

## Muestra de ejecución

### Caso 1: Consulta de registros sin información en la tabla (SIN ID)

GET /products get

Parameters

Name	Description
id	id
string (query)	id - id

Execute Clear

Responses Response content type \*/\*

Curl

```
curl -X GET "http://localhost:8080/products" -H "accept: */*"
```

Request URL

http://localhost:8080/products

Server response

Code	Details
200	Response body

Response body

```
[]
```

Download

### Caso 2: Consulta de registros sin información en la tabla (CON ID)

GET /products get

Parameters

Name	Description
id	id
string (query)	SDASDSA

Execute Clear

Responses Response content type \*/\*

Curl

```
curl -X GET "http://localhost:8080/products?id=SDASDSA" -H "accept: */*"
```

Request URL

http://localhost:8080/products?id=SDASDSA

Server response

Code	Details
200	Response body

Response body

```
[]
```

Download

### Caso 3: Intento de actualizar un registro no existente.

PUT /products update

Parameters

Cancel

Name	Description
<b>productDto</b> <span>required</span> (body)	<b>productDto</b> Example Value Model <pre>{   "description": "actualizado",   "id": "myZddfgfd",   "model": "abcd",   "price": 10 }</pre>

Cancel

Parameter content type  
application/json

Execute

Clear

Responses

Response content type \*/\*

Curl

```
curl -X PUT "http://localhost:8080/products" -H "accept: */*" -H "Content-Type: application/json" -d '{"description": "actualizado", "id": "myZddfgfd", "model": "abcd", "price": 10}'
```

Request URI

http://localhost:8080/products

Server response

Code

422  
Undocumented

Details

Error:  
Response body  

```
{
  "description": "Informacion invalida",
  "errores": [
    "El producto que esta intentando actualizar no existe."
  ]
}
```

Download

#### Caso 4: Intento de borrar un registro no existente.

DELETE

/products/{id} delete

Parameters

Cancel

Name	Description
id <small>required</small>	id
string (path)	njsdf

ExecuteClear

Responses

Response content type \*/\*

Curl

curl -X DELETE "http://localhost:8080/products/njsdf" -H "accept: \*/\*"

Request URL

http://localhost:8080/products/njsdf

Server response

Code	Details
422 <i>Undocumented</i>	<div>Error:</div> <div>Response body</div> <div><pre>{   "descripcion": "Informacion invalida",   "statusCode": 422,   "message": "El producto que esta intentando eliminar no existe." }</pre></div> <div>Download</div>

## Caso 5: Intento de guardar registros rebasando los limites

POST

/products save

Parameters

Cancel

Name	Description
productDto <span>required</span>	productDto
(body)	Example Value Model

```
{  "descripcion": "Mi descripcion del producto",  "id": "MiIdDelProducto",  "model": "ModeloDelProducto",  "name": "NombreDelProductoDemasiadoLargo",  "price": 20.5}
```

Cancel

Parameter content type  
application/json

Execute

Clear

Responses

Response content type \*/\*

Curl

```
curl -X POST "http://localhost:8080/products" -H "accept: */*" -H "Content-Type: application/json" -d '{"descripcion": "Mi descripcion del producto", "id": "MiIdDelProducto", "model": "ModeloDelProducto", "name": "NombreDelProductoDemasiadoLargo", "price": 20.5}'
```

Request URL

http://localhost:8080/products

Server response

Code

422

Undocumented

Details

Error.

Response body

```
{  "descripcion": "Informacion invalida.",  "errores": [    "El modelo solo soporta 10 caracteres o menos",    "El id solo soporta 10 caracteres o menos",    "El nombre solo soporta 20 caracteres o menos"  ]}
```

Download

Caso 6: se da de alta un registro:

The screenshot displays a REST client interface with the following sections:

- Execute** (blue button) and **Clear** (button).
- Responses** (tab) and **Response** (tab).
- Curl** section showing the command:

```
curl -X POST "http://localhost:8080/products" -H "accept: */*" -H "Content-Type: application/json" -d '{"description": "Descripcion", "id": "ABCD", "name": "Nombre", "price": 20.5}'
```
- Request URL** section showing `http://localhost:8080/products`.
- Server response** section with a table:

Code	Details
201	<div><b>Response body</b><pre>{  "id": "ABCD",  "name": "Nombre",  "description": "Descripcion",  "price": 20.5,  "model": "Model"}</pre></div> <div><b>Response headers</b></div>

## Caso 7: Búsqueda del registro previamente dado de alta.

Parameters

Name	Description
id	id
string (query)	ABCDFGH123

Execute

Responses

Curl

```
curl -X GET "http://localhost:8080/products?id=ABCDFGH123" -H "accept: */*"
```

Request URL

```
http://localhost:8080/products?id=ABCDFGH123
```

Server response

Code	Details
200	<div>Response body</div> <pre>{   "id": "ABCDFGH123",   "name": "Nombre",   "description": "Descripcion",   "price": 20.5,   "model": "Model" }</pre> <div>Response headers</div>

## Caso 8: Actualizar el registro y solo permite actualizar los permitidos

The screenshot shows a REST client interface with the following components:

- Example Value / Model:** A JSON object representing a product update:

```
{  "description": "actualizado",  "id": "ABCDGHI23",  "name": "act",  "price": "999",  "model": "999",  "BC156": 10}
```
- Parameter content type:** A dropdown menu set to `application/json`.
- Buttons:** `Cancel`, `Execute`, and `Clear`.
- Responses:** A section showing the `Curl` command and the `Server response`.
  - Curl:**

```
curl -X PUT https://localhost:8080/products -H "accept: */*" -H "Content-Type: application/json" -d "{ \"description\": \"actualizado\", \"id\": \"ABCDGHI23\", \"model\": \"act\", \"name\": \"\", \"price\": 10}"
```
  - Request URL:** `https://localhost:8080/products`
  - Server response:** A `200` status code and a JSON response body:

```
{  "id": "ABCDGHI23",  "name": "Nombre",  "description": "actualizado",  "price": 20.5,  "model": "act"}
```

## Caso 9: Consulta de multiples registros sin enviar ID

Name

Description

id

id

string

id - id

(query)

Execute

Clear

Responses

Response content type \*/\*

Curl

curl -X GET "http://localhost:8080/products" -H "accept: \*/\*"

Request URL

http://localhost:8080/products

Server response

Code

Details

200

Response body

[{"id": "ABCDGHI23", "name": "Nombre", "description": "actualizado", "price": 20.5, "model": "Act"}, {"id": "otro123", "name": "names", "description": "otra descr", "price": 17.5, "model": "Modeldo"}]

Download

Response headers

connection: keep-alive  
content-type: application/json  
date: Fri, 30 Apr 2021 03:20:57 GMT  
keep-alive: timeout=60  
transfer-encoding: chunked

Responses

Code

Description

200

OK

Caso 10: Borrado lógico de un registro



DELETE

/products/{id} delete

Parameters

Cancel

Name	Description
<div>id</div> <div>required</div> <div>string</div> <div>(path)</div>	<div>id</div> <div>ABCDFGH123</div>

Execute

Clear

Responses

Response content type \*/\*

Curl

curl -X DELETE "http://localhost:8080/products/ABCDFGH123" -H "accept: \*/\*"

Request URL

http://localhost:8080/products/ABCDFGH123

Server response

Code

Details

200	<div>Response headers</div> <div>           connection: keep-alive            content-length: 0            date: Fri, 30 Apr 2021 03:22:05 GMT            keep-alive: timeout=60         </div>
-----	---

Responses

Code	Description
200	OK

Example Value

Model

{}

## Resultados en BD

Data Output		Explain	Messages	Notifications		
	enabled boolean	id [PK] character varying (10)	name character varying (20)	description character varying (200)	model character varying (10)	price double precision
1	false	ABCDFGH123	Nombre	actualizado	act	20.5
2	true	otro123	names	otra descr	Modeldo	17.5