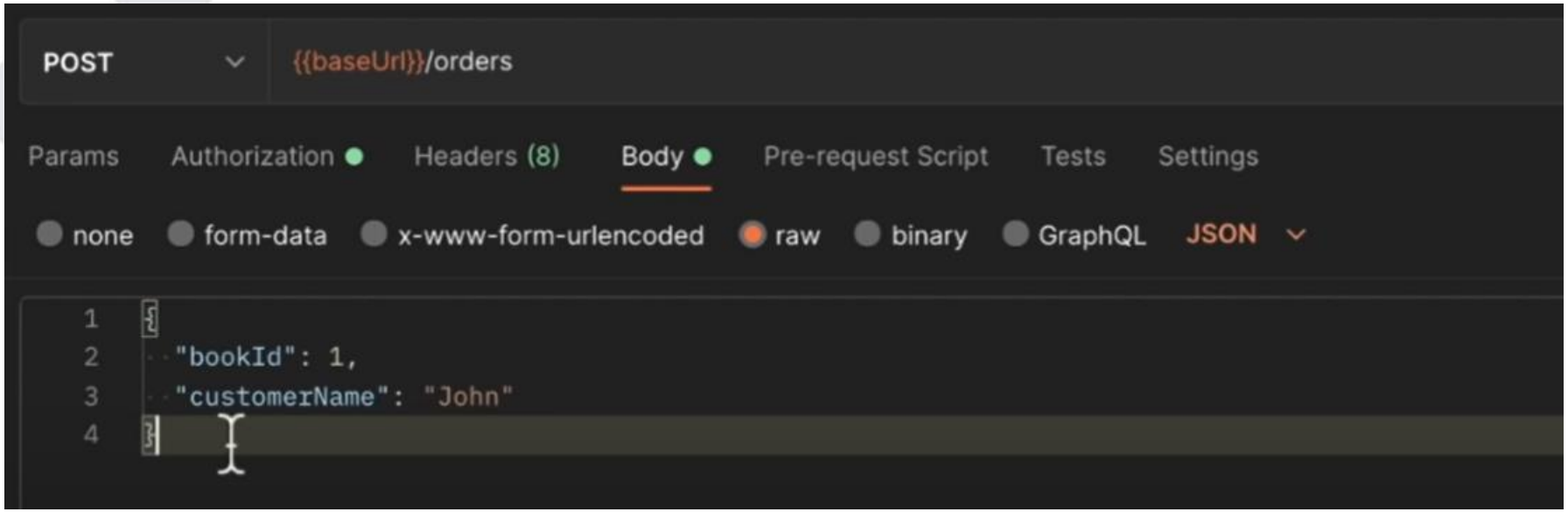


Post Requests



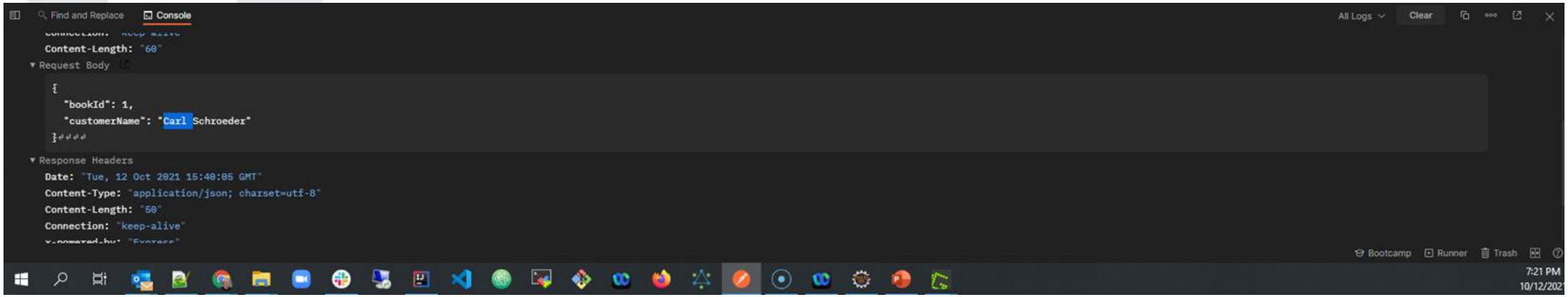
Post Requests - Practice

Check error if insert invalid id for order



Random Variables

Postman console



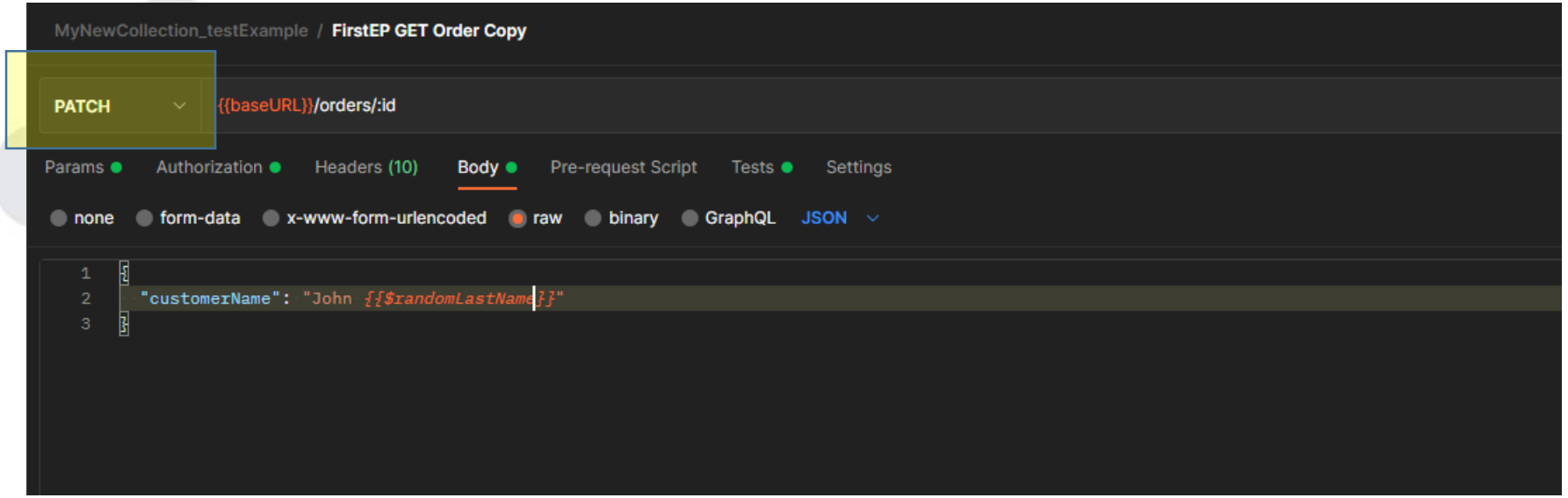
כלי חשוב הוא הקונסול שמראה את הערכים שהמשתנים שלו מכילים ,אנו יכולים להדפיס בעזרת פקודת console.logמשתנים שאנו רוצים לראות את תוכנם.

