



API Testing Overview

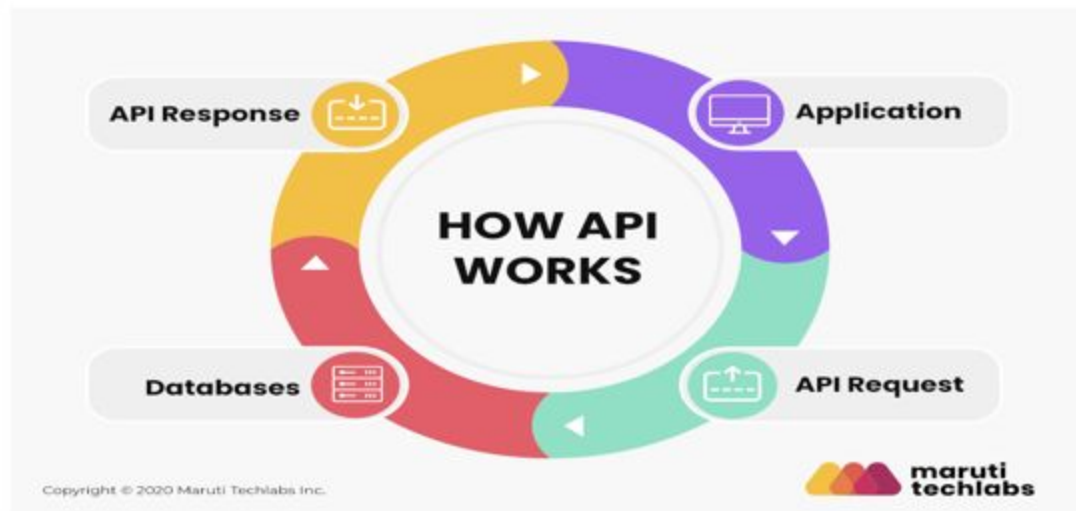
- What is API and API testing
- How API works
- 3 Layer Architecture
- API Testing
- Types of API Testing
- API Testing Tools
- Postman Workspaces
- HTTP requests methods with Postman
- Status code

What is API and API testing

- Application Programming Interface.
- A Software intermediary that allows two applications to talk to each other.
- It enables companies to open up their applications' data and functionality to external third-party like developers, business partners, and internal departments within their companies.
- API testing is a type of software testing that analyzes an application program interface (API) .
- To verify it fulfills its expected functionality, security, performance and reliability.

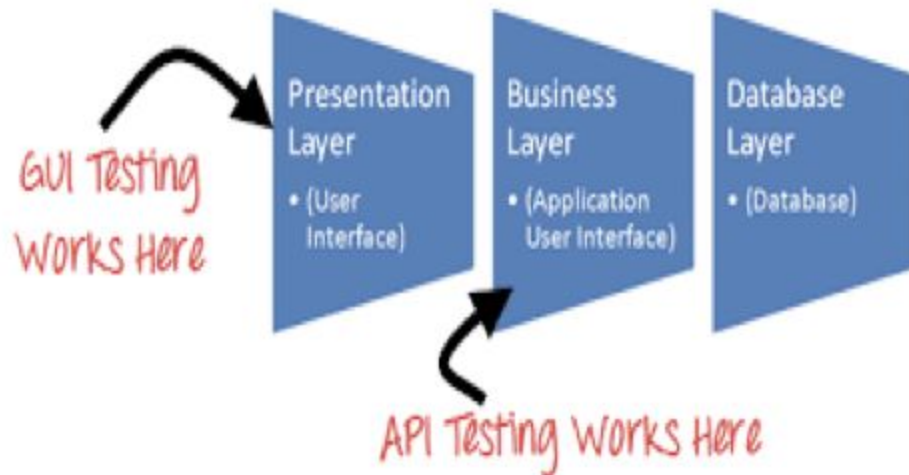
How an API works

- A client application initiates an API call.
- After receiving a valid request.
- The server sends a *response*.
- The API transfers the data



