

Api desplegada en Heroku: <https://rhg-final-coder.herokuapp.com/>

Variable URL

- DEV: <http://localhost:3000>
- PROD: <https://rhg-final-coder.herokuapp.com>

Variables para simplificar las pruebas

Ecom Final

Share

Fork

0

Run

Save

...

Authorization

Pre-request Script

Tests

Variables

These variables are specific to this collection and its requests. [Learn more about collection variables.](#)

	VARIABLE	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ	...	Persist All	Reset All
<input checked="" type="checkbox"/>	cart_to_finish	630df8fac3a4c99bc22a0fd7	630df8fac3a4c99bc22a0fd7			
<input checked="" type="checkbox"/>	order_to_find	6310c5d0a1750abf548fe18	6313d4d144baedfd410a6ffe			
<input checked="" type="checkbox"/>	cart_to_find	631664b0c9068b39ca5b2e90	631664b0c9068b39ca5b2e90			
<input checked="" type="checkbox"/>	Email_to_find	rubenhg@gmail.com	rubenhg@gmail.com			
<input checked="" type="checkbox"/>	token-a	eyJhbGciOiJIUzI1NiIsInR5cCI6Ikp...	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6InYmVuaGdAZ21haWw...			
	Add a new variable					

Auth

- Register: (/user/register) crea un usuario con un objeto en el body como parametro
- Login: (/user/login) genera el token para el usuario registrado

El token resultante se debe establecer en la variable “token-a” de la colección para que los request de los otros endpoints sean autenticados.

Body

Cookies

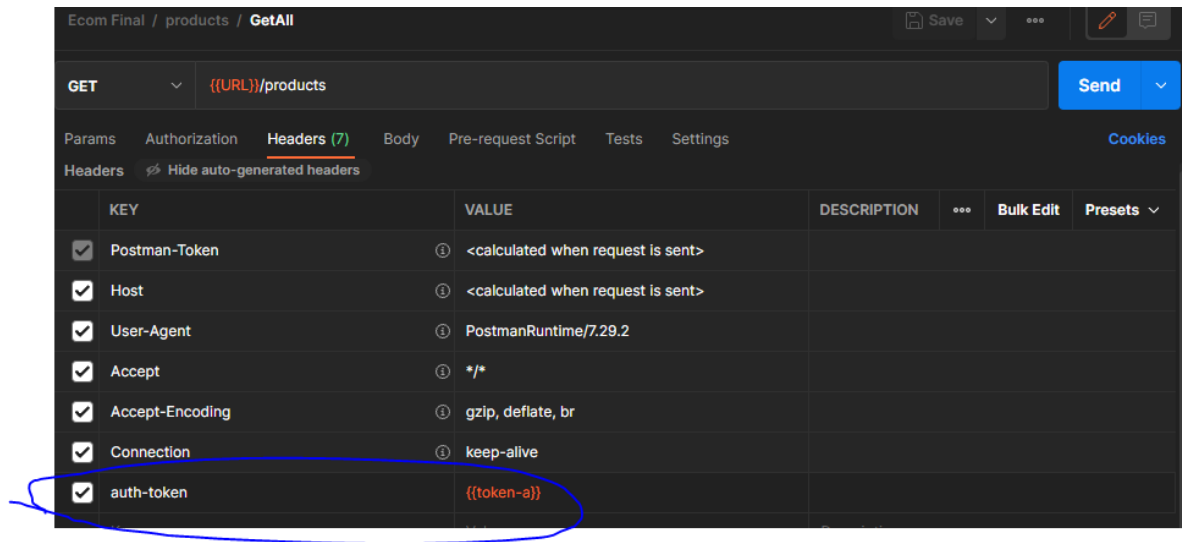
Headers (8)