Practice

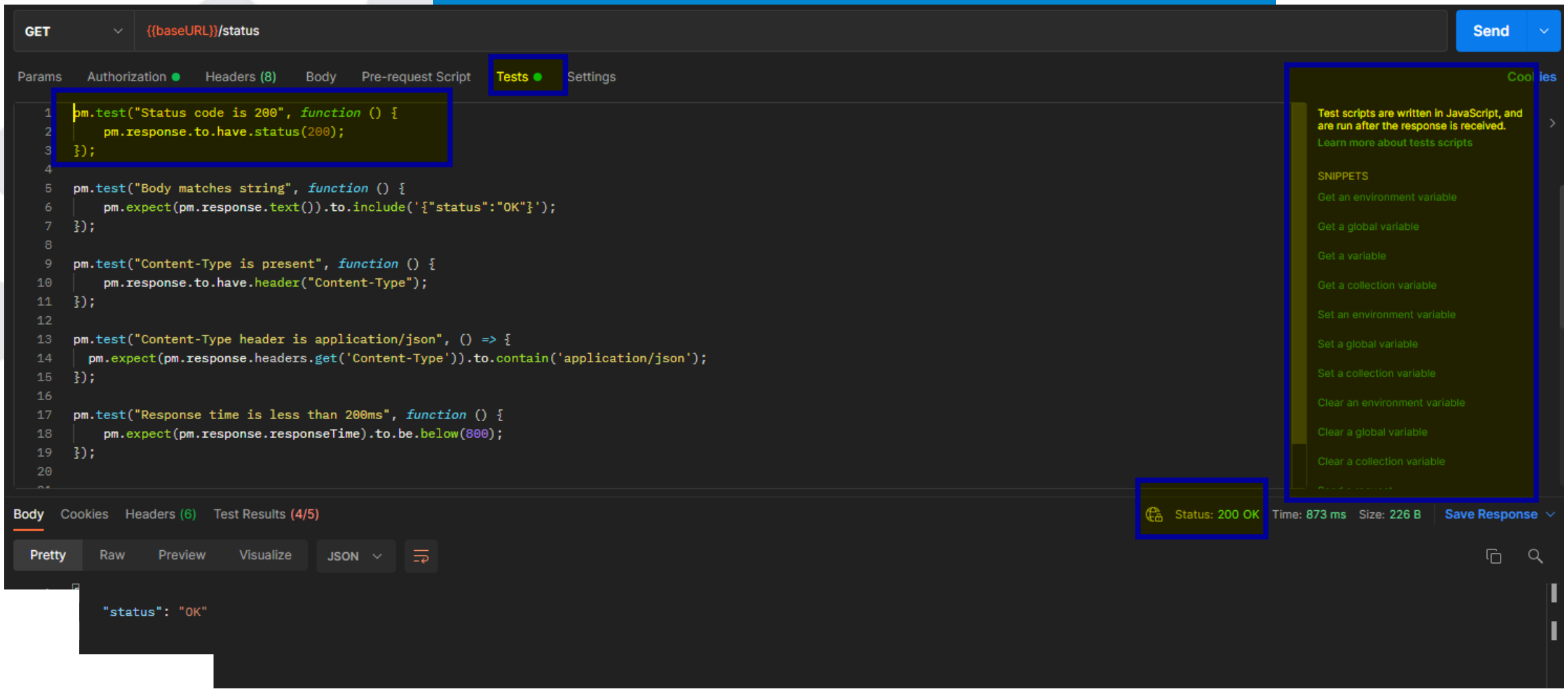
* Get all orders made

* Get specific order made

Patch Action (Update)

