

Javier Alberto Romero Oviedo	Ingeniero de sistemas
Ejercicio práctico- Postman	Api https://github.com/vdespa/introduction-to-postman-course/blob/main/simple-books-api.md .

Ejercicio Práctico:

Valentin Despa es un desarrollador de software que está trabajando en un proyecto, en el cual hace un servicio Rest, para poder ver libros disponible, así como poder hacer ordenes de libros, así como actualización de estas órdenes, y hacer un borrado de una orden en caso de ser necesario. Por este motivo después de terminado el desarrollo del servicio el desarrollador quiere probar el correcto funcionamiento de este y para esto contrata a la empresa Choucair Testing.

1. Se debe crear un ambiente de pruebas para hacer la ejecución de nuestras pruebas
2. También se hace necesario el uso de variables desde el ambiente creado.
3. Poder hacer un crud con cada uno de los endpoints del servicio (incluye probar caminos felices y caminos no felices)
4. Cada una de las validaciones que se hacen deben estar respaldadas con aserciones que lo justifiquen, no solo validaciones de código de respuesta.

➤ ***Ejecución:***

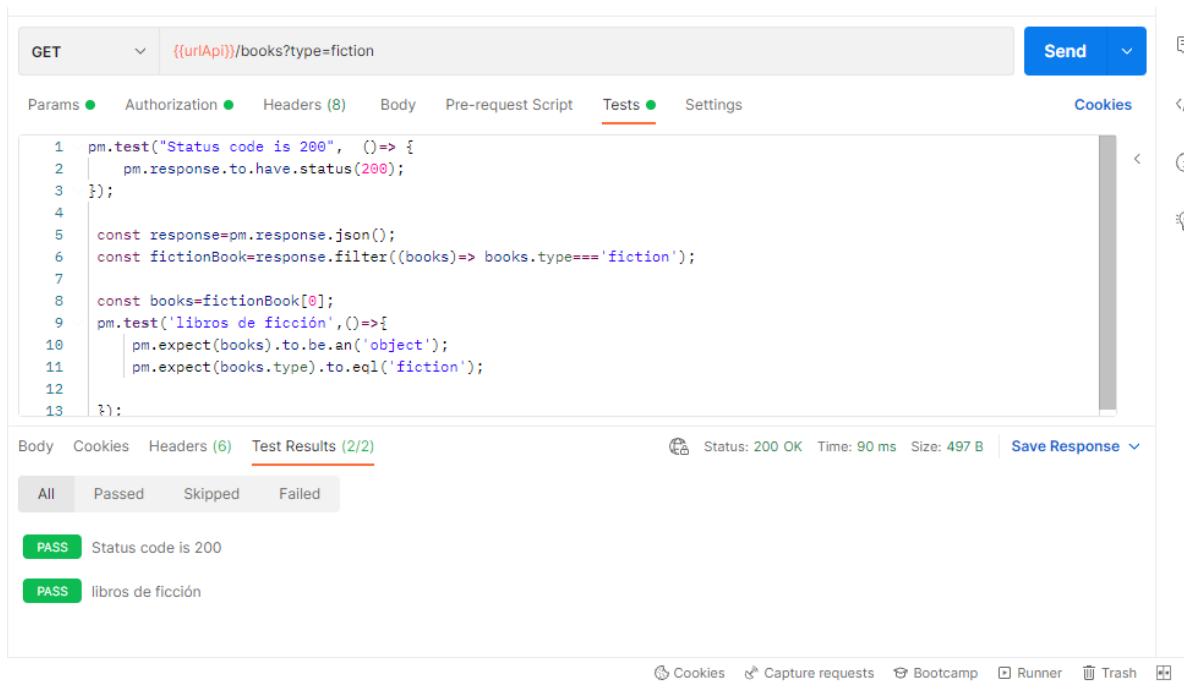
Devuelve el estado de la API.



