

Servidor API

Descripción

Servidor API para gestionar inventario de productos (Northwind - tabla “products”).

Request

ENDPOINT	http://localhost:8080/api/product
MÉTODO	GET
PARÁMETROS	-

Response

RESPUESTA OK	HEADER	HTTP/ 200 OK
	CONTENT-TYPE	application/json
	BODY	<pre>{ "productoid": 1, "nombreProducto": "Chai", "proveedorId": 1, "categoriaId": 1, "cantidadPorUnidad": "10 boxes x 20 bags", "precioUnidad": 18.00, }, { "productoid": 2, "nombreProducto": "Chang", "proveedorId": 1, "categoriaId": 1, "cantidadPorUnidad": "24 - 12 oz bottles", "precioUnidad": 19.00, } ...</pre>

RESPUESTA ERROR	HEADER	HTTP/ 401 UNAUTHORIZED
	BODY	{ "timestamp": "2025-11-13T01:37:27.555+00:00", "status": 401, "error": "Unauthorized", "path": "/api/product/" }

Server API 1 - Ejemplo

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections' (selected), 'Environments', 'Flows', and 'History'. The main area shows a collection named 'CESAR FABRICIO TIRADO VERA's ...'. A 'GET Get data' request is selected, with the URL 'localhost:8080/api/product'. The 'Authorization' tab is active, showing 'Basic Auth' with 'Username' set to 'admin' and 'Password' set to '.....'. The response section shows a 200 OK status with a JSON body containing two product objects:

```
1 [  
2 {  
3     "productoId": 1,  
4     "nombreProducto": "Chais",  
5     "proveedorId": 1,  
6     "categoriaId": 1,  
7     "cantidadPorUnidad": "10 boxes x 20 bags",  
8     "precioUnidad": 18  
9 },  
10 {  
11     "productoId": 2,  
12     "nombreProducto": "Chang",  
13     "proveedorId": 1,  
14     "categoriaId": 1,  
15     "cantidadPorUnidad": "24 - 12 oz bottles",  
16     "precioUnidad": 19  
17 },  
18 ]
```

At the bottom, there are buttons for 'Console', 'Runner', 'Start Proxy', 'Cookies', 'Vault', 'Trash', and a help icon.

Home Workspaces API Network

Search Postman Ctrl K

Invite Upgrade

CESAR FABRICIO TIRADO VERA's ... New Import

GET Get data POST Post data PUT localhost:8080/api/proc DEL localhost:8080/api/proc + No environment

Collections Environments Flows History

My Collection

GET Get data POST Post data

HTTP My Collection / Get data

GET localhost:8080/api/product

Params Authorization Headers (7) Body Scripts Settings Cookies

Auth Type Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about [Basic Auth](#) authorization.

Username Password

Body Cookies Headers (12) Test Results (0/1)

401 Unauthorized 79 ms 486 B Save Response

{ } JSON Preview Debug with AI

```
1 {  
2   "timestamp": "2025-11-13T02:35:57.995+00:00",  
3   "status": 401,  
4   "error": "Unauthorized",  
5   "path": "/api/product"  
6 }
```

Console Runner Start Proxy Cookies Vault Trash

Request

ENDPOINT	http://localhost:8080/api/product/{id}
MÉTODO	GET
PARÁMETROS	Parámetro de ruta: "id" (entero)

Response

RESPUESTA OK	HEADER	HTTP/ 200 OK
	CONTENT-TYPE	application/json
	BODY	{ "productId": 1, "nombreProducto": "Chais", "proveedorId": 1, "categoriald": 1, "cantidadPorUnidad": "10 boxes x 20 bags", "precioUnidad": 18 }
RESPUESTA ERROR	HEADER	HTTP/ 404 NOT FOUND
	BODY	{ "mensaje": "Producto con id 80 no encontrado", "estado": "error" }

Response

The screenshot shows the Postman application interface. In the top navigation bar, there are links for Home, Workspaces, API Network, and a search bar labeled "Search Postman". On the right side of the header are icons for Invite, Settings, Notifications, and Upgrade.

The left sidebar contains sections for Collections (with "CESAR FABRICIO TIRADO VERA's ..." selected), Environments, Flows, History, and a plus sign for creating new items.

The main workspace displays a collection named "My Collection" with a "Get data" request. The request details are as follows:

- Method:** GET
- URL:** localhost:8080/api/product/1
- Authorization:** Basic Auth (selected)
- Username:** admin
- Password:** (redacted)

The "Body" tab shows the response JSON:

```
1 {  
2   "productoId": 1,  
3   "nombreProducto": "Chais",  
4   "proveedorId": 1,  
5   "categoriaId": 1,  
6   "cantidadPorUnidad": "10 boxes x 20 bags",  
7   "precioUnidad": 18  
8 }
```

The status bar at the bottom indicates a successful response: 200 OK, 1.48 s, 466 B. Other options include Save Response, Preview, Visualize, and a copy icon.

At the bottom of the interface, there are buttons for Online, Find and replace, Console, Runner, Start Proxy, Cookies, Vault, Trash, and Help.

Home Workspaces API Network

Search Postman Ctrl K

Invite Upgrade

CESAR FABRICIO TIRADO VERA's ... New Import

GET Get data POST Post data PUT localhost:8080/api/proc DEL localhost:8080/product/+ No environment

Collections Environments Flows History

HTTP My Collection / Get data

GET localhost:8080/api/product/80

Params Authorization Headers (7) Body Scripts Settings Cookies

Auth Type Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about [Basic Auth](#) authorization.

Username admin

Password  

Body Cookies Headers (11) Test Results (0/1) 404 Not Found 1.46 s 404 B Save Response

{ } JSON Preview Debug with AI

```
1 {  
2   "mensaje": "Producto con id 80 no encontrado",  
3   "estado": "error"  
4 }
```

Online Find and replace Console Runner Start Proxy Cookies Vault Trash

Request

ENDPOINT	http://localhost:8080/api/product/abc
MÉTODO	GET
PARÁMETROS	Parámetro de ruta: "abc"

Response

RESPUESTA ERROR	HEADER	HTTP/ 400 BAD REQUEST
	BODY	{ "mensaje": "ID de producto inválido" }

Home Workspaces API Network

Search Postman Ctrl K

Invite Upgrade

CESAR FABRICIO TIRADO VERA's ... New Import

GET Get data POST Post data PUT localhost:8080/api/proc DEL localhost:8080/product/+ No environment

Collections + Search collections My Collection GET Get data POST Post data

HTTP My Collection / Get data

GET localhost:8080/api/product/abc

Save Share

Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Auth Type Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about [Basic Auth](#) authorization.

Username admin

Password admin123

Body Cookies Headers (10) Test Results (0/1) 400 Bad Request 80 ms 352 B Save Response

{ } JSON Preview Debug with AI

```
1 {  
2   "mensaje": "ID de producto inválido"  
3 }
```

Console Runner Start Proxy Cookies Vault Trash

This screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections' (selected), 'Environments', 'Flows', and 'History'. The main area displays a collection named 'My Collection' with a 'Get data' endpoint. The endpoint URL is set to 'localhost:8080/api/product/abc'. The 'Authorization' tab is active, showing 'Basic Auth' configured with 'admin' for Username and 'admin123' for Password. The response status is 400 Bad Request with the message 'ID de producto inválido'. The bottom navigation bar includes 'Console', 'Runner', 'Start Proxy', 'Cookies', 'Vault', and 'Trash'.

Request

ENDPOINT	http://localhost:8080/api/product
MÉTODO	POST
PARÁMETROS	{ "productoid": 100, "nombreProducto": "Nuevo Producto", "proveedordid": 1, "categoriald": 1, "cantidadPorUnidad": "10 cajas", "precioUnidad": 25.50 }

Response

RESPUESTA OK	HEADER	HTTP/ 201 CREATED
	CONTENT-TYPE	application/json
	BODY	{ "producto": { "productoid": 100, "nombreProducto": "Nuevo Producto", "proveedordid": 1, "categoriald": 1, "cantidadPorUnidad": "10 cajas", "precioUnidad": 25.50 }, "mensaje": "Producto creado exitosamente" }

Home Workspaces API Network

Search Postman Ctrl K

Invite Settings Notifications Upgrade

CESAR FABRICIO TIRADO VERA's ... New Import

GET Get data POST Post data PUT localhost:8080/api/proc DEL localhost:8080/product/ +

No environment

Collections Environments Flows History

My Collection / Post data

POST localhost:8080/api/product

Params Authorization Headers (9) Body Scripts Settings

Body (raw JSON) none form-data x-www-form-urlencoded

1 {
2 "productoId": 100,
3 "nombreProducto": "Nuevo Producto",
4 "proveedorId": 1,
5 "categoriaId": 1,
6 "cantidadPorUnidad": "10 cajas",
7 "precioUnidad": 25.50
8 }
9
10

201 Created 1.34 s 529 B Save Response

Body Cookies Headers (11) Test Results (1/1)

{ } JSON Preview Visualize

1 {
2 "producto": {
3 "productoId": 100,
4 "nombreProducto": "Nuevo Producto",
5 "proveedorId": 1,
6 "categoriaId": 1,
7 "cantidadPorUnidad": "10 cajas",
8 "precioUnidad": 25.50
9 },
10 "mensaje": "Producto creado exitosamente"
11 }

Online Find and replace Console

Runner Start Proxy Cookies Vault Trash

Request

ENDPOINT	http://localhost:8080/api/product
MÉTODO	POST
PARÁMETROS	{ "nombreProducto": "Nuevo Producto", "proveedorId": 1, "categoriaId": 1, "cantidadPorUnidad": "10 cajas", "precioUnidad": 25.50 }

Response

RESPUESTA ERROR	HEADER	HTTP/ 400 BAD REQUEST
	BODY	{ "mensaje": "Error al crear el producto: Identifier of entity 'org.example.servidorapi_20222297_laboratorio9.entity.Producto' must be manually assigned before calling 'persist()'" }

Home Workspaces API Network

Search Postman Ctrl K

Invite Upgrade

CESAR FABRICIO TIRADO VERA's ... New Import

GET Get data • POST Post data • PUT localhost:8080/api/proc • DEL localhost:8080/product/ • + No environment

Collections

+ Search collections

My Collection

GET Get data POST Post data

HTTP My Collection / Post data

POST localhost:8080/api/product

Params Authorization Headers (9) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Schema Beautify

```
1 {  
2   "nombreProducto": "Nuevo Producto",  
3   "proveedorId": 1,  
4   "categoriaId": 1,  
5   "cantidadPorUnidad": "10 cajas",  
6   "precioUnidad": 25.50  
7 }  
8  
9  
10  
11
```

Body Cookies Headers (10) Test Results (0/1)

400 Bad Request 913 ms 493 B Save Response

{ } JSON Preview Debug with AI

```
1 {  
2   "mensaje": "Error al crear el producto: Identifier of entity 'org.example.servidorapi_20222297_laboratorio9.entity.Producto' must be  
3   manually assigned before calling 'persist()'"  
4 }
```

Online Find and replace Console

Runner Start Proxy Cookies Vault Trash

Request

ENDPOINT	http://localhost:8080/api/product
MÉTODO	PUT
PARÁMETROS	{ "productOld": 1, "precioUnidad": 22.50, "cantidadPorUnidad": "15 cajas" }

Response

RESPUESTA OK	HEADER	HTTP/ 200 OK
	CONTENT-TYPE	application/json
	BODY	{ "resultado": "ok" }

Home Workspaces API Network

Search Postman Ctrl K

Invite Upgrade

CESAR FABRICIO TIRADO VERA's ... New Import

GET Get data • POST Post data • PUT localhost:8080/api/proc • DEL localhost:8080/product/ • + No environment

Collections

+ Search collections

My Collection

GET Get data POST Post data

localhost:8080/api/product

PUT localhost:8080/api/product

Params Authorization Headers (9) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "productoId": 1,  
3   "precioUnidad": 22.50,  
4   "cantidadPorUnidad": "15 cajas"  
5 }
```

Cookies Beautify

Body Cookies Headers (11) Test Results

200 OK 2.27 s 352 B

{ } JSON Preview Visualize

```
1 {  
2   "resultado": "ok"  
3 }
```

Online Find and replace Console

Runner Start Proxy Cookies Vault Trash

Request

ENDPOINT	http://localhost:8080/api/product
MÉTODO	PUT
PARÁMETROS	{ "precioUnidad": 22.50, "cantidadPorUnidad": "15 cajas" }

Response

RESPUESTA ERROR	HEADER	HTTP/ 400 BAD REQUEST
	BODY	{ "mensaje": "Error al crear el producto: Identifier of entity 'org.example.servidorapi_20222297_laboratorio9.entity.Producto' must be manually assigned before calling 'persist()'" }

Home Workspaces API Network

Search Postman Ctrl K

Invite Upgrade

CESAR FABRICIO TIRADO VERA's ... New Import

GET Get data • POST Post data • PUT localhost:8080/api/proc • DEL localhost:8080/product/ • + No environment

Collections

+ Search collections

My Collection

GET Get data POST Post data

localhost:8080/api/product

PUT localhost:8080/api/product

Params Authorization Headers (9) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {  
2   "precioUnidad": 22.50,  
3   "cantidadPorUnidad": "15 cajas"  
4 }  
5  
6  
7  
8  
9
```