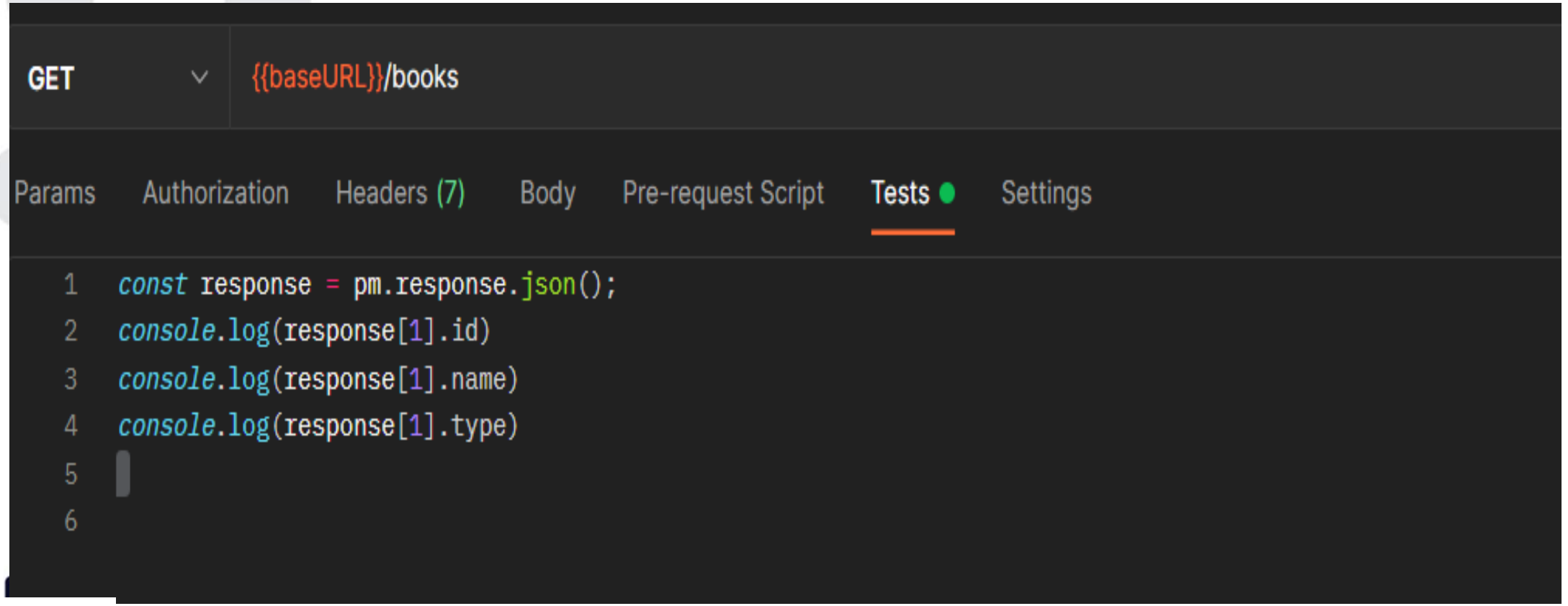


Test and Manipulation



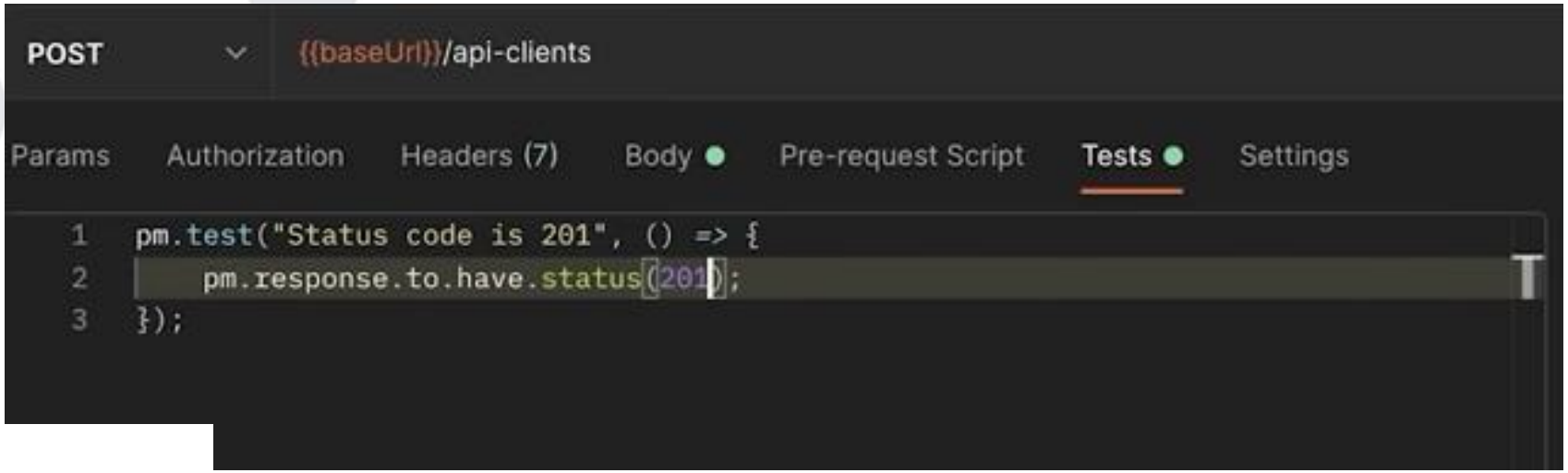
Test and Manipulation

Test Section



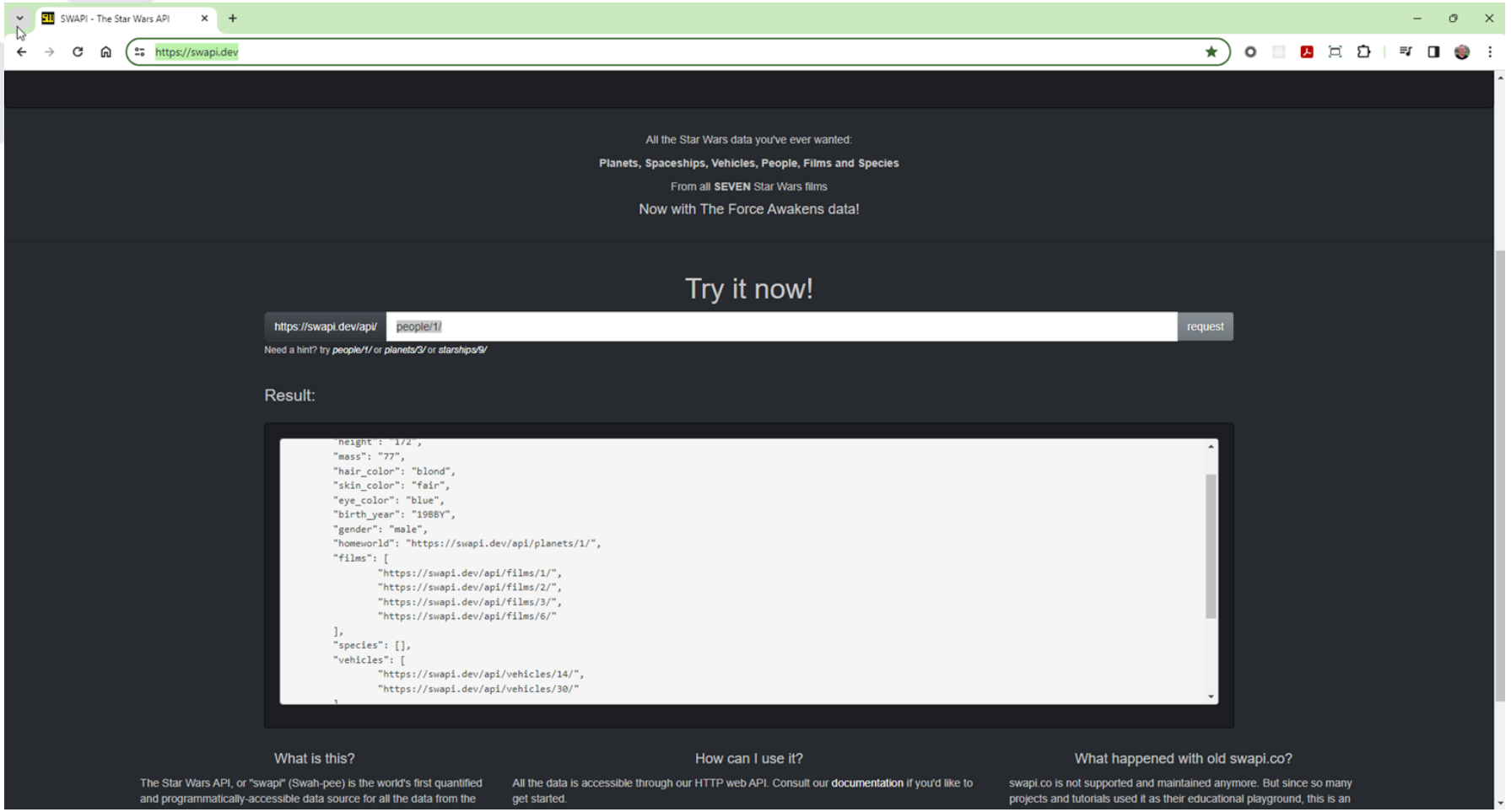
Test and Manipulation

Practice: add at least one **test** for each request we made.



Star wars API

https://swapi.dev/



Twilio.com API: Whatsapp / SMS

https://console.twilio.com

https://www.twilio.com/en-us/blog/send-test-http-requests-twilio-sms-postman

The use of Postman in this article will replace the code below:

Text

```
curl -X POST https://api.twilio.com/2010-04-01/Accounts/{{ACCOUNT_SID}}/Messages.jso
--data-urlencode "From=+<TWILIO_PHONE_NUMBER>" \
--data-urlencode "Body=body" \
--data-urlencode "To=+<YOUR PHONE_NUMBER>" \
-u your_account_sid:your_auth_token
```

```
$ curl wttr.in/London
$ curl wttr.in/Moscow
$ curl wttr.in/Salt+Lake+City
```

Coverage

5	SOAP Requests using POSTMAN
	Multipart FileUpload using Postman
	Interceptor & Proxy to capture browser network traffic
	Mock Servers & Mock Requests in Postman
	Using Authentication in POSTMAN
	Data Driven testing using Postman
6	Scripting fundamentals in POSTMAN
	Practical Examples on Scripting in POSTMAN
	Newman for CLI execution
	Source control with Git, GitHub
	Jenkins,HTML Reports - Executing collections in Jenkins

Make HTTP calls using the SOAP protocol

- https://learning.postman.com/docs/sending-requests/soap/making-soap-requests/

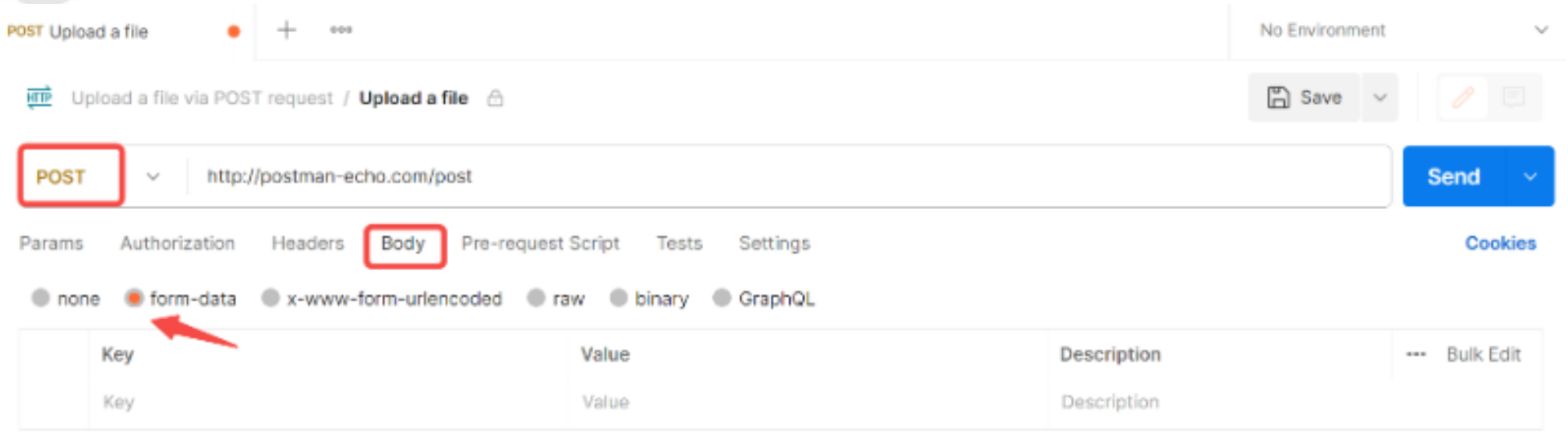
https://copilot.microsoft.com/:

* What is SOAP API ?
* which one is older SOAP or REST ?

