



9) HTTP &amp; Web Services Lesson

# POST Requests with Postman



6 min to complete · By Ryan Desmond, Jared Larsen

## Contents

- Introduction
- Making a POST Request
- Discussing the Response
- Summary: Postman POST Request

You've now learned how to make GET & DELETE requests to a server through a RESTful API. It's time to learn how to send new information to the server with POST requests. The internet is a collaborative medium, after all.

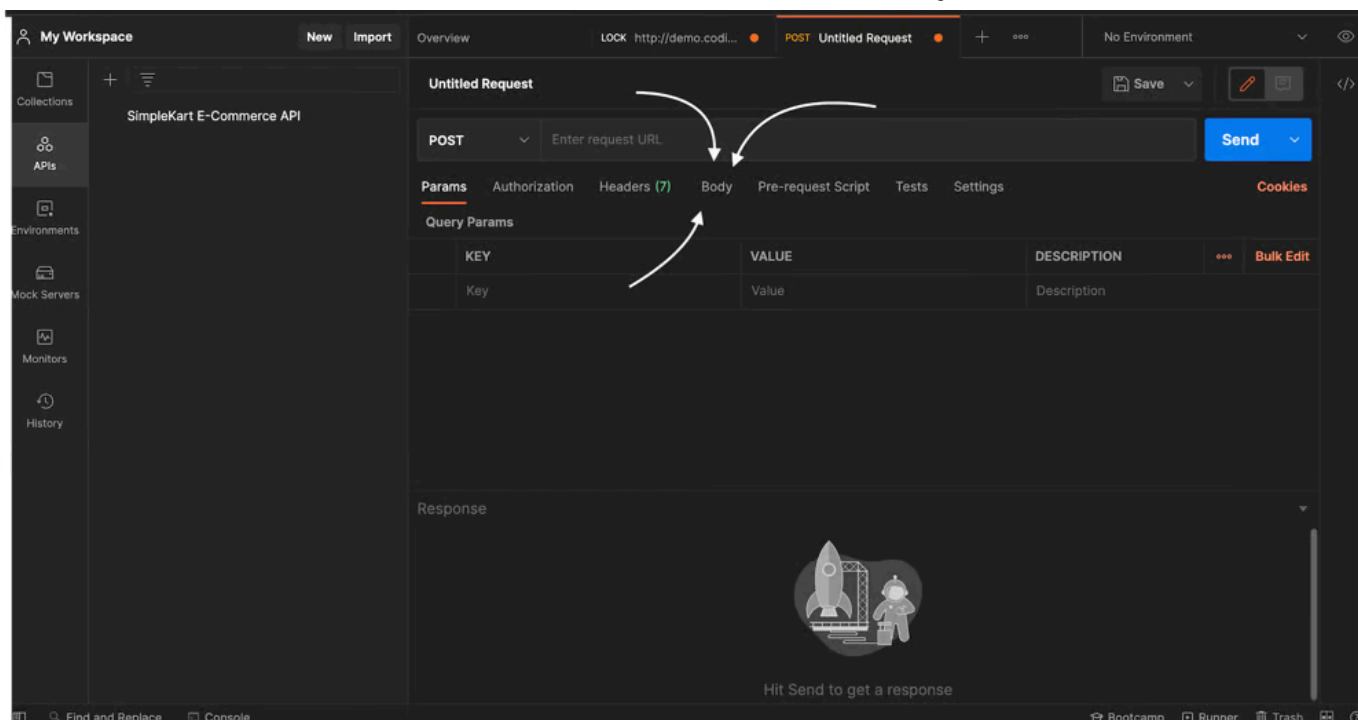
## Making a POST Request

Remember that a POST request sends data to the server and instructs it to create a new resource based on the data and endpoint you've entered. Here's an example that POSTs a new Task to the CodingNomads API using Postman:

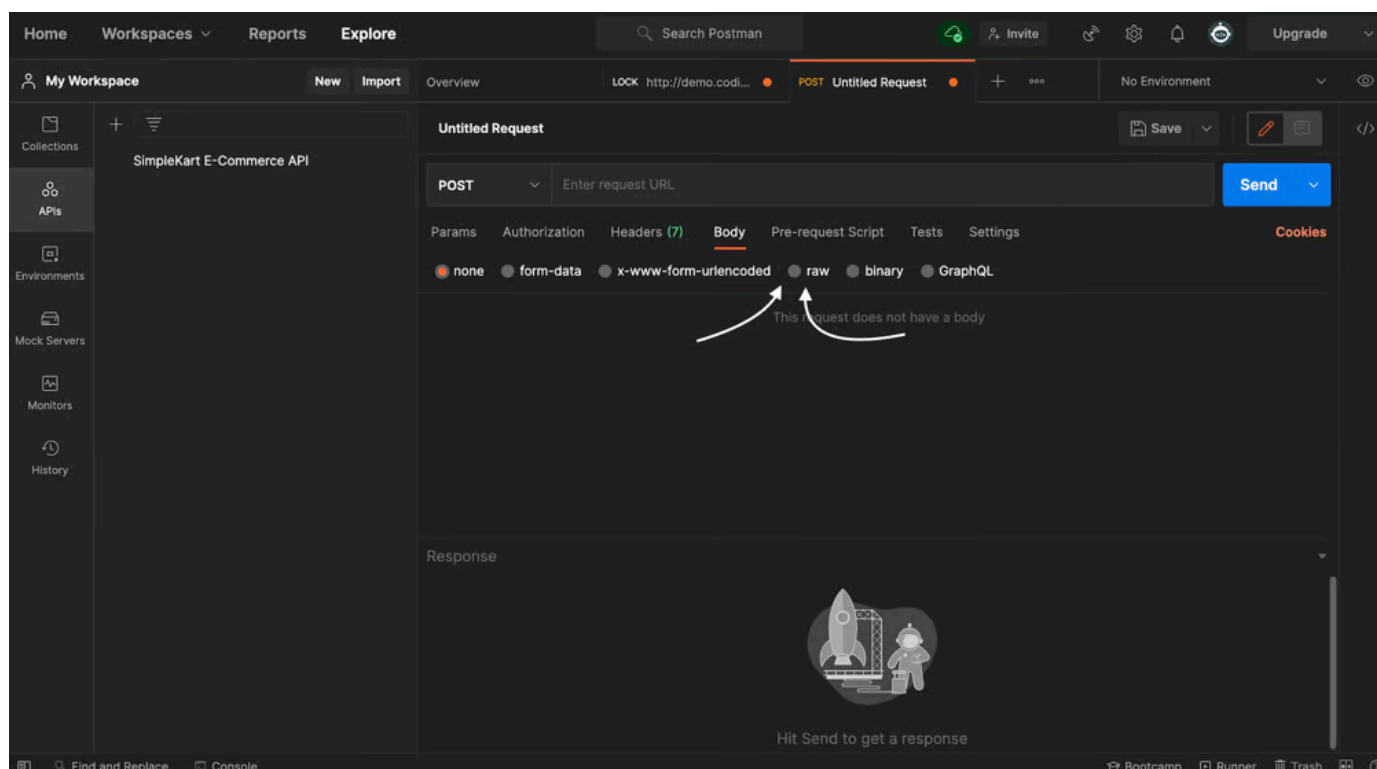
**Step 1:** Open a new tab.

**Step 2:** Change the HTTP method from `GET` to `POST`.

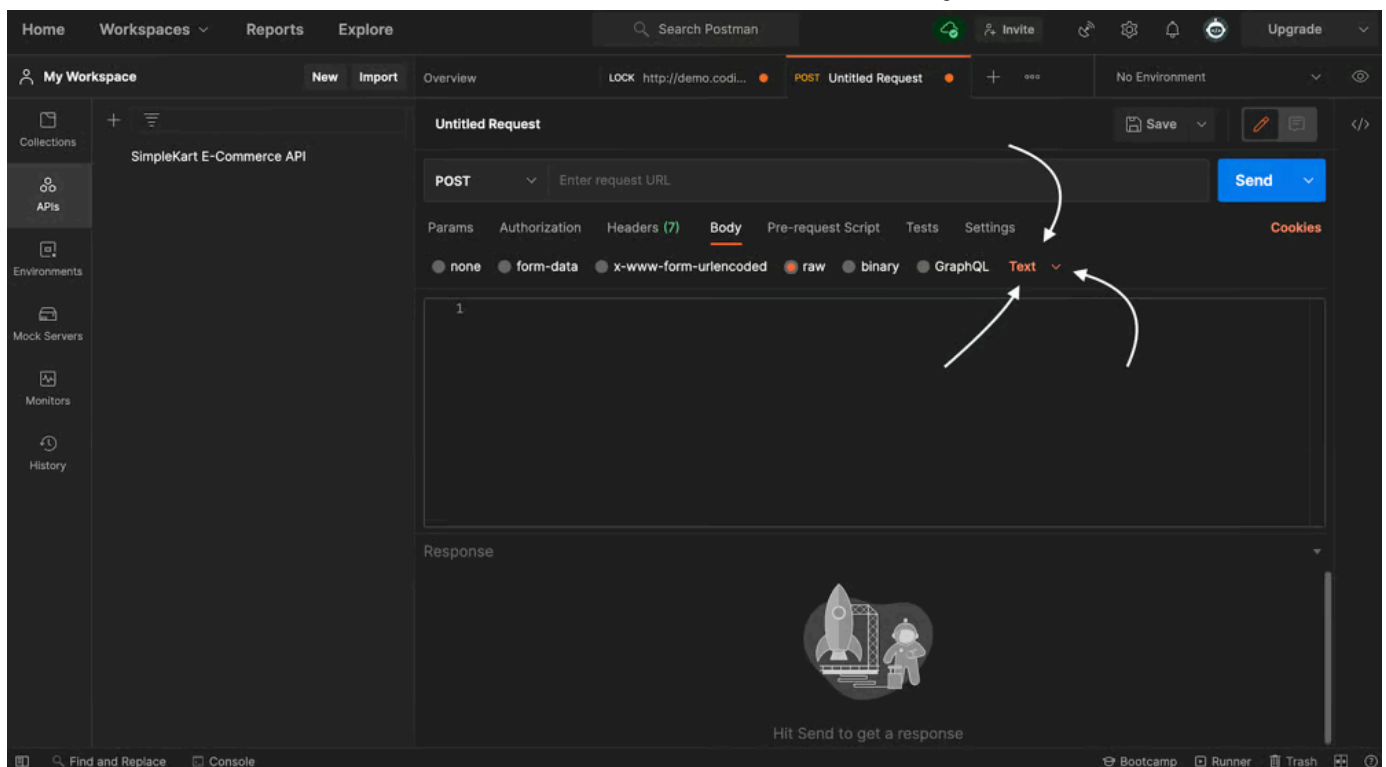
**Step 3:** Switch the editor to the body of the HTTP request. You can see how this works in the image below.



**Step 4:** Change the format from **none** to **raw**.



**Step 5:** Click the text button as shown and select JSON in the dropdown menu. This will make Postman auto-format your JSON.



**Step 6:** Copy this JSON Task object into the body section:

```
{  
  "userId": 3,  
  "name": "anything you want",  
  "description": "anything you want",  
  "completed": false  
}
```

**Step 7:** Enter the URL indicated by the API documentation. To make your life easier here it is:

[http://demo.codingnomads.co:8080/tasks\\_api/tasks](http://demo.codingnomads.co:8080/tasks_api/tasks)

**Step 8:** Hit enter or the blue 'Send' button and await the response.

## Discussing the Response

If the request was successful, you should see a result similar to the following appear at the bottom of the Postman window.

```
{
  "data": {
    "id": 86,
    "userId": 3,
    "name": "your Task name",
    "description": "your Task description",
    "createdAt": 1612023065720,
    "updatedAt": 1612023065720,
    "completed": false
  },
  "error": {
    "message": "none"
  },
  "status": "201 CREATED"
}
```



**Hint:** If you do not change the values for name and description and submit the request again, the server will throw an error since a task already exists with that name and description. Go ahead and try it!

There's a lot more information here than what you sent over. These other fields contain information the server was able to procure for itself. This happens frequently. Think about the Task object that was just created on the server - the constructor took in `userId`, `name`, `description`, and `completed`. The `createdAt` / `updatedAt` fields were automatically populated, and upon being successfully persisted to a database the `id` "86" was assigned. The server then decides this was a successful transaction and creates a response. Notice `error` and `statusCode` are not part of the Task object, but relate to the request itself.



**Info:** When in doubt, consult the API documentation. It should lay out the fields required to make the POST request. If there is no documentation, try making a GET request and use its response to model your POST request.

Learn by Doing



**Location:** [http://demo.codingnomads.co:8080/tasks\\_api/tasks](http://demo.codingnomads.co:8080/tasks_api/tasks)

- Using Postman, POST at least four new tasks to this API endpoint.
- Specify a different `userId`, `name`, and `description` each time.
- Now, submit a GET request to the same endpoint, and notice your new tasks in the response!

## Summary: Postman POST Request

Making a POST request can be complicated. Any number of errors can cause the request to fail. If you encounter problems, step back and try to run through the above steps again. Make sure the URL & HTTP method are correct. Confirm the data is correctly formatted, that you're not missing any fields, and that you are not including any fields the server may reject. Most importantly, use all the resources available for deciphering the API. Documentation is key.

### Online Spring Boot(camp) Fall Session!

**October 8–December 10, part-time.** Save \$250 by Sept. 19. Learn how we built CodingNomads with Java & Spring Boot in a 9-week online bootcamp. Code from day one with weekly live workshops, twice weekly 1:1 mentorship sessions, and hands-on projects. Join a small peer cohort, get tailored expert feedback, and graduate with an Advanced Java & Spring Framework [certificate](#). **Seats are limited!**



[Learn more](#)

[Previous](#)[Next → Video: POST Requests with Postman](#)

Want to go faster with dedicated 1-on-1 support? Enroll in a Bootcamp program.

[Learn more](#)

### Beginner - Intermediate Courses

[Java Programming](#)  
[Python Programming](#)  
[JavaScript Programming](#)  
[Git & GitHub](#)  
[SQL + Databases](#)

### Career Tracks

[Java Engineering Career Track](#)  
[Python Web Dev Career Track](#)  
[Data Science / ML Career Track](#)  
[Career Services](#)

### Intermediate - Advanced Courses

[Spring Framework](#)  
[Data Science + Machine Learning](#)  
[Deep Learning with Python](#)  
[Django Web Development](#)  
[Flask Web Development](#)

### Resources

[About CodingNomads](#)  
[Corporate Partnerships](#)  
[Contact us](#)  
[Blog](#)  
[Discord](#)