Body Cookies Headers (10) Test Results

400 Bad Request 74 ms 378 B

{ } JSON Preview Debug with AI

```
1 {  
2   "resultado": "error",  
3   "mensaje": "Debe enviar un producto con ID"  
4 }
```

Online Find and replace Console

Runner Start Proxy Cookies Vault Trash

Request

ENDPOINT	http://localhost:8080/api/product
MÉTODO	PUT
PARÁMETROS	{ "productId": 80, "precioUnidad": 22.50, "cantidadPorUnidad": "15 cajas" }

Response

RESPUESTA ERROR	HEADER	HTTP/ 400 BAD REQUEST
	BODY	{ "resultado": "error", "mensaje": "El ID del producto enviado no existe" }

Home Workspaces API Network

Search Postman Ctrl K

Invite Upgrade

CESAR FABRICIO TIRADO VERA's ... New Import

GET Get data • POST Post data • PUT localhost:8080/api/proc • DEL localhost:8080/product/ • + No environment

Collections

+ Search collections

My Collection

GET Get data

POST Post data

localhost:8080/api/product

PUT localhost:8080/api/product

Params Authorization Headers (9) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {  
2   "productoId": 80,  
3   "precioUnidad": 22.50,  
4   "cantidadPorUnidad": "15 cajas"  
5 }
```

Send

Body Cookies Headers (10) Test Results

400 Bad Request 1.45 s 384 B

{ } JSON Preview Debug with AI

```
1 {  
2   "resultado": "error",  
3   "mensaje": "El ID del producto enviado no existe"  
4 }
```

Online Find and replace Console

Runner Start Proxy Cookies Vault Trash

Request

ENDPOINT	http://localhost:8080/api/product/74
MÉTODO	DELETE
PARÁMETROS	-

Response

RESPUESTA OK	HEADER	HTTP/ 200 OK
	CONTENT-TYPE	application/json
	BODY	{ "orderDetailsEliminados": 7, "productold": 74, "mensaje": "Producto eliminado exitosamente", "estado": "success" }

Home Workspaces API Network

Search Postman Ctrl K

Invite Upgrade

CESAR FABRICIO TIRADO VERA's ... New Import

GET Get data • POST Post data • PUT localhost:8080/api/prod • DEL localhost:8080/api/prod • + No environment

Save Share

Collections

+ Search collections

My Collection

GET Get data POST Post data

localhost:8080/api/product/74

DELETE localhost:8080/api/product/74

Params Authorization Headers (7) Body Scripts Settings Cookies

Auth Type Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about [Basic Auth](#) authorization.

Username admin

Password  

Body Cookies Headers (11) Test Results

200 OK 1.91 s 441 B

{ } JSON Preview Visualize

```
1 {  
2   "orderDetailsEliminados": 7,  
3   "productoId": 74,  
4   "mensaje": "Producto eliminado exitosamente",  
5   "estado": "success"  
6 }
```

Online Find and replace Console

Runner Start Proxy Cookies Vault Trash

Request

ENDPOINT	http://localhost:8080/api/product/101
MÉTODO	DELETE
PARÁMETROS	-

Response

RESPUESTA ERROR	HEADER	HTTP/ 400 BAD REQUEST
	BODY	{ "estado": "error", "mensaje": "Producto con id 101 no encontrado" }

Home Workspaces API Network

Search Postman Ctrl K

Invite Upgrade

CESAR FABRICIO TIRADO VERA's ... New Import

GET Get data • POST Post data • PUT localhost:8080/api/prod • DEL localhost:8080/api/prod • + No environment

Save Share

Collections

+ Search collections

My Collection

GET Get data POST Post data

localhost:8080/api/product/101

DELETE localhost:8080/api/product/101

Params Authorization Headers (7) Body Scripts Settings Cookies

Auth Type: Basic Auth

The authorization header will be automatically generated when you send the request. Learn more about [Basic Auth](#) authorization.

Username: admin

Password:  

Body Cookies Headers (11) Test Results

404 Not Found 1.06 s 405 B

{ } JSON Preview Debug with AI

```
1 {  
2   "estado": "error",  
3   "mensaje": "Producto con id 101 no encontrado"  
4 }
```

Online Find and replace Console Runner Start Proxy Cookies Vault Trash