		SOAP Representational State Transfer	REST Representational State Transfer
	When	late 1990s.	early 2000s
	Who	Microsoft	Dr. Roy Fielding
	key principles	Structured and Formalized: • **Envelope**: Defines the overall structure of the message. • **Encoding**: Specifies rules for expressing data types. • **Requests**: Describes how each SOAP API request is structured. • **Responses**: Outlines the structure of SOAP API responses.	• HTTP verbs (such as GET, PUT, POST, DELETE) and URIs (resources) for communication. Each request receives an HTTP **response** with a **status code** and a **body** • Stateless: Each request is self-descriptive, containing enough context for the server • Client-Server: • Cacheable
	Language	XML	JSON

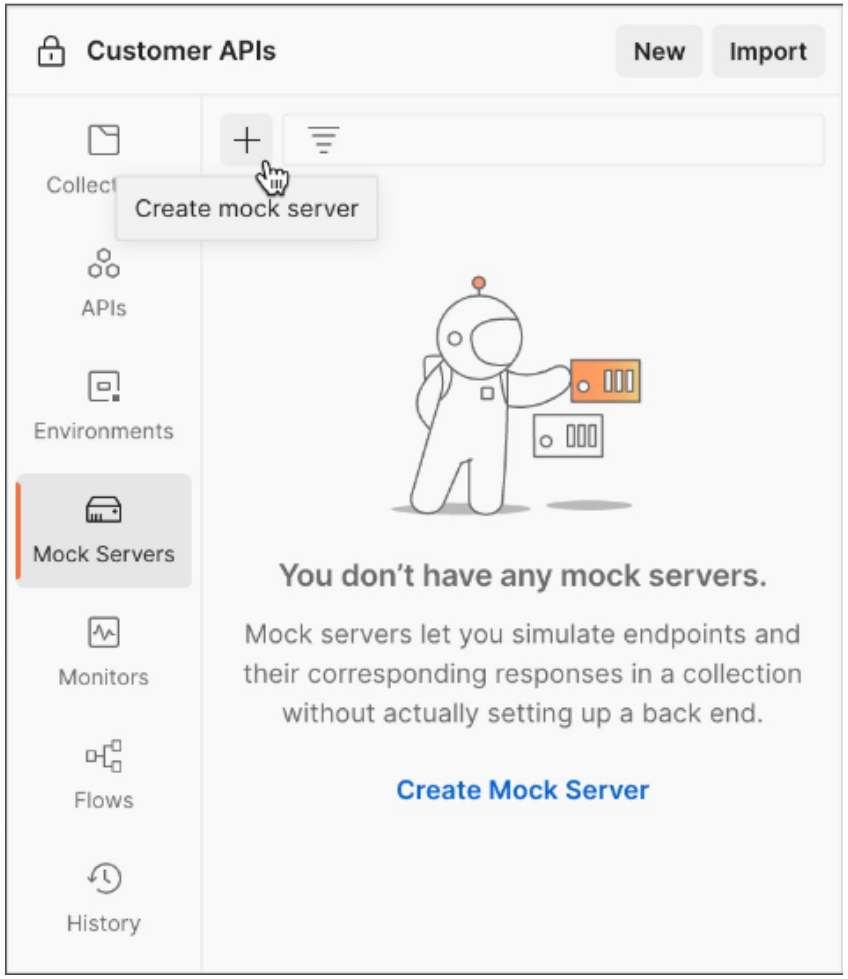
Testing file uploads with Postman (multipart/form-data)

https://apidog.com/blog/how-to-upload-a-file-and-json-data-in-postman/



Configure and use a Postman mock server

https://learning.postman.com/docs/designing-and-developing-your-api/mocking-data/setting-up-mock/



