Crash Course:
POSTMAN

presented by: Sriya Annem

POSTMAN

Table of contents

What is POSTMAN?

Postman is a powerful tool for API development and testing, providing a user-friendly interface to send, manage, and analyze HTTP requests. It is used to streamline the process of creating, testing, and debugging APIs, as well as organizing and documenting them for better collaboration and development.

WORDS TO NOTE

API

(Application Programming Interface) allows different software systems to communicate by defining how requests and responses should be structured. It facilitates integration and interaction between applications.

HTTP

(Hypertext Transfer Protocol) is a protocol used for transmitting data over the web. It defines how messages are formatted and transmitted between clients and servers.

Endpoint

a specific URL or URI within an API that defines where requests are sent and where responses are received. It represents a particular function or resource within the API.

Requests

GET: Think of GET like asking someone to give you information. When you use GET, you're requesting data from a website or app, like asking to see a list of products.

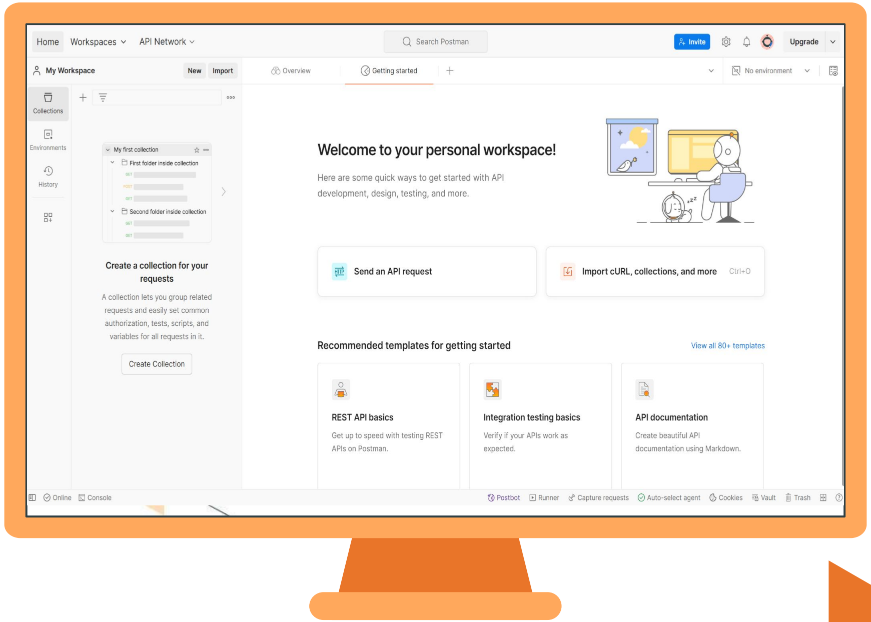
POST: POST is like sending a letter to someone with new information. When you use POST, you're sending data to a website or app to create something new, such as adding a new comment or submitting a form.

PUT: PUT is like updating an existing document. When you use PUT, you're sending data to change something that already exists, like updating your profile information.

DELETE: DELETE is like removing a file from a folder. When you use DELETE, you're asking to remove something from a website or app, such as deleting an old post or a file.

now you get to try!

Go to Postman.com



Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.

Welcome to your personal workspace!

Here are some quick ways to get started with API development, design, testing, and more.

Recommended templates for getting started

REST API basics

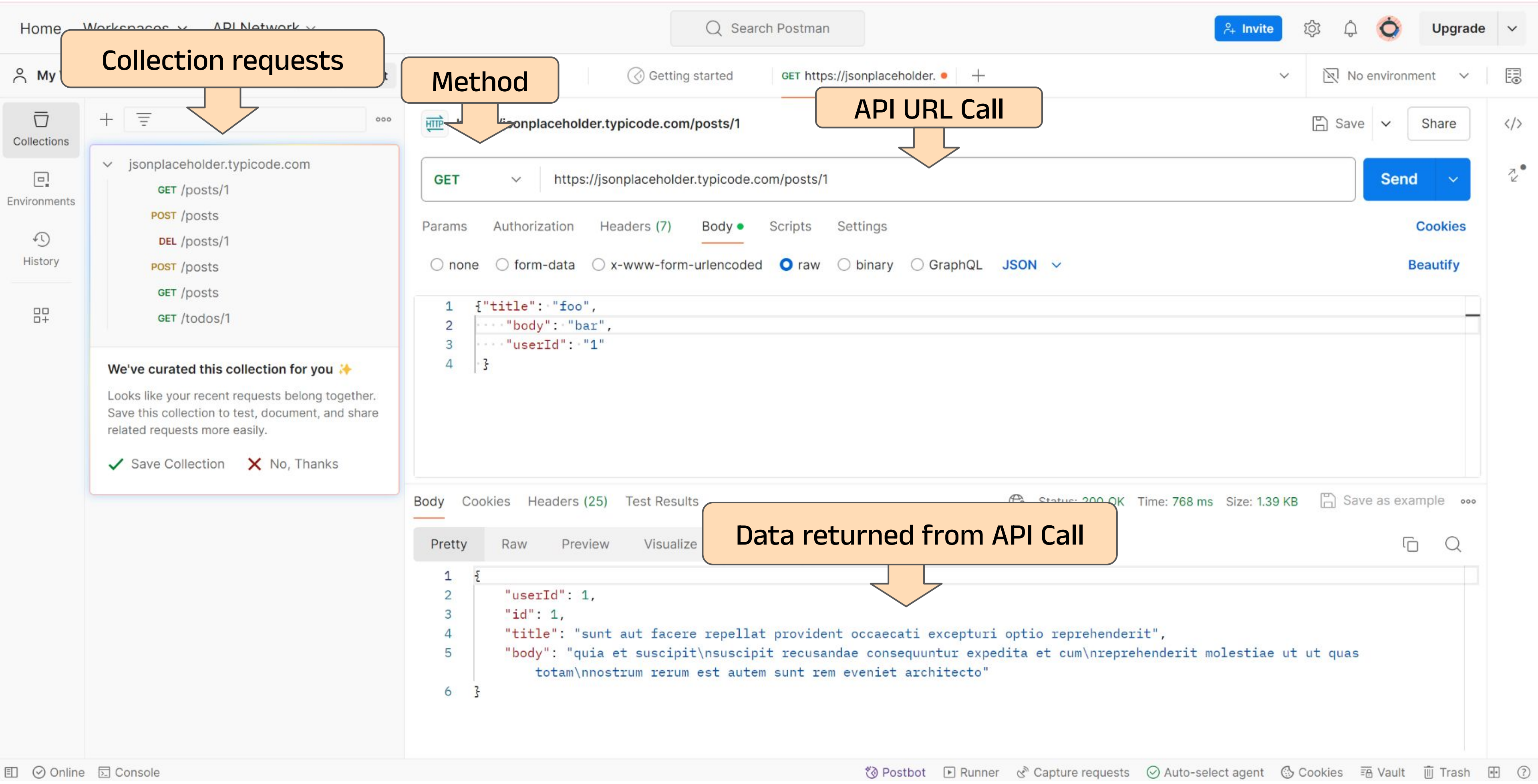
Get up to speed with testing REST APIs on Postman.

Integration testing basics

Verify if your APIs work as expected.

API documentation

Create beautiful API documentation using Markdown.



jsonplaceholder.typicode.com

- GET /posts/1
- POST /posts
- DEL /posts/1
- POST /posts
- GET /posts
- GET /todos/1

We've curated this collection for you ✨

Looks like your recent requests belong together. Save this collection to test, document, and share related requests more easily.

✓ Save Collection ✗ No, Thanks

GET https://jsonplaceholder.typicode.com/posts/1

```
{"title": "foo",
    "body": "bar",
    "userId": "1"
 }
```

```
{
    "userId": 1,
    "id": 1,
    "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
    "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas
        totam\nnostrum rerum est autem sunt rem eveniet architecto"
}
```

Getting a resource

```
fetch('https://jsonplaceholder.typicode.com/posts/1')
  .then((response) => response.json())
  .then((json) => console.log(json));
```

👇 *Output*

```
{
  id: 1,
  title: '...',
  body: '...',
  userId: 1
}
```

Listing all resources

```
fetch('https://jsonplaceholder.typicode.com/posts')
  .then((response) => response.json())
  .then((json) => console.log(json));
```

