

Somos Compu-Express, una empresa destinada a la venta de equipos tecnológicos de diversas utilidades. A manera de facilitar el acceso a realizar compras, se implementó una página web donde se brindan los elementos necesarios para navegar por nuestras distintas secciones tales como visualizar todos los productos o seleccionar productos según su categoría.

## Documentación de Pruebas realizadas con Postman

## Entidad: Categoría

Create

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists collections, APIs, environments, mock servers, and monitors. A 'Well done! You got the basics.' message with a trophy icon is displayed. The main workspace shows a POST request to 'http://localhost:9000/api/v1/categorias/'. The request body is set to JSON and contains the following data:

```

1 {
2   "id": 4,
3   "codigoCategoria": 1002,
4   "nombreCategoria": "Video",
5   "productos": null
6 }

```

The response status is 200 OK, time 15 ms, size 327 B. Below the request, there are tabs for Body, Cookies, Headers (8), and Test Results. The Body tab shows the JSON response:

```

1 {
2   "id": 4,
3   "codigoCategoria": 1002,
4   "nombreCategoria": "Video",
5   "productos": null
6 }

```

## Update

The screenshot shows the Postman application interface. The 'My Workspace' sidebar is identical to the previous screenshot. The main workspace shows a PUT request to 'http://localhost:9000/api/v1/categorias/4'. The request body is set to JSON and contains the following data:

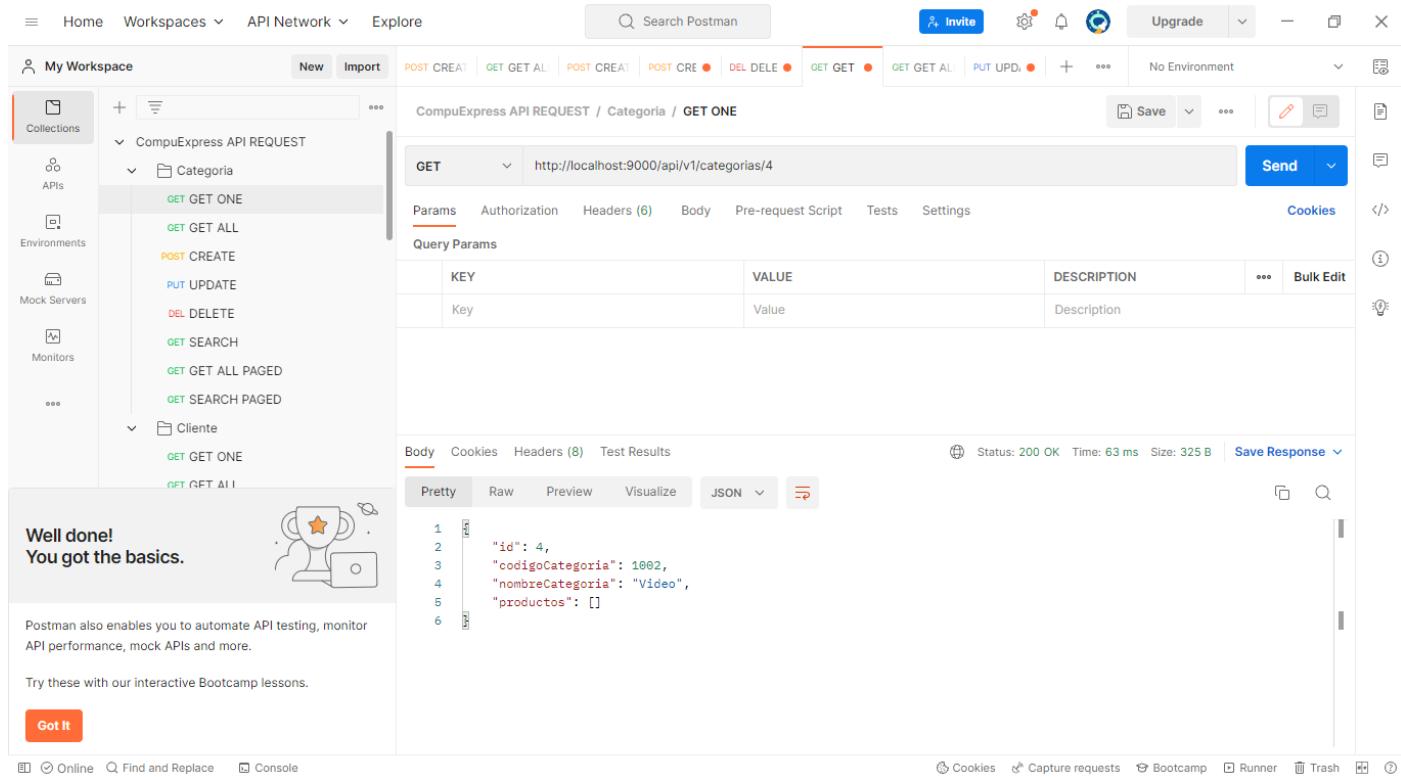
```

1 {
2   "id": 4,
3   "codigoCategoria": 1001,
4   "nombreCategoria": "Perifericos"
5 }

```

The response status is 200 OK, time 54 ms, size 333 B. The interface is otherwise similar to the first screenshot, with tabs for Body, Cookies, Headers (8), and Test Results.

## GetOne



The screenshot shows the Postman interface with the following details:

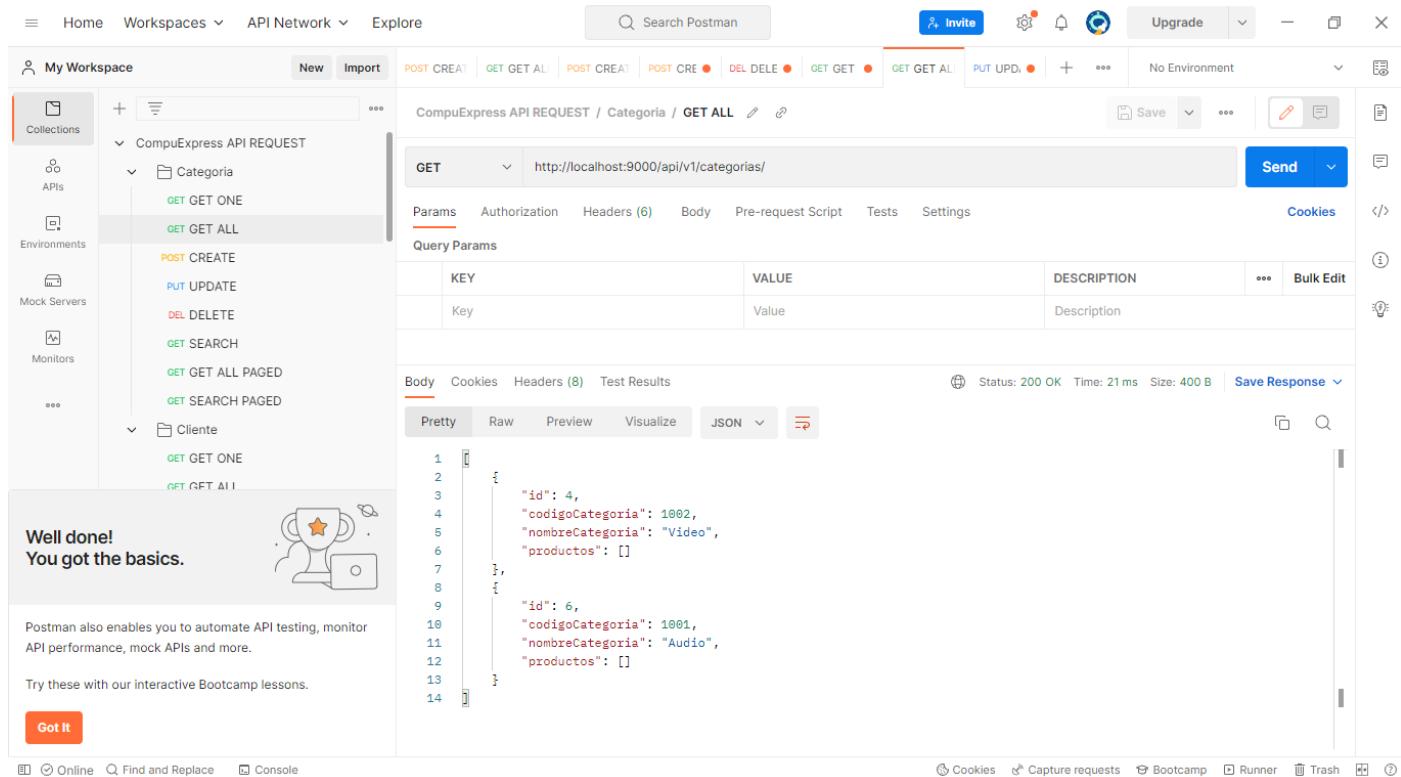
- Request URL:** http://localhost:9000/api/v1/categorias/4
- Method:** GET
- Body:** JSON response (Pretty view)

```

1 {
2   "id": 4,
3   "codigoCategoria": 1002,
4   "nombreCategoria": "Video",
5   "productos": []
6 }
    
```

The response status is 200 OK with a time of 63 ms and a size of 325 B.

## GetAll



The screenshot shows the Postman interface with the following details:

- Request URL:** http://localhost:9000/api/v1/categorias/
- Method:** GET
- Body:** JSON response (Pretty view)

```

1 [
2   {
3     "id": 4,
4     "codigoCategoria": 1002,
5     "nombreCategoria": "Video",
6     "productos": []
7   },
8   {
9     "id": 6,
10    "codigoCategoria": 1001,
11    "nombreCategoria": "Audio",
12    "productos": []
13  }
14 ]
    
```

The response status is 200 OK with a time of 21 ms and a size of 400 B.

## Delete

The screenshot shows the Postman interface with the following details:

- Request Type:** DELETE
- URL:** http://localhost:9000/api/v1/categorias/6
- Headers:** (6)
- Body:** (Empty)
- Response Status:** 204 No Content
- Response Time:** 24 ms
- Response Size:** 233 B

The left sidebar shows a collection named "CompuExpress API REQUEST" containing categories like "Categoría" and "Cliente". A "DELETE" operation under "Categoría" is selected.

A message box on the left says: "Well done! You got the basics." with an icon of a trophy.

At the bottom, there's a "Got It" button and navigation links for Online, Find and Replace, Console, Cookies, Capture requests, Bootcamp, Runner, and Trash.

## Search

The screenshot shows the Postman interface with the following details:

- Request Type:** GET
- URL:** http://localhost:9000/api/v1/categorias/search?filtro=per
- Params:** filtro: per
- Headers:** (6)
- Body:** (Empty)
- Response Status:** 200 OK
- Response Time:** 74 ms
- Response Size:** 333 B

The left sidebar shows a collection named "CompuExpress API REQUEST" containing categories like "Categoría" and "Cliente". A "GET" operation under "Categoría" is selected.

A message box on the left says: "Well done! You got the basics." with an icon of a trophy.

At the bottom, there's a "Got It" button and navigation links for Online, Find and Replace, Console, Cookies, Capture requests, Bootcamp, Runner, and Trash.

## Get All Paged

The screenshot shows the Postman interface with the following details:

- Collection:** CompuExpress API REQUEST / Categoria
- Request Type:** GET
- URL:** http://localhost:9000/api/v1/categorias/paged?page=0&size=5&sort=id,asc
- Params:**
  - page: 0
  - size: 5
  - sort: id,asc
- Body:** JSON response (Pretty) showing the first page of categorized products.

Response status: 200 OK, Time: 111 ms, Size: 647 B

Postman also enables you to automate API testing, monitor API performance, mock APIs and more. Try these with our interactive Bootcamp lessons.

Well done! You got the basics.

The screenshot shows the Postman interface with the following details:

- Collection:** CompuExpress API REQUEST / Categoria
- Request Type:** GET
- URL:** http://localhost:9000/api/v1/categorias/paged?page=0&size=5&sort=id,asc
- Params:**
  - page: 0
  - size: 5
  - sort: id,asc
- Body:** JSON response (Pretty) showing the first page of categorized products.

Response status: 200 OK, Time: 111 ms, Size: 647 B

Postman also enables you to automate API testing, monitor API performance, mock APIs and more. Try these with our interactive Bootcamp lessons.

Well done!  
You got the basics.

## Search Paged

The screenshot shows the Postman interface with the following details:

- Collection:** CompuExpress API REQUEST / Categoria
- Request Type:** GET
- URL:** http://localhost:9000/api/v1/categorias/searchPaged?filtro=per&page=0&size=5&sort=id,desc
- Params:**

KEY	VALUE	DESCRIPTION
filtro	per	
page	0	
size	5	
sort	id,desc	
- Body:** JSON response (Pretty) showing the search results.
- Status:** 200 OK

On the left sidebar, under "Collections", the "Categoría" collection is expanded, showing various endpoints like GET ONE, GET ALL, POST CREATE, etc. A "Well done! You got the basics." message with a trophy icon is displayed.

The screenshot shows the Postman interface with the following details:

- Collection:** CompuExpress API REQUEST / Categoria
- Request Type:** GET
- URL:** http://localhost:9000/api/v1/categorias/searchPaged?filtro=per&page=0&size=5&sort=id,desc
- Params:**

KEY	VALUE	DESCRIPTION
filtro	per	
page	0	
size	5	
sort	id,desc	
- Body:** JSON response (Pretty) showing a more detailed search results object.
- Status:** 200 OK

On the left sidebar, under "Collections", the "Categoría" collection is expanded, showing various endpoints like GET ONE, GET ALL, POST CREATE, etc. A "Well done! You got the basics." message with a trophy icon is displayed.

## Entidad: Cliente

### *Create*

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists collections: 'CompuExpress API REQUEST' (containing 'Categoria', 'Cliente', 'Marca', and 'ProductoDescripción') and 'