

COURSEWARE

Professional Skills
Agile Fundamentals
Jira
Git
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS
Javascript
Spring Boot
<div><div></div>Introduction to Spring Boot</div>
<div><div></div>Multi-Tier Architecture</div>
<div><div></div>Beans</div>
<div><div></div>Bean Scopes</div>
<div><div></div>Bean Validation</div>
<div><div></div>Dependency Injection</div>
<div><div></div>Components</div>
<div><div></div>Configuration</div>
<div><div></div>Connecting to a Database</div>
<div><div></div>Entities</div>
<div><div></div>Postman</div>
<div><div></div>Controllers</div>
<div><div></div>Services</div>
<div><div></div>Repositories</div>
<div><div></div>Custom Queries</div>
<div><div></div>Data Transfer Objects</div>
<div><div></div>Lombok</div>
<div><div></div>Custom Exceptions</div>
<div><div></div>Swagger</div>
<div><div></div>Profiles</div>
<div><div></div>Pre-Populating Databases for Testing</div>
<div><div></div>Unit testing with Mockito</div>

Postman

Contents

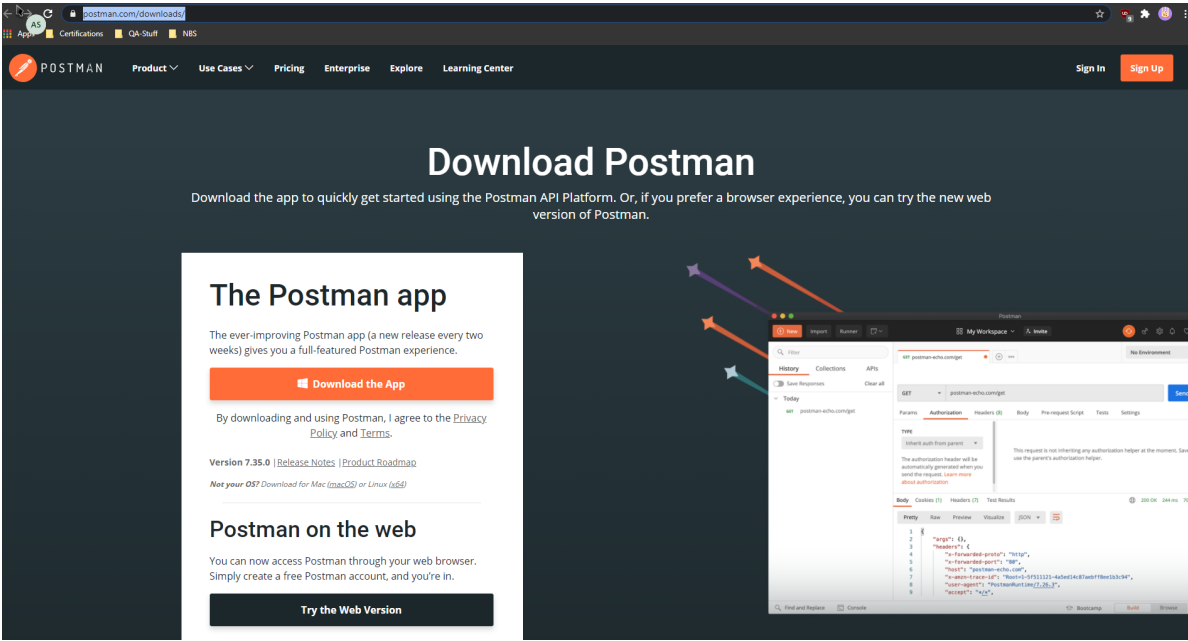
- [Overview](#)
  - [Installing Postman](#)
  - [How to use Postman](#)
- [Tutorial](#)
- [Exercises](#)

Overview

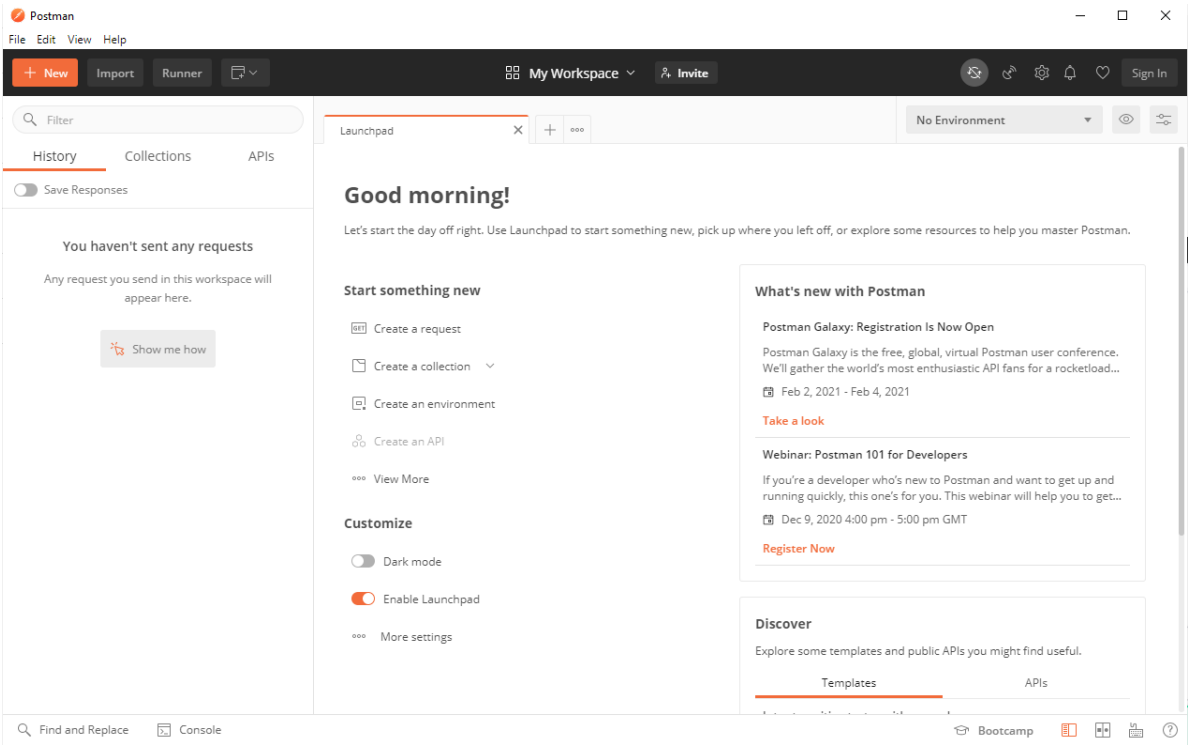
Postman is a standalone tool that utilises web APIs by making HTTP requests from outside the service.  
Postman allows you to test user created end-points efficiently.

Installing Postman

1. Download Postman from this [link](#)



2. Open the link and run Postman and you should see the following output:



How to use Postman

Postman works by sending HTTP requests to an end-point.  
In order to test the application, ensure that Spring is running or you have got a end-point to test, i.e. [a fake placeholder](#)

1. Click **create a request** on postman

<div><div></div><div>Testing</div></div>
Selenium
Sonarqube
Advanced Testing (Theory)
Cucumber
MongoDB
Express
NodeJS
React
Express-Testing
Networking
Security
Cloud Fundamentals
AWS Foundations
AWS Intermediate
Linux
DevOps
Jenkins Introduction
Jenkins Pipeline
Markdown
IDE Cheatsheet

2. Specify the type of request you're looking for i.e. GET/POST/DELETE

3. Enter the URL you're looking to get information from - this would either be your localhost or an external URL

GET

https://jsonplaceholder.typicode.com/todos/

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE	DESCRIPTION
	Key	Value	Description

4. Press 'send' and observe the output

GET

https://jsonplaceholder.typicode.com/todos/

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Body

Cookies (1)

Headers (26)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

[

2

{

3

"userId": 1,

4

"id": 1,

5

"title": "delectus aut autem",

6

"completed": false

7

},

8

{

9

"userId": 1,

10

"id": 2,

11

"title": "quis ut nam facilis et officia qui",

12

"completed": false

13

},

14

{

15

"userId": 1,

16

"id": 3,

17

"title": "fugiat veniam minus",

18

"completed": false

19

},

20

{

21

"userId": 1,

22

"id": 4,

23

"title": "et porro tempora",

24

"completed": true

25

},

26

{

27

"userId": 1,

28

"id": 5,

29

"title": "laboriosam mollitia et enim quasi adipisci quia provident illum",

30

"completed": false

31

},

32

{

33

"userId": 1,

34

"id": 6,

35

"title": "qui ullam ratione quibusdam voluptatem quia omnis",

36

"completed": false

## Tutorial

In this tutorial, we will be looking at creating a resource with Postman.

1. Specify the type of the request as POST

2. Add the URL endpoint in the request URL  
(<https://jsonplaceholder.typicode.com/posts>)

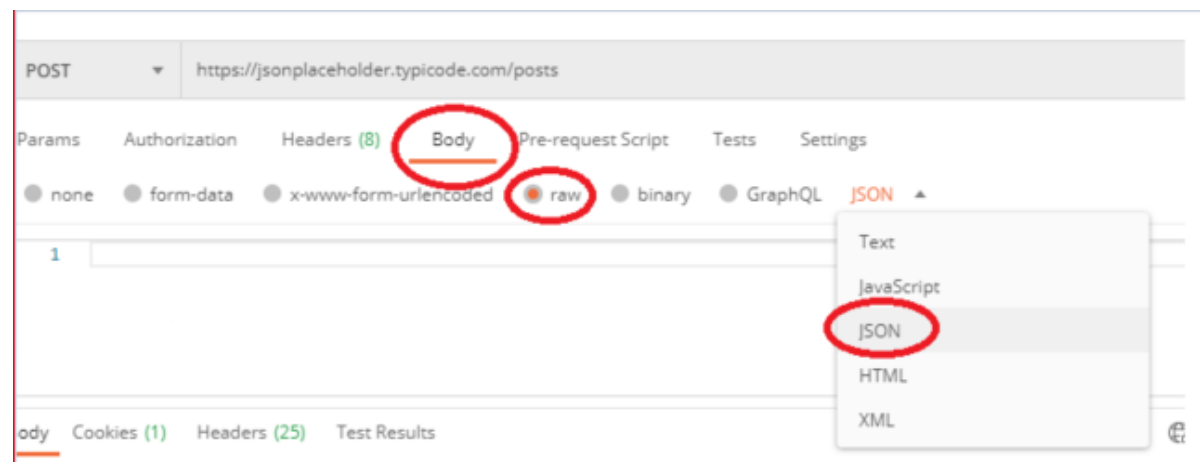
POST

https://jsonplaceholder.typicode.com/posts

3. Navigate to the "body" tab

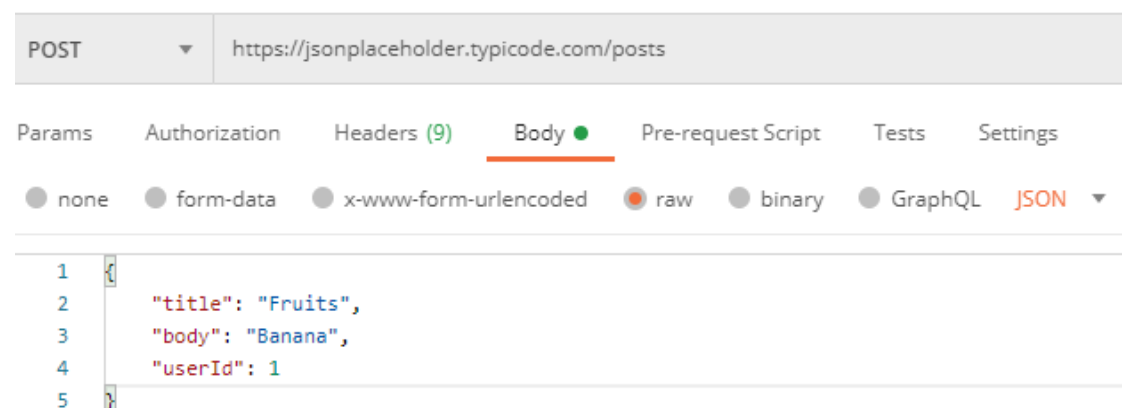
4. Select "raw"

5. From the drop-down on the right, select 'JSON'

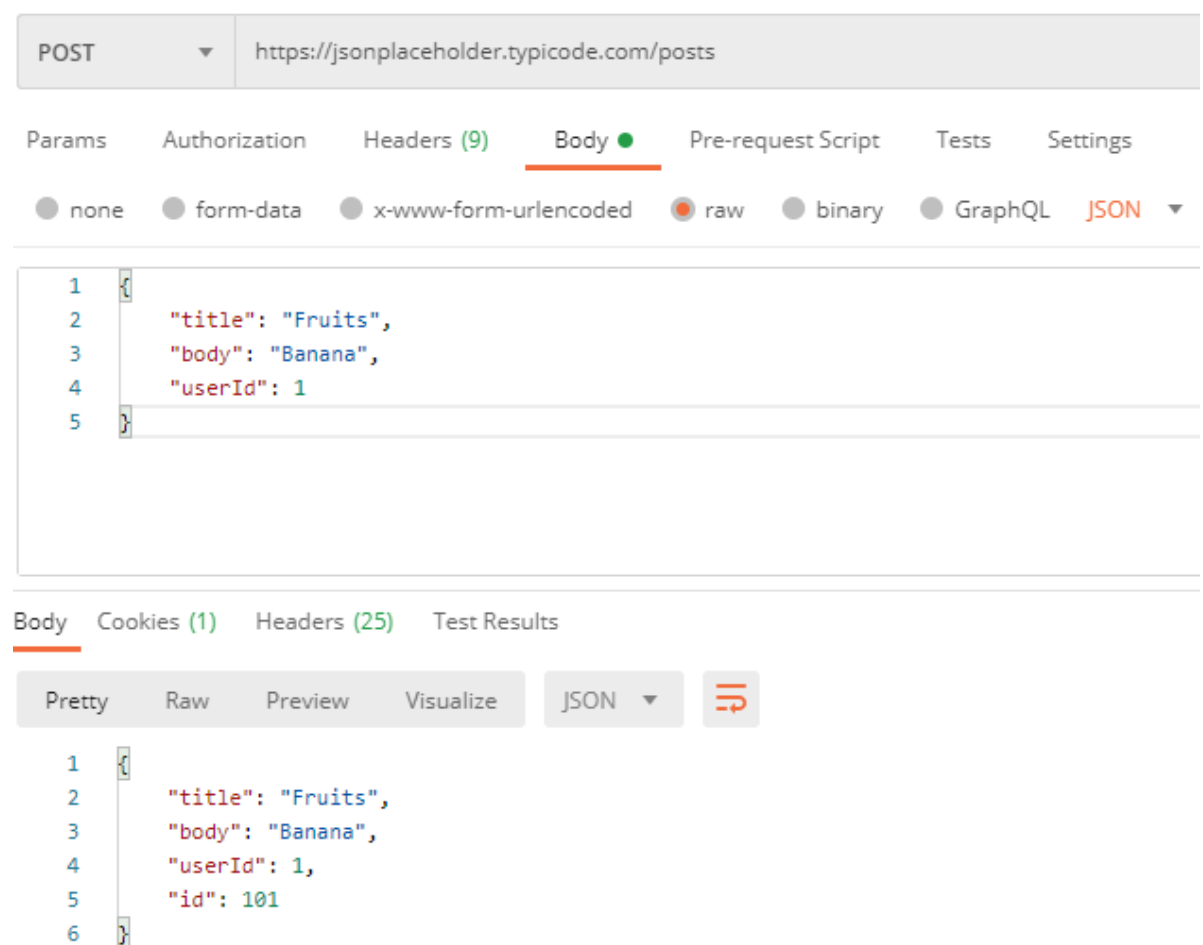


6. Add the following information to the request body as a JSON object:

- Title
- Body
- UserId



7. Press 'Send' and observe the output:



## Exercises

1. Update the existing code so that the title is - `{"title": "Something Else"}`