Test Results

Status: 200 OKTime: 266 msSize: 465 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {  
2   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
      eyJlbnRpbiCI6InJ1YmVuaGdAZ2Z1haWwUY29tIiwiaWQiOiI2MzE2NWVhYmUxMTBINTllMWw5YTRmNjkiLCJpcyYXQiojE2NjjiOUMTQ4NzR9.  
      BitEscetVYWlMtZUqdm748rYuaZOLu0l2YkISzrUW08o"  
3 }
```



Products

- GetAll (/products) devuelve un array de objetos con todos los productos
- GetById (/products/{{product_to_find}}) devuelve un producto por su Id
- GetByCategory(/products/category/{{category_to_find}}) devuelve la lista de productos con la category indicada como parámetro
- Delete (/products/{{product_to_delete}}) elimina el producto con el id indicada como parámetro
- POST /products crear un producto con el objeto enviado como parámetro
- PUT (/products) modifica el producto con los datos enviados en un objeto como parámetro
- DELETE (/products/{{product_to_delete}}) elimina el producto con el id enviado como parámetro (devuelve el id del producto borrado)

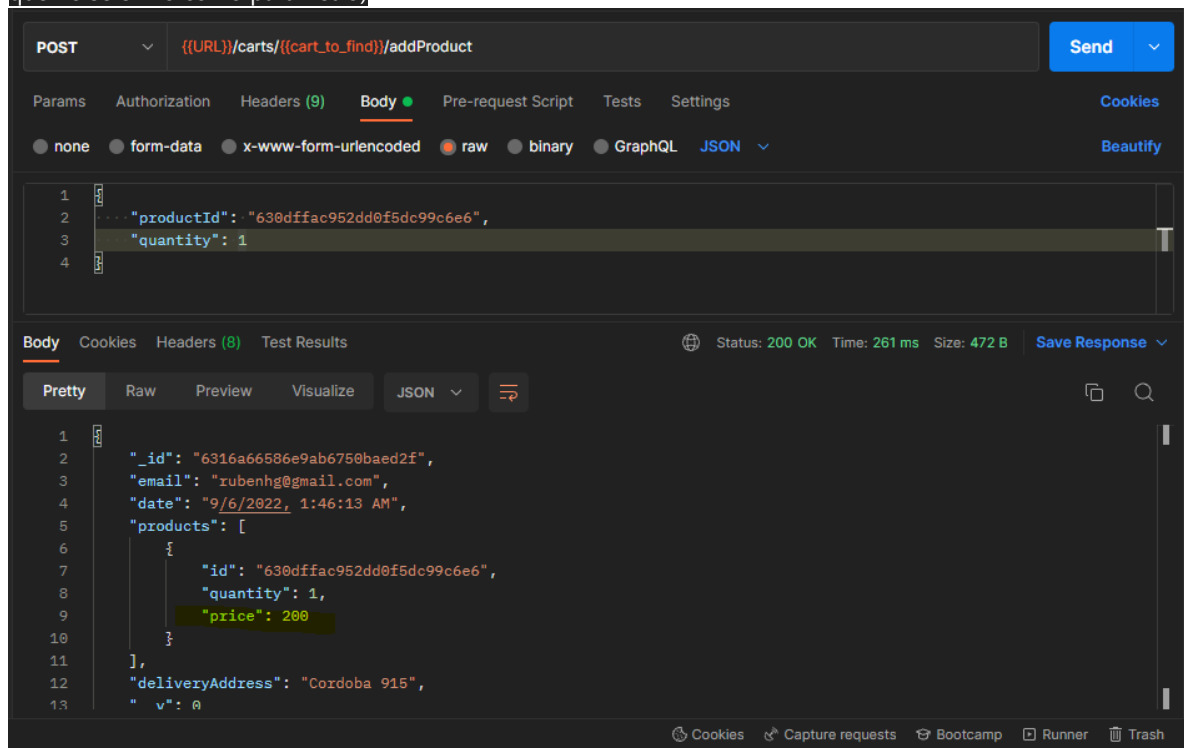
Carts

Se exponen endpoints para el CRUD de Cart, se explican las operaciones no triviales:

Create POST /carts: crea el carrito con los datos enviados con un objeto como parámetro

AddProduct ({{URL}}/carts/{{cart_to_find}}/addProduct) agrega un producto al carrito especificado, envia como parámetro un objeto con el id del producto y la cantidad. Devuelve el objeto cart (notose que incluye el Price

que no se envío como parámetro)



DeleteProduct (DELETE `{{URL}}/carts/{{cart_to_find}}/product/{{product_to_delete_from_cart}}`) elimina el producto del carrito.

GetProductsByCart (`{{URL}}/carts/{{cart_to_find}}/products`) obtiene los productos del carrito (incluyendo los datos actuales del catalogo de productos y el precio al momento de agregar el producto)