

Task: Get a book by id

Quiz #4: Query vs. path parameters

Sending data with POST

Task: Add a book

Task: Add an authorization header

Task: Use Postman Auth instead!

Quiz #5: Sending Data With Postman

Introduction to variables and scripting

Variables in Postman (Continued)

Setting variables programmatically

Task: Your first script

Task: Grab the new book id

Halfway Test

Quiz #6: Intro to Variables

PATCH and DELETE

Task: Checkout your book

Task: Delete your book

Generating code

Postman's codegen feature

Wrapping Up

Skillcheck (required)

Recap

Submit your Postman collection

Claim your badge!

Feedback

Task: Grab the new book id

Combining the power of variables and scripting gives you superpowers! Let's explore how you can automatically set a value for a variable via scripting.

Saving a value as a variable allows you to use it in other requests. Using a **Post-response** script, let's grab the **id** of a newly added book and save it so we can use it in future requests.

Setting and getting collection variables

The **pm** object allows you to set and get collection variables.

To **set** a collection variable, use the **.set()** method with two parameters: the variable name and the variable value

```
pm.collectionVariables.set("variableName", value)
```

To **get** a collection variable use the **.get()** method and specify the name of the variable you want to retrieve:

```
pm.collectionVariables.get("variableName")
```

Local variables

We can also store local variables inside our scripts using JavaScript.

There are two ways to define a variable in JavaScript: using the **const** or **let** keywords. **const** is for variables that won't change value, whereas **let** allows you to reassign the value later.

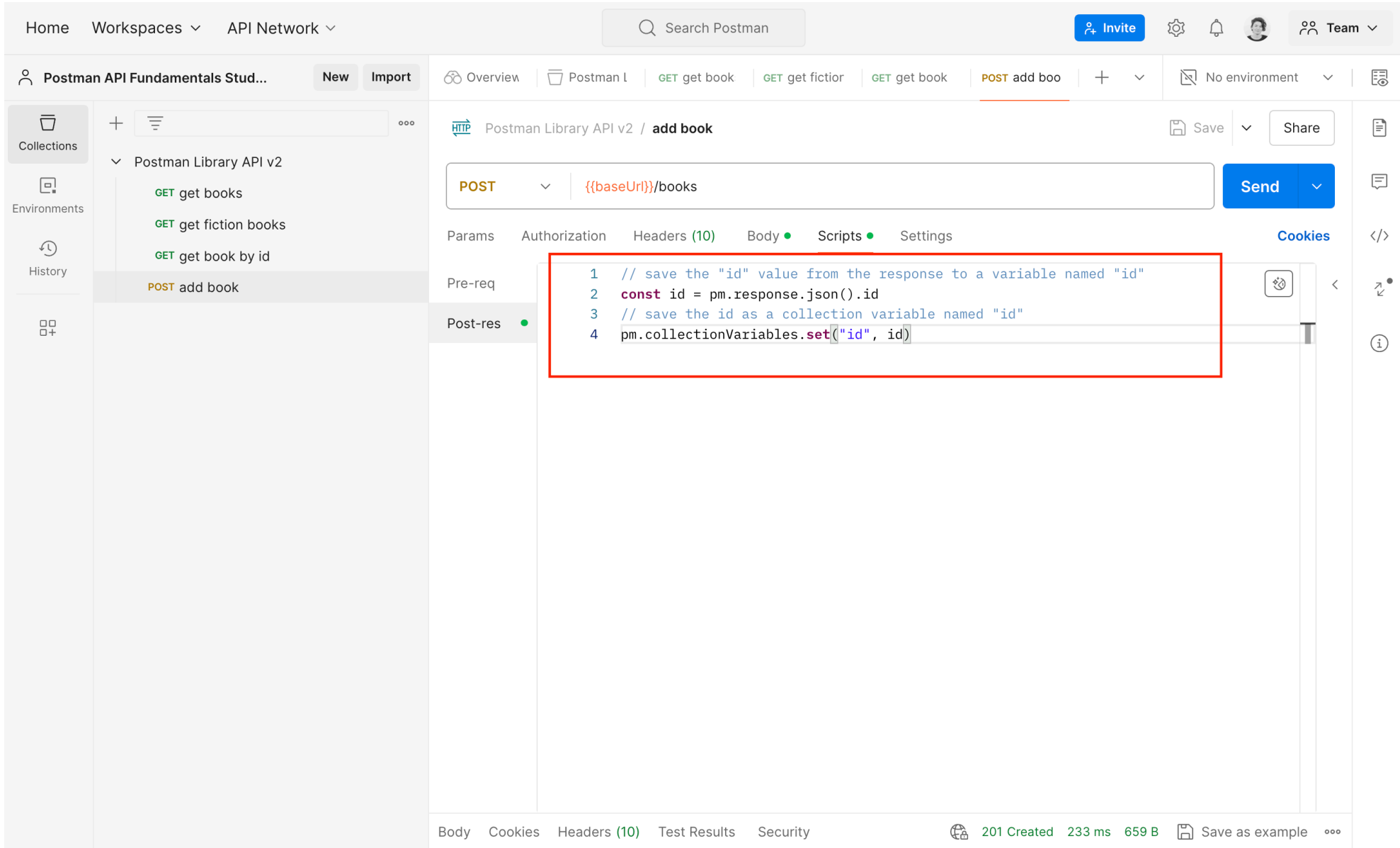
Set the new book **id** as a variable

- In the **Body** tab of the "add a book" request, **change the book's details to add a new book!**
- In the **Post-response** tab in **Scripts** of the "add a book" request, replace the console.log() statement with this code:

```
// save the "id" value from the response to a variable named "id"
const id = pm.response.json().id
// save the id as a collection variable named "id"
pm.collectionVariables.set("id", id)
```

The comments above that start with **//** help explain what the code is doing, and are ignored when we run the script.