👇 *Output*

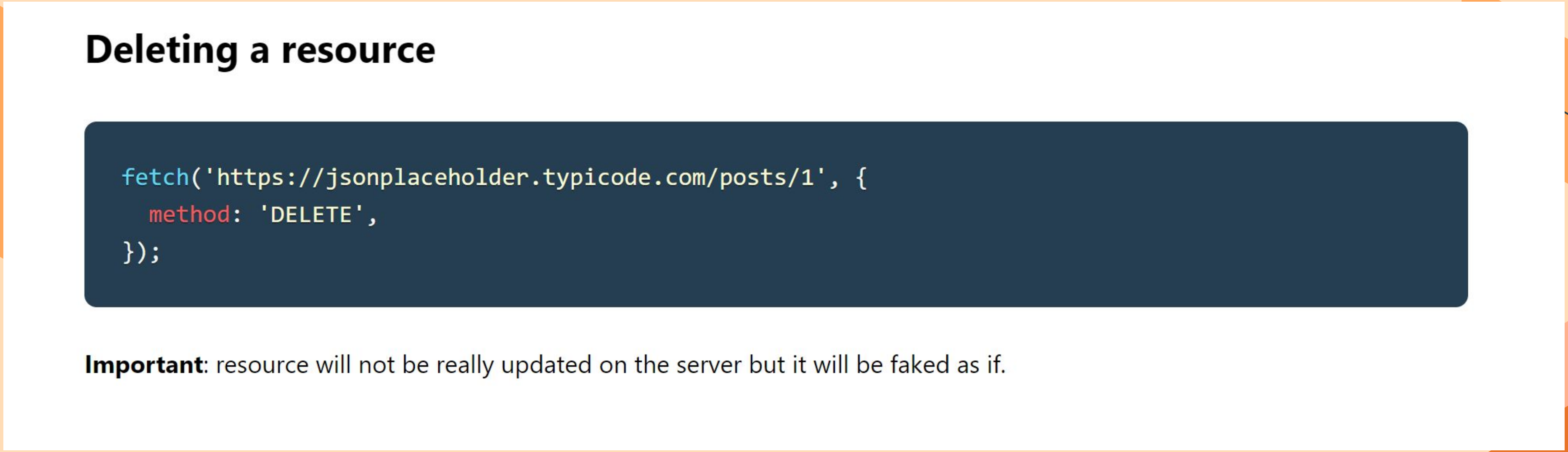
```
[
  { id: 1, title: '...' /* ... */ },
  { id: 2, title: '...' /* ... */ },
  { id: 3, title: '...' /* ... */ },
  /* ... */
  { id: 100, title: '...' /* ... */ },
];
```

Creating a resource

```
fetch('https://jsonplaceholder.typicode.com/posts', {
  method: 'POST',
  body: JSON.stringify({
    title: 'foo',
    body: 'bar',
    userId: 1,
  }),
  headers: {
    'Content-type': 'application/json; charset=UTF-8',
  },
})
  .then((response) => response.json())
  .then((json) => console.log(json));
```

👇 *Output*

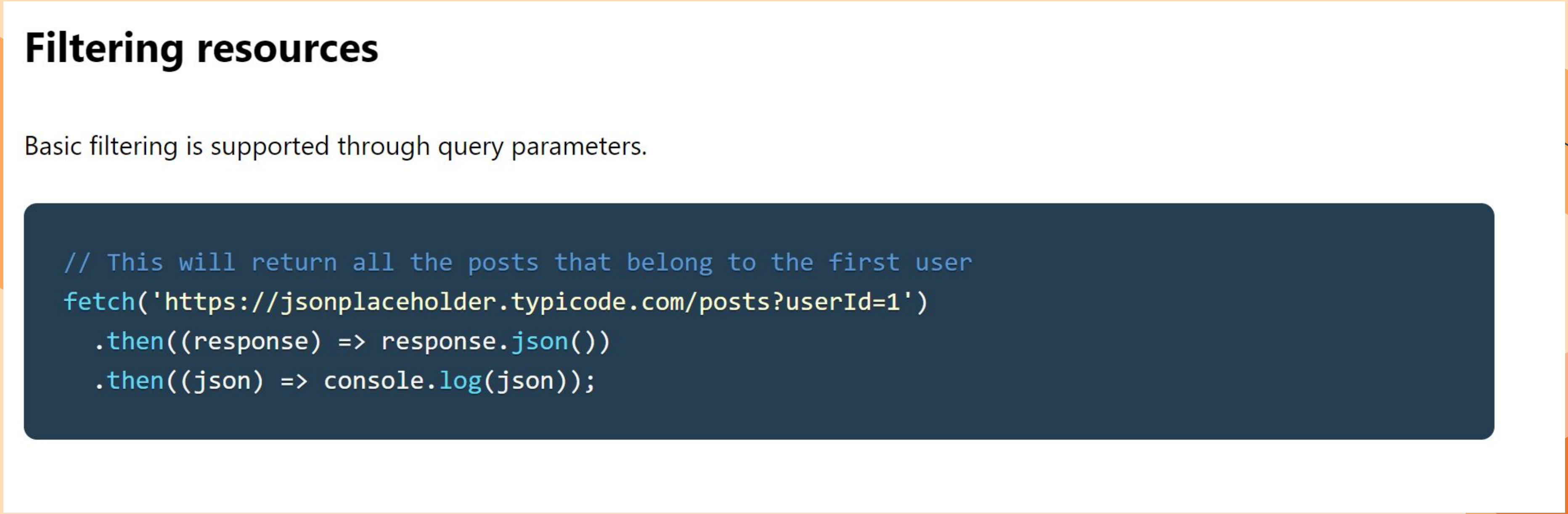
```
{
  id: 101,
  title: 'foo',
  body: 'bar',
  userId: 1
}
```



Deleting a resource

```
fetch('https://jsonplaceholder.typicode.com/posts/1', {
  method: 'DELETE',
});
```

Important: resource will not be really updated on the server but it will be faked as if.