Javier Alberto Romero Oviedo	Ingeniero de sistemas
Ejercicio práctico- Postman	Api https://github.com/vdespa/introduction-to-postman-course/blob/main/simple-books-api.md .

Devuelve una lista de libros

- Camino Feliz



GET `{{urlApi}}/books?type=fiction` Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

```

1 pm.test("Status code is 200", () => {
2   pm.response.to.have.status(200);
3 });
4
5 const response=pm.response.json();
6 const fictionBook=response.filter((books)=> books.type==='fiction');
7
8 const books=fictionBook[0];
9 pm.test('libros de ficción',()=>{
10   pm.expect(books).to.be.an('object');
11   pm.expect(books.type).to.eql('fiction');
12
13 });

```

Body Cookies Headers (6) Test Results (2/2) Status: 200 OK Time: 90 ms Size: 497 B Save Response

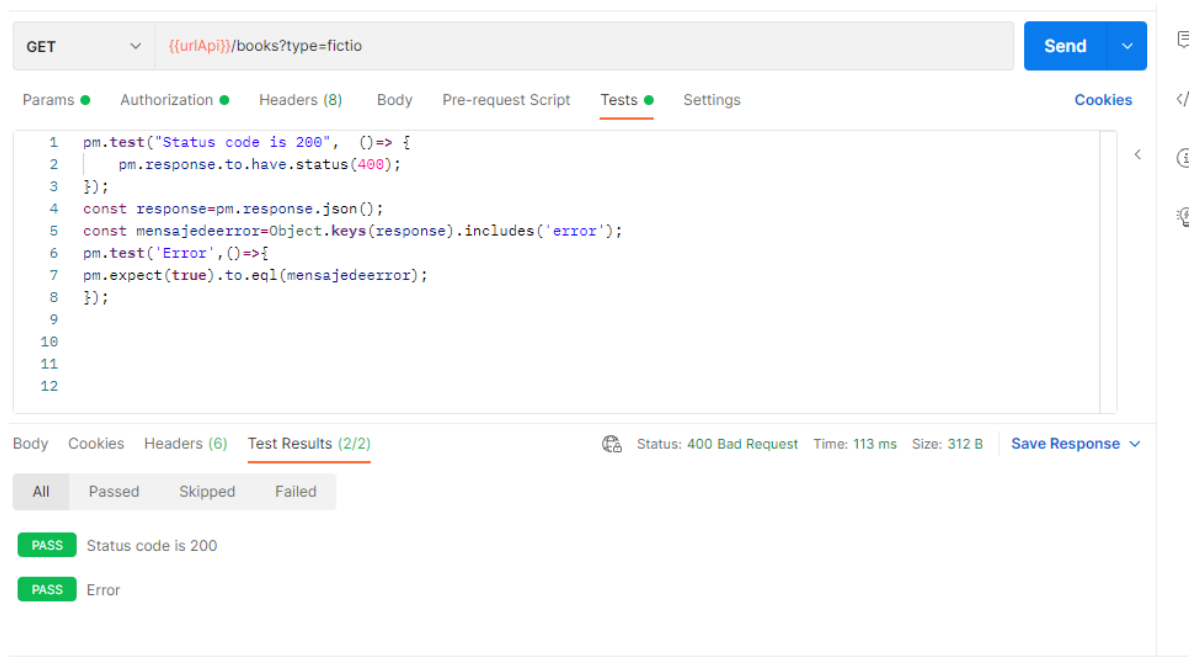
All Passed Skipped Failed

PASS Status code is 200

PASS libros de ficción

Cookies Capture requests Bootcamp Runner Trash

- Camino no Feliz



GET `{{urlApi}}/books?type=fictio` Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

```

1 pm.test("Status code is 200", () => {
2   pm.response.to.have.status(400);
3 });
4 const response=pm.response.json();
5 const mensajeerror=Object.keys(response).includes('error');
6 pm.test('Error',()=>{
7   pm.expect(true).to.eql(mensajeerror);
8 });
9
10
11
12

```

Body Cookies Headers (6) Test Results (2/2) Status: 400 Bad Request Time: 113 ms Size: 312 B Save Response

All Passed Skipped Failed

PASS Status code is 200

PASS Error

Javier Alberto Romero Oviedo	Ingeniero de sistemas
Ejercicio práctico- Postman	Api https://github.com/vdespa/introduction-to-postman-course/blob/main/simple-books-api.md .