Note: If there is no Collection variable named "id" Postman will create a new variable named id and assign the value.

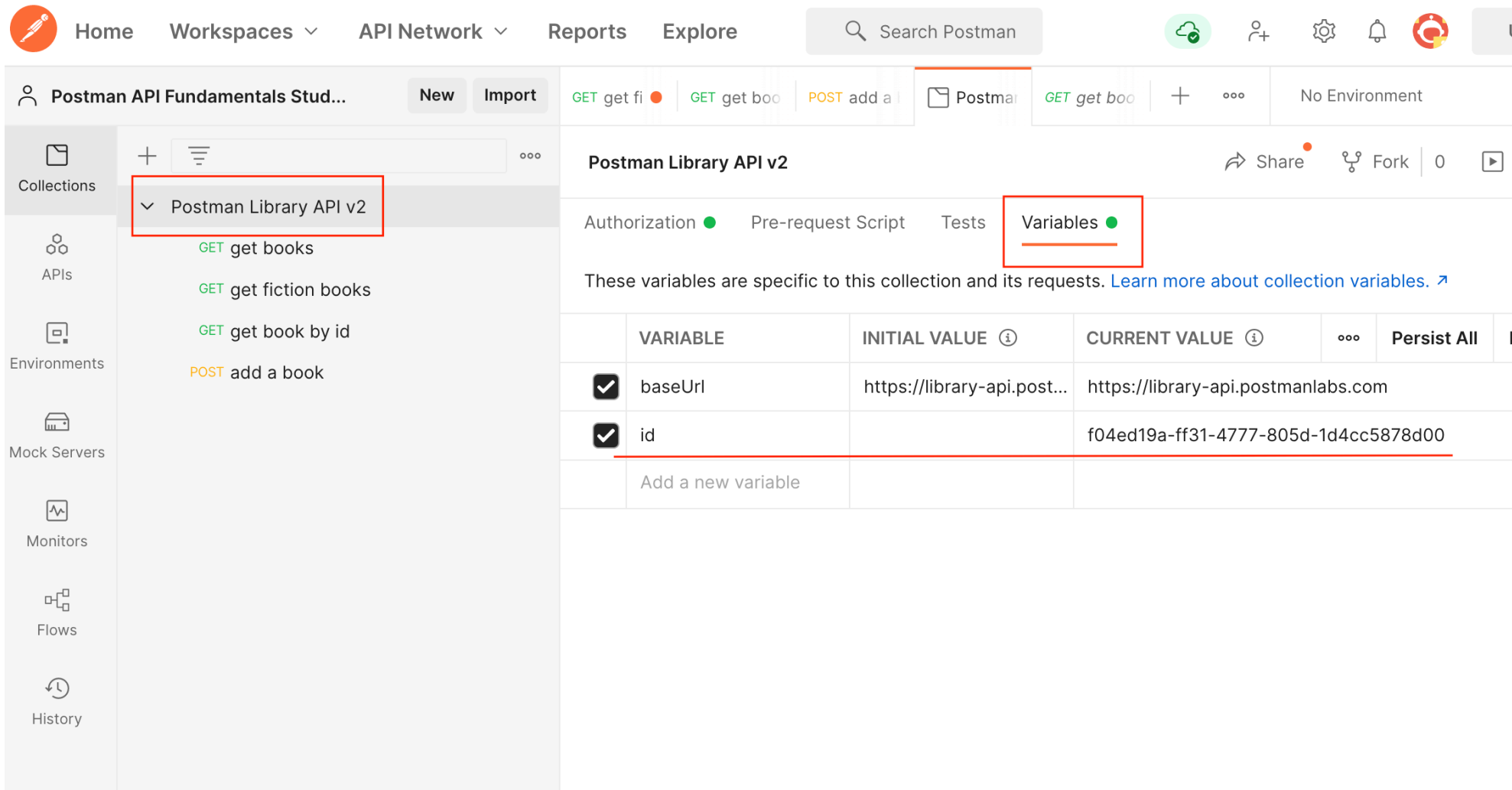


- Save and send** your request.

When the **201** the response comes back from the API with your newly created book, and the test script will run and save the book's **id** as a collection variable automatically.

- View your collection variables by clicking on your **Postman Library API v2** collection, then the **Variables** tab.

The **id** variable has been automatically assigned the id of your new book as its Current Value!



You can now use **{{id}}** anywhere in your collection to access this value! This will come in handy for our next request.