3 Layer Architecture

- **Presentation** (or user interface) layer
- **Business** layer
- **Database** layer for modeling and manipulating data



Types of API Testing

- **Functional testing** - Check that an API returns the right response for a given request.
- **Load testing**- Check how an API handles a large volume of requests over a short period.
- **Runtime and error detection testing** – It is used to evaluate the actual running of the API and typically focus on monitoring, execution errors, resource leaks, or error detection.
- **Security testing** - How an API responds to and resists cyberattacks.
- **Penetration Testing** - Penetration tests involve users with limited API knowledge trying to attack the API, which enables testers to assess the threat vector from an outside perspective.
- **Fuzz Testing** - This type of API test sends a large number of randomized requests to see if your API responds with errors, processes any of these inputs incorrectly, or crashes.
- **Validation testing** - Validation tests are run late in the testing stage to verify the behavior and efficiency of the API.

API Testing tools

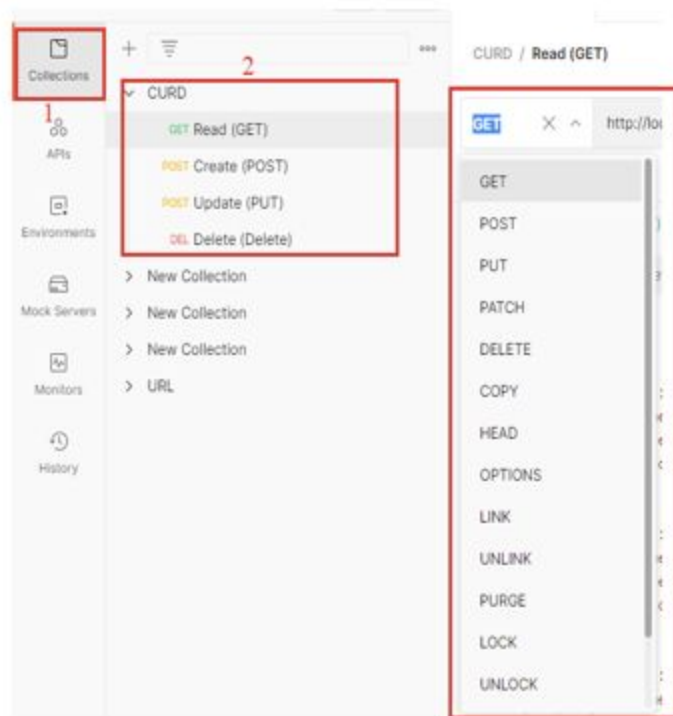
- RapidAPI.
- REST-assured.
- Postman
- Paw
- SoapUI
- Katalon Studio
- JMeter.

Postman Workspaces

There are three kinds of workspaces in Postman:

- Personal workspace - personal usage.
- Private workspace - is only available to people who you invite to collaborate within the workspace
- Public Workspaces - allow you to share your APIs with the world. They are searchable and accessible for free.
- A Postman collection consists of a group of HTTP requests. As the name suggests, collections help you organize your workspace.

- GET – Retrieve the data
- POST - add new data
- PUT - replace existing data
- DELETE - delete existing data



POST Method

Home Workspaces ▾ Reports Explore

Search Postman



Sign In

Create Account

Working locally in Scratch Pad. Switch to a Workspace

Scratch Pad

New

Import

GET Read (GET)

POST Create (POST)

POST Update (PUT)

DEL Delete (Delete)

New Environment



No Environment



Collections



CURD / Create (POST)



▼ CURD ☆



APIs

GET Read (GET)

1

POST

http://localhost/rest-api-crud-using-php/RestController.php?page_key=create ...

Send



POST Create (POST)

2

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

Cookies



Environments

POST Update (PUT)

DEL Delete (Delete)

3

Query Params

KEY

VALUE

DESCRIPTION



Bulk Edit



page_key

create

Mock Servers



Monitors

> New Collection

> New Collection

> New Collection

> URL

POST Method Contd..