Obtener un solo libro.

- *Camino Feliz*



GET `{{urlApi}}/books/:bookId` Send

Params ● Authorization Headers (7) Body Pre-request Script **Tests ●** Settings Cookies

```

1 pm.test("Status code is 200", ()=> {
2   pm.response.to.have.status(200);
3 });
4 const response=pm.response.json();
5 const bookId=Object.keys(response).includes('id');
6
7 pm.test('validar id', ()=>{
8   pm.expect(true).to.eql(bookId);
9 });
10

```

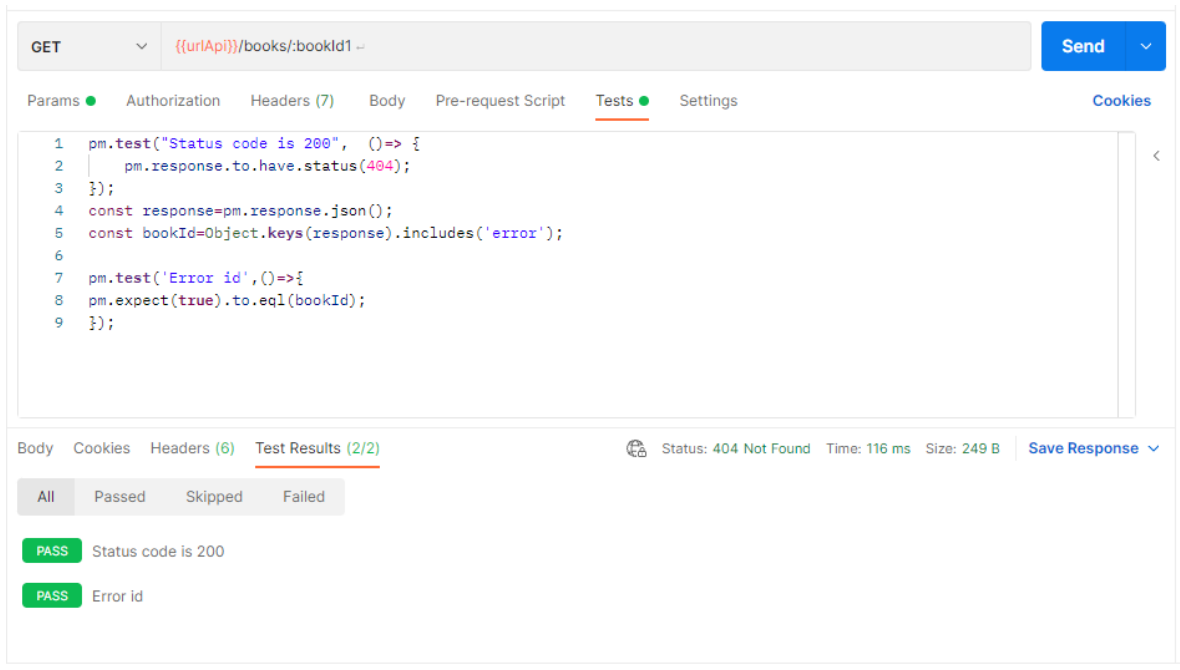
Body Cookies Headers (6) **Test Results (2/2)** 🔗 Status: 200 OK Time: 88 ms Size: 338 B Save Response ▼

All Passed Skipped Failed

PASS Status code is 200

PASS validar id

- *Camino no Feliz*



GET `{{urlApi}}/books/:bookId1` Send

Params ● Authorization Headers (7) Body Pre-request Script **Tests ●** Settings Cookies

```

1 pm.test("Status code is 200", ()=> {
2   pm.response.to.have.status(404);
3 });
4 const response=pm.response.json();
5 const bookId=Object.keys(response).includes('error');
6
7 pm.test('Error id', ()=>{
8   pm.expect(true).to.eql(bookId);
9 });

