

Simple Books API Using Data-Driven Approach

The Simple Books API provides endpoints to manage book orders. Below are the steps for testing the API using data-driven testing with JSON/CSV input for dynamic parameters.

Step 1: Submit an Order

Description: Place a new book order.

- **Endpoint:**
POST <https://simple-books-api.glitch.me/orders>
- **Authentication:** Bearer Token
- **Request Body (JSON):**

```
{  
  "bookId": "{{BookID}}",  
  "customerName": "{{CustomerName}}"  
}
```

- **Post-response Validation:**
 - Check for status code 201.
 - Extract and store the **orderId** from the response for subsequent steps.

```
pm.test("Status code is 201", () => {  
  pm.response.to.have.status(201);  
});  
  
var jsonData = pm.response.json();  
pm.collectionVariables.set("orderId", jsonData.orderId);
```

Step 2: Get a Single Order

Description: Retrieve the details of a specific order.

- **Endpoint:**
GET <https://simple-books-api.glitch.me/orders/{{orderId}}>
- **Authentication:** Bearer Token
- **Post-response Validation:**
 - Check for status code 200.
 - Verify the orderId in the response matches the stored variable.

```
pm.test("Status code is 200", () => {
    pm.response.to.have.status(200);
});

pm.test("Check orderId present in the response body", () => {
    var jsonData = pm.response.json();
    pm.expect(jsonData.id).to.eql(pm.collectionVariables.get("orderId"));
});
```

Step 3: Remove an Order

Description: Delete an existing order.

- **Endpoint:**
DELETE <https://simple-books-api.glitch.me/orders/{{orderId}}>
- **Authentication:** Bearer Token
- **Post-response Validation:**
 - Check for status code 204.
 - Remove the orderId variable after successful deletion.

```
pm.test("Status code is 204", () => {
    pm.response.to.have.status(204);
});

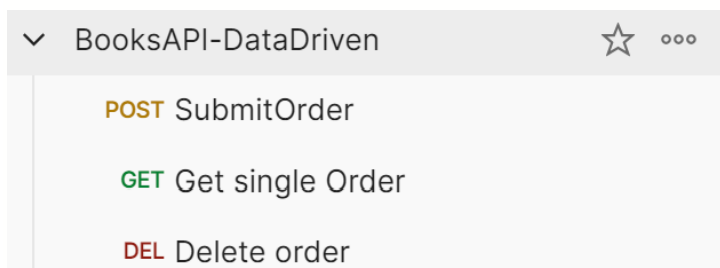
pm.collectionVariables.unset("orderId");
```

Key Notes:

- **Dynamic Data:** Use JSON or CSV to pass BookID and CustomerName for data-driven testing.
- **Chained Testing:** Each step relies on data from the previous step. Proper variable management is critical.
- **Postman Tests:** Automate response validation and variable handling using Postman scripts.

This approach ensures end-to-end validation of the API functionalities with reusable and dynamic test cases.

Run The collection in Postman



Runner

Run order

Deselect All | Select All | Reset

☒ POST SubmitOrder

☒ GET Get single Order

☒ DEL Delete order

Functional | Performance

Choose how to run your collection

☒ Run manually
Run this collection in the Collection Runner.

☐ Schedule runs
Periodically run collection at a specified time on the Postman Cloud.

☐ Automate runs via CLI
Configure CLI command to run on your build pipeline.

Run configuration

Iterations ⓘ
5

Delay ⓘ
0 ms

Data file ⓘ

Select File

orders_csv_data.csv ×

Data File Type
text/csv Preview

☐ Persist responses for a session ⓘ
☐ Turn off logs during run ⓘ

> Advanced settings

Run BooksAPI-DataDriven

<https://www.pavanonlinetrainings.com>

<https://www.youtube.com/@sdetpavan>