The screenshot displays the Postman interface for configuring a POST request. The left sidebar shows a collection of requests under the 'CURD' folder, with 'POST Create (POST)' selected. The main panel shows the request configuration for the endpoint `http://localhost/test-api-crud-using-php/RestController.php?page_key=create...`. The 'Authorization' tab is selected, and the 'Bearer Token' dropdown menu is open, showing various authentication options. The 'Token' input field is also visible.

Request Configuration:

- Method:** POST
- URL:** `http://localhost/test-api-crud-using-php/RestController.php?page_key=create...`
- Params:** (Empty)
- Authorization:** Bearer Token (Selected)
- Headers (10):** (Empty)
- Body:** (Empty)
- Pre-request Script:** (Empty)
- Tests:** (Empty)
- Settings:** (Empty)
- Cookies:** (Empty)

Authentication Options (from dropdown):

- Inherit auth from parent
- No Auth
- API Key
- Bearer Token
- Basic Auth
- Digest Auth
- OAuth 1.0
- OAuth 2.0
- Hawk Authentication
- AWS Signature

Response:

Status: 200 OK Time: 596 ms Size: 260 B Save Response

The response body is shown in 'Pretty' format, displaying a JSON object with a 'success' property set to true.

POST Method Contd..

New Import GET Read (GET) POST Create (POS) POST Update (PU) DEL Delete (Delete) New Environment + ... No Environment

CURD / Create (POST) Save ...

POST http://localhost/test-api-crud-using-php/RestController.php?page_key=create ... Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

Headers 8 hidden

	KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
☒	Accept	application/json				
☒	Content-Type	application/x-www-form-urlencoded				
	Key	Value	Description			

2

POST Method Contd..

REST Client interface showing a POST request configuration and response.

Request Configuration:

- Method:** POST
- URL:** http://localhost:rest-api-crud-using-php/RestController.php?page_key=create ...
- Params:** none
- Authorization:** none
- Headers (10):** none
- Body:** form-data
- Pre-request Script:** none
- Tests:** none
- Settings:** none

Body Data:

KEY	VALUE	DESCRIPTION
☒ name	Apple	
☒ model	iPhone11	
☒ color	White	

Response:

Status: 200 OK Time: 596 ms Size: 260 B

Body Content:

```
1 {
2   "success": 1
3 }
```

GET Method

New Import GET Read (GET) POST Create (POST) POST Update (PUT) DEL Delete (Delete) New Environment + ... No Environment

CURD / Read (GET) Save ...

1 GET http://localhost/test-api-crud-using-php/RestController.php?page_key=list ... Send

2 Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

3 Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
☒	page_key	list			

GET Method Condt...

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 40 ms Size: 446 B Save Response

Pretty Raw Preview Visualize JSON

```
2  "output": [  
3    {  
4      "id": "1",  
5      "name": "Apple",  
6      "model": "iPhone11",  
7      "color": "White"  
8    },  
9    {  
10     "id": "2",  
11     "name": "Apple",  
12     "model": "iPhone12",  
13     "color": "Black"  
14   },  
15   {  
16     "id": "3",  
17     "name": "Apple",  
18     "model": "iPhone13 Pro",  
19     "color": "Gold"
```

PUT Method

New Import GET Read (GET) POST Create (POST) **POST Update (PUT)** DEL Delete (Delete) New Environment + ... No Environment

+ ... CURD / Update (PUT) Save ...

2 POST http://localhost/test-api-crud-using-php/RestController.php?page_key=update&id=2 Send

3 Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
☒	page_key	update			
☒	id	2			
	Key	Value	Description		

4

PUT Method Contd...

REST Client interface showing a PUT request configuration and response.

Request Configuration:

- Method:** PUT
- URL:** `http://localhost/rest-api-crud-using-php/RestController.php?page_key=update&id=2`
- Params:** none
- Authorization:** none
- Headers (10):** none
- Body:** form-data
- Pre-request Script:** none
- Tests:** none
- Settings:** none

Body Data:

KEY	VALUE	DESCRIPTION
☒ name	Apple	
☒ model	iPhone 12	
☒ color	Purple	

Response:

Status: 200 OK Time: 109 ms Size: 260 B

Body:

```
1 {  
2   "success": 1  
3 }
```

PUT Method Contd...