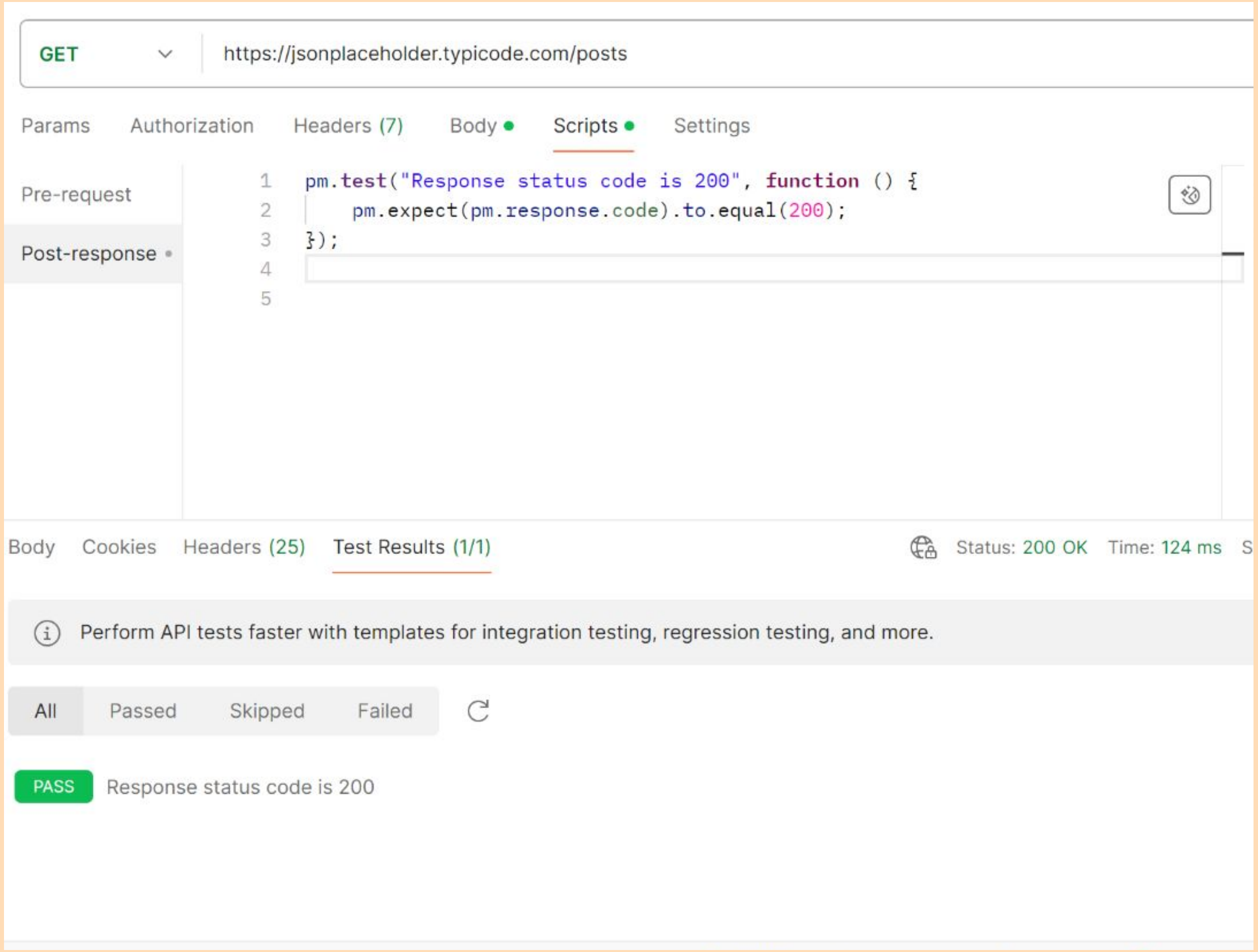


Filtering resources

Basic filtering is supported through query parameters.

```
// This will return all the posts that belong to the first user
fetch('https://jsonplaceholder.typicode.com/posts?userId=1')
  .then((response) => response.json())
  .then((json) => console.log(json));
```

Test Scripts



https://jsonplaceholder.typicode.com/posts

```
pm.test("Response status code is 200", function () {
    pm.expect(pm.response.code).to.equal(200);
});
```

Status: 200 OK Time: 124 ms

Perform API tests faster with templates for integration testing, regression testing, and more.

PASS Response status code is 200

1.29 BILLION

Individual requests in the past year.

30 MILLION

Registered users

500,000

Organizations that use Postman

THANK YOU!!

Please fill out the Website Meeting Feedback survey that will be sent out in the discord soon!