```

Body Cookies Headers (6) **Test Results (2/2)** 🔗 Status: 404 Not Found Time: 116 ms Size: 249 B Save Response ▼

All Passed Skipped Failed

PASS Status code is 200

PASS Error id

Javier Alberto Romero Oviedo	Ingeniero de sistemas
Ejercicio práctico- Postman	Api https://github.com/vdespa/introduction-to-postman-course/blob/main/simple-books-api.md .

Obtener

- *Camino Feliz*

Book / Crear Orden / Crear Orden

POST {{urlApi}}/orders Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

```

1 pm.test("Status code is 201", ()=> {
2   pm.response.to.have.status(201);
3 });
4 var respuesta = pm.response.json();
5 var validate = {
6   "items": {
7     "created": {"type": "boolean"},
8     "orderId": {"type": "string"}
9   }
10 };
11
12 pm.test('Validacion de la respuesta ', function () {
13   pm.expect(tv4.validate(respuesta, validate)).to.be.true;
14 });
15

```

Body Cookies Headers (6) Test Results (2/2) Status: 201 Created Time: 577 ms Size: 267 B Save Response

All Passed Skipped Failed

PASS Status code is 201

PASS Validacion de la respuesta

- *Camino no Feliz*

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

```

1 pm.test("Status code is 409", ()=> {
2   pm.response.to.have.status(409);
3 });
4
5 pm.test("ide faltante o no valido", function () {
6   pm.expect(pm.response.text()).to.include("Invalid or missing bookId.");
7 });
8

```

Body Cookies Headers (6) Test Results (2/2) Status: 409 Bad Request Time: 150 ms Size: 259 B Save Response

All Passed Skipped Failed

PASS Status code is 409

PASS ide faltante o no valido

Javier Alberto Romero Oviedo	Ingeniero de sistemas
Ejercicio práctico- Postman	Api https://github.com/vdespa/introduction-to-postman-course/blob/main/simple-books-api.md .

Obtener todos los pedidos

Book / Get / Obtener Todos los pedidos

GET `{{urlApi}}/orders` Send

Params Authorization Headers (8) Body Pre-request Script **Tests** Settings Cookies

```

1 pm.test("Status code is 200", ()=> {
2   pm.response.to.have.status(200);
3 });
4 pm.test("Buscar valores en el cuerpo", ()=> {
5   pm.expect(pm.response.text()).to.include("javier");
6 });
7

```

Body Cookies Headers (6) **Test Results (2/2)** Status: 200 OK Time: 95 ms Size: 1.83 KB Save Response

All Passed Skipped Failed

PASS Status code is 200

PASS Buscar valores en el cuerpo

Obtener un pedido los pedidos

- Camino feliz

Params Authorization Headers (8) Body Pre-request Script **Tests** Settings

```

1 pm.test("Status code is 200", ()=> {
2   pm.response.to.have.status(200);
3 });
4 const response=pm.response.json();
5 const idOrdenes=Object.keys(response).includes('id');
6
7 pm.test('validar id',()=>{
8   pm.expect(true).to.eql(idOrdenes);
9 });
10
11
12

```

Body Cookies Headers (6) **Test Results (2/2)** Status: 200 OK Time: 151 ms Siz

All Passed Skipped Failed

PASS Status code is 200

PASS validar id

Javier Alberto Romero Oviedo	Ingeniero de sistemas
Ejercicio práctico- Postman	Api https://github.com/vdespa/introduction-to-postman-course/blob/main/simple-books-api.md .

- Camino no feliz

Book / Get / Filtrar Pedidos Id Fallo

GET {{urlApi}}/orders/idOrdenes1 Send

Params Authorization Headers (8) Body Pre-request Script **Tests** Settings Cookies

```

1 pm.test("Status code is 404", ()=> {
2   pm.response.to.have.status(404);
3 });
4 const response=pm.response.json();
5 const pedidoId=Object.keys(response).includes('error');
6
7 pm.test('Error id',()=>{
8   pm.expect(true).to.eql(pedidoId);
9 });
10
11

```

Body Cookies Headers (6) **Test Results (2/2)** Status: 404 Not Found Time: 91 ms Size: 269 B Save Response

All Passed Skipped Failed

PASS Status code is 404

PASS Error id

Cookies Capture requests Bootcamp Runner Trash

Actualizar pedido

PUT {{urlApi}}/orders/idOrdenes Send

Params Authorization Headers (10) Body Pre-request Script **Tests** Settings Cookies

```

2
3   customerName: '',
4   createdBy: '',
5   quantity: ''
6 }
7
8 const body=JSON.parse(pm.request.body.raw);
9 const Actualizacion=JSON.stringify(Object.keys(actualizarpedido))===JSON.stringify(Object.keys(body));
10
11 pm.test('Actualizando pedido',()=>{
12   pm.expect(true).to.be.eql(Actualizacion)
13 });

```

Body Cookies Headers (7) **Test Results (1/1)** Status: 404 Not Found Time: 109 ms Size: 416 B Save Response

All Passed Skipped Failed

PASS Actualizando pedido

Cookies Capture requests Bootcamp Runner Trash

Javier Alberto Romero Oviedo	Ingeniero de sistemas
Ejercicio práctico- Postman	Api https://github.com/vdespa/introduction-to-postman-course/blob/main/simple-books-api.md .

Eliminar pedido

- Camino feliz

DELETE ▼ `{{urlApi}}/orders/idOrdenes` Send ▼

Params ● Authorization ● Headers (8) Body Pre-request Script Tests ● Settings Cookies

```

1 pm.test("Verificar código de estado 204, Petición DELETE Exitosa", function () {
2   pm.expect(pm.response.code).to.be.oneOf([204]);
3 });

```

Body Cookies Headers (3) Test Results (1/1) 🔗 Status: 204 No Content Time: 295 ms Size: 111 B Save Response ▼

All Passed Skipped Failed

PASS Verificar código de estado 204, Petición DELETE Exitosa

🍪 Cookies 📄 Capture requests 🏠 Bootcamp 🏃 Runner 🗑️ Trash 🔍

- Camino no feliz

DELETE ▼ `{{urlApi}}/orders/idOrdenes` Send ▼

Params ● Authorization ● Headers (8) Body Pre-request Script Tests ● Settings Cookies

```

1 pm.test("Verificar código de estado 204, Petición DELETE Exitosa", function () {
2   pm.expect(pm.response.code).to.be.oneOf([404]);
3 });
4 pm.test("Verificar Mensaje de error, con el Id de Orden N", function () {
5   var jsonData = pm.response.json();
6   q = pm.globals.get("idOrdenes");
7   pm.expect(jsonData.error).to.eql("No order with id "+q+".");
8 });

```

Body Cookies Headers (6) Test Results (2/2) 🔗 Status: 404 Not Found Time: 94 ms Size: 270 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼ 🔍

```

1 {
2   "error": "No order with id Rd1ruah-fTneR-39E1JnZ."
3 }

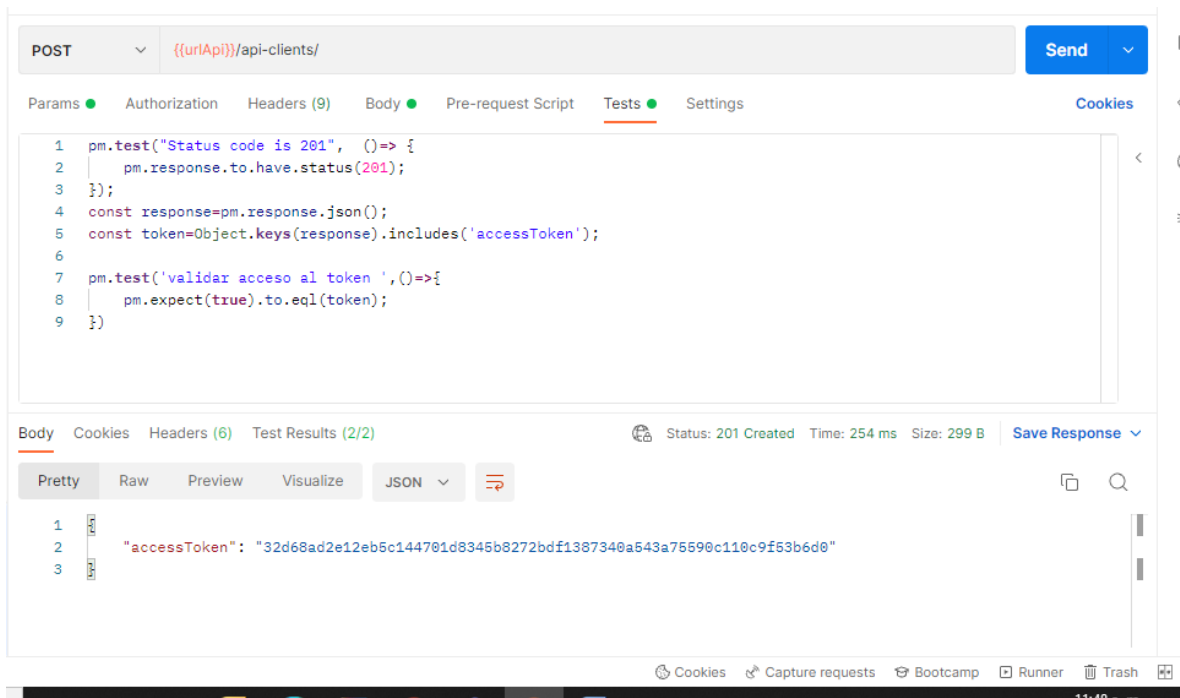
```

🍪 Cookies 📄 Capture requests 🏠 Bootcamp 🏃 Runner 🗑️ Trash 🔍

Javier Alberto Romero Oviedo	Ingeniero de sistemas
Ejercicio práctico- Postman	Api https://github.com/vdespa/introduction-to-postman-course/blob/main/simple-books-api.md .

Autenticación cliente

- Camino feliz



POST ▼ {{urlApi}}/api-clients/ Send ▼

Params ● Authorization Headers (9) Body ● Pre-request Script Tests ● Settings Cookies

```

1 pm.test("Status code is 201", ()=> {
2   pm.response.to.have.status(201);
3 });
4 const response=pm.response.json();
5 const token=Object.keys(response).includes('accessToken');
6
7 pm.test('validar acceso al token ',()=>{
8   pm.expect(true).to.eql(token);
9 })

```

Body Cookies Headers (6) Test Results (2/2) Status: 201 Created Time: 254 ms Size: 299 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼

```

1 {
2   "accessToken": "32d68ad2e12eb5c144781d8345b8272bdf1387340a543a75590c110c9f53b6d0"
3 }

```

11:40

- Camino no feliz



POST ▼ {{urlApi}}/api-clients/ Send ▼

Params ● Authorization Headers (9) Body ● Pre-request Script Tests ● Settings Cookies

```

1 pm.test("Status code is 409", ()=> {
2   pm.response.to.have.status(409);
3 });
4 const response=pm.response.json();
5 pm.test('Cliente ya registrado',()=>{
6   pm.expect(response.error).to.eql('API client already registered. Try a different email.');
```

7 }
8

Body Cookies Headers (6) Test Results (2/2) Status: 409 Conflict Time: 95 ms Size: 283 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼

```

1 {
2   "error": "API client already registered. Try a different email."
3 }

```

11:40