New Import GET Read (GET) POST Create (POST) POST Update (PUT) DEL Delete (Delete) New Environment + ... No Environment

CURD / Read (GET) Save

GET http://localhost/rest-api-crud-using-spring/RestController.php?page_key=list ... Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
4      "id": "1",
5      "name": "Apple",
6      "model": "iPhone11",
7      "color": "White"
8    },
9    {
10     "id": "2",
11     "name": "Apple",
12     "model": "iPhone 12",
13     "color": "Purple"
14   },
15   {
16     "id": "3",
17     "name": "Apple",
18     "model": "iPhone13 Pro",
19     "color": "Gold"
```

Status: 200 OK Time: 27 ms Size: 513 B Save Response

PUT Method Contd...

PUT Method Contd...

1 POST Update (PUT)

2 POST http://localhost/test-api-crud-using-php/RestController.php?page_key=update&id=4

3 Params Authorization Headers (10) Body Pre-request Script Tests Settings

4

KEY	VALUE	DESCRIPTION
☒ page_key	update	
☒ id	4	

Body Cookies Headers (7) Test Results

7

8 Status: 404 Not Found Time: 86 ms Size: 267 B Save Response

9

```
1 {
2   "success": 0
3 }
```

DELETE Method

1 **DEL Delete (Delete)**

2 **DELETE** http://localhost/rest-api-crud-using-php/RestController.php?page_key=delete&id=1 **Send**

3 **Params** Authorization Headers (10) Body Pre-request Script Tests Settings

4 **Query Params**

	KEY	VALUE	DESCRIPTION
☒	page_key	delete	
☒	id	1	

5 **Body** Cookies Headers (7) Test Results

6 **Body** Pretty Raw Preview Visualize JSON `1 "success": 1`

Status: 200 OK Time: 189 ms Size: 260 B **Save Response**

DELETE Method Cond...

GET

http://localhost/rest-api-crud-using-php/RestController.php?page_key=list ...

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Cookies

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 60 ms Size: 452 B

Save Response

Pretty

Raw

Preview

Visualize

JSON



```
1
2
3
4
5
6
7
8
9
10
11
12
13
14

{
  "output": [
    {
      "id": "2",
      "name": "Apple",
      "model": "iPhone 12",
      "color": "Purple"
    },
    {
      "id": "3",
      "name": "Apple",
      "model": "iPhone13 Pro",
      "color": "Gold"
    }
  ]
}
```

Collection

working locally in Scratch Pad. [Switch to a workspace](#)

New

Import

GET Read (GET)

POST Create (POST)

POST Update (PUT)

DEL Delete (Delete)

New Enviro

CURD

CURD

1

Authorization

Pre-request Script

Tests

Variables

This authorization method will be used for every request. You can override this by specifying one in the request.

Type

API Key

The authorization header will be automatically generated for you.

[Learn more about authorization](#)

Key

Key

Share

Save

Run

2

Create a fork

Create Pull Request

Merge changes

Pull changes

Export

Manage roles

Status code

The status codes are divided into five categories.

- **1xx: Informational** – Communicates transfer protocol-level information.
- **2xx: Success** – Indicates that the client's request was accepted successfully.
- **3xx: Redirection** – Indicates that the client must take some additional action in order to complete their request.
- **4xx: Client Error** – This category of error status codes points the finger at clients.
- **5xx: Server Error** – The server takes responsibility for these error status codes.

Few Status codes

- 200 - OK
- 201- Created
- 400 - Bad Request
- 401-Unauthorized
- 402 - Payment Required
- 403 - Forbidden
- 404 - Not Found
- 417- Timeout Error
- 500 - Internal Server Error
- 503 - Service Unavailable
- 504 -Gateway Timeout



Thank You