Newman the cli companion for postman

https://www.npmjs.com/package/newman

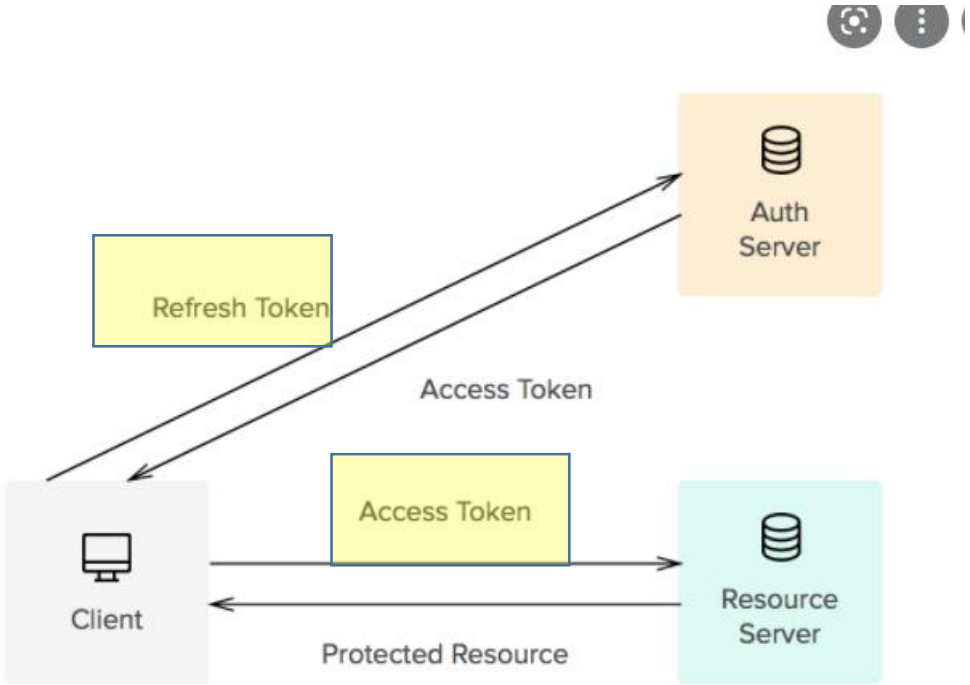
Newman is a command-line collection runner for Postman. It allows you to effortlessly run and test a Postman collection directly from the command-line. It is built with extensibility in mind so that you can easily integrate it with your continuous integration servers and build systems.

```
C:\Data\Newman>newman run dataaccess.com.postman_collection.json
newman

dataaccess.com

→ example_l1
  POST https://www.dataaccess.com/webservicesserver/NumberConversion.wso [200 OK, 588B, 1499ms]
  √  Status code is 200
  √  Body matches string: NumberToWordsResult
```

	executed	failed
iterations	1	0
requests	1	0
test-scripts	2	0
prerequest-scripts	1	0
assertions	2	0

```
total run duration: 1583ms
total data received: 323B (approx)
average response time: 1499ms [min: 1499ms, max: 1499ms, s.d.: 0µs]
```

Access Token

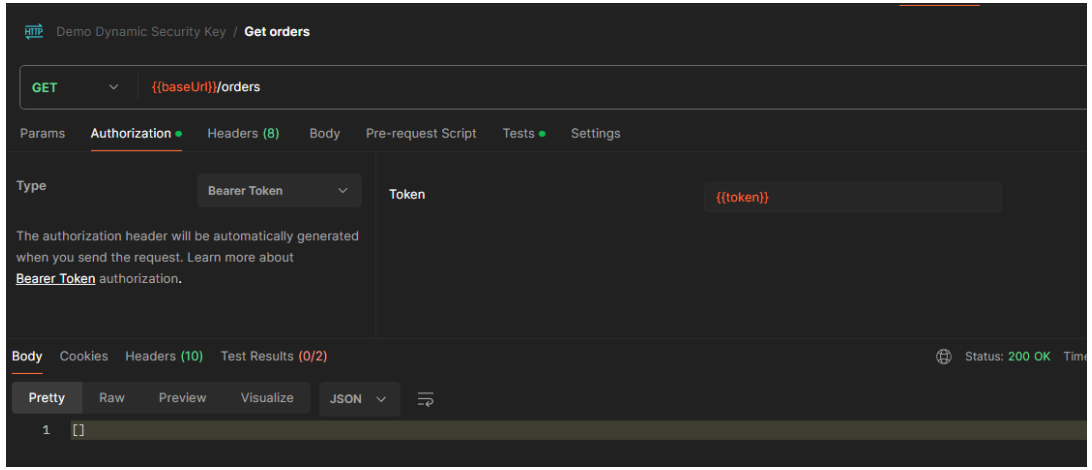


API Authentication

https://github.com/vdespa/Postman-Complete-Guide-API-Testing/blob/main/simple-grocery-store-api.md#API-Authentication

Some endpoints may require authentication. To submit or view an order, you need to register your API client and obtain an access token.

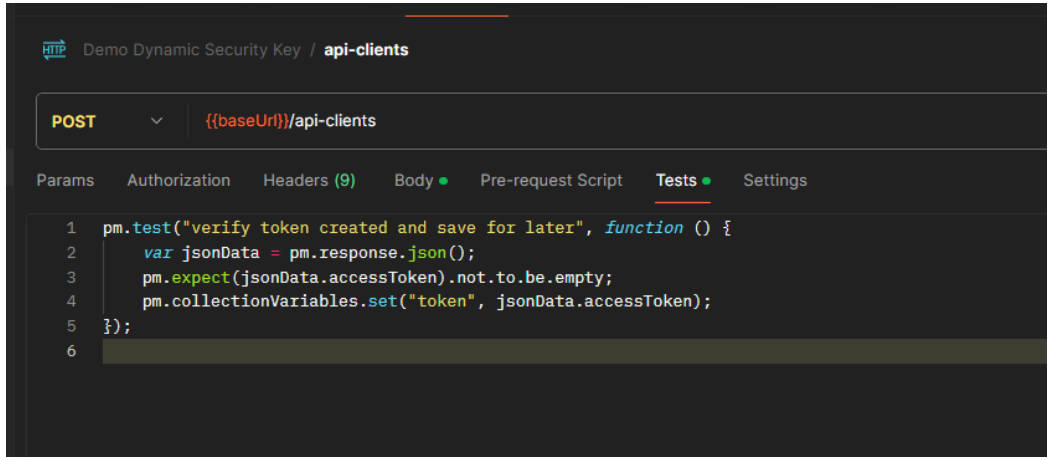
The endpoints that require authentication expect a bearer token sent in the `Authorization` header.



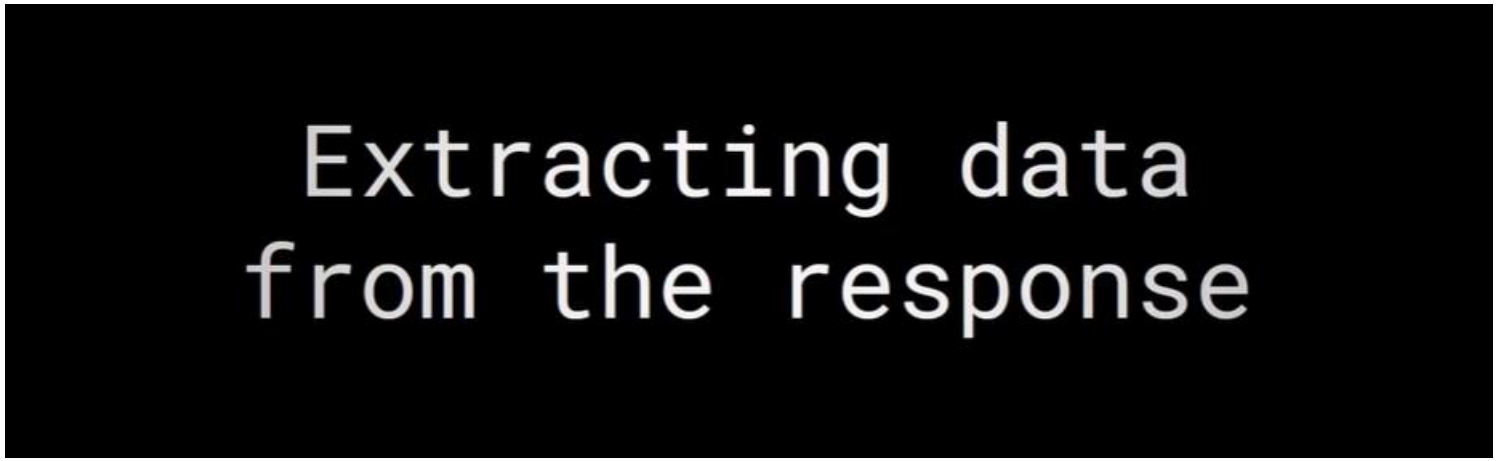
API Authentication

https://github.com/vdespa/Postman-Complete-Guide-API-Testing/blob/main/simple-grocery-store-api.md#API-Authentication

> Some endpoints may require authentication. To submit or view an order, you need to register your API client and obtain an access token.
>
> The endpoints that require authentication expect a bearer token sent in the `Authorization` header.



Getting Data by Filter



Getting Data Dynamically

```
const response = pm.response.json();

const avaliableBook = response.filter((item) => item.available === true);

console.log(avaliableBook[0]);
```

Getting Data Dynamically

```
pm.environment.set("bookId", avaliableBook[0].id);
console.log("Id of the ordered book is: " + pm.environment.get("bookId"));
```

GET {{baseURL}}/books?type=fiction

Params Authorization Headers (7) Body Pre-request Script Tests Settings

```
2   const response = pm.response.json();
3   const avaliableBook = response.filter((item) => item.available === true);
4
5   if (avaliableBook[0]){
6   const book = avaliableBook[0];
7   pm.test("is book found", () => {
8           pm.expect(book.available).to.eql(true);
9
10  })
11
12  if (book.available){
13          pm.environment.set("bookId", avaliableBook[0].id);
```

סיכום ושאלות ☺