



- ✓ Task: Multiple query parameters
- ✓ Path Variable
- ✓ 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

Setting variables programmatically

Scripting in Postman

Postman allows you to add automation and dynamic behaviors to your collections with **scripting**.

Postman will automatically execute any provided scripts during two events in the request flow:

1. Immediately before a request is sent: **pre-request script** (**Pre-request Script** of Scripts tab).
2. Immediately after a response comes back: **post-response script** (**Post-response** of Scripts tab).

In this lesson, we will focus on writing scripts in the **Post-response** tab, which are executed when a response comes back from an API.

The **pm** object

Postman has a **helper object named pm** that gives you access to data about your Postman environment, requests, responses, variables and testing utilities.

For example, you can access the JSON response body from an API with:

```
pm.response.json()
```

You can also programmatically get collection variables like the value of **baseUrl** with:

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

In addition to getting variables, you can also set them with **pm.collectionVariables.set("variableName", "variableValue")** like this:

```
pm.collectionVariables.set("myVar", "foo")
```

Get ready!

In the next task, we will use scripting and the **pm** object to set a new book's automatically **id** as a collection variable so we can use the **id** in other requests.

