Integration testing

Get started here

Postman allows you to test your APIs using simple Javascript code. You can evaluate your response body, headers, cookies, and more using the ChaiJS BDD syntax.

This template guides you through the process of setting up an integration test to ensure that all individual components of an API function together seamlessly.

The API under test in this collection includes three endpoints for registering and receiving a token, accessing your unique generated name, and unregistering a token:

- POST `/register`
- POST `/unregister`
- GET `/my-name`

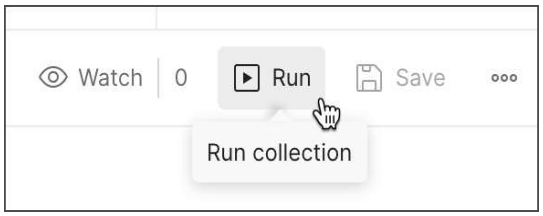
By setting up requests in the order of the operation, we can test the flow of data to and from the endpoints and ensure they work together as expected. We also verify that the data persists between requests on the back end.

How to use this template

Step 1: Check out the requests' documentation to learn more about -

- what each request is meant to do.
- the tests we've added against each one.

Step 2: Run this collection by clicking on "Run".



Step 3: To customize this template, replace the request URLs with your API endpoints and add or edit the tests if needed.

Resources

Scripting in Postman

Test script examples

Postman Sandbox API reference

POST Register

localhost:8080/admin/register

This returns a `token` that you can use to retrieve information later on.

We have included a test to confirm if a token is returned. We have also added test scripts to copy the token to the `token` collection variable. This makes it easy for us to reuse this token in other requests in the collection.

Body raw (json)

json

```
{
    "name":"gamma1",
    "email":"gamma1@gmail.com",
    "password":"gamma11@"
}
```

POST Login

localhost:8080/admin/login

This request uses the saved `token` collection variable to access a secret 'name' stored against that token.

In this request, we used the `token` collection variable as a query param to fetch the name generated for that token. We have added a test to check that a name is correctly returned.

Body raw (json)

json

```
{

    "email":"gamma1@gmail.com",
    "password":"gamma11@"
}
```

GET CategoryCreate

localhost:8080/category/all

This request un-registers a token by using the token collection variable in the POST body.

We also added a test to ensure the response has a 200 OK status code.

Category Create

Body raw (json)

```json
{
    "token": ""
}
```

GET CategoryAll

localhost:8080/category/all

To see the category based on category Type

Body raw (json)

```json
{
    "categoryType":"Sql"
}
```

POST CourseCreate

localhost:8080/course/create

Body raw (json)

```json
{
   "category":"java"
}
```

POST CourseView

localhost:8080/course/view

Body raw (json)

```json
{
    "category":"Python"
}
```

POST CourseViewId

localhost:8080/course/view/2

Course View as Per Id

Body raw (json)

```json
{
    "category":"Python"
}
```

POST FavouriteMake

localhost:8080/favourite/make

To Make the course Favourite like Star mark

Body raw (json)

```json
{
    "email":"gamma1@gmail.com",
    "courseId":"4"
}
```

POST FavouriteRemove

localhost:8080/favourite/remove

To Remove it from favourite

Body raw (json)

```json
{
    "email":"gamma1@gmail.com",
    "courseId":"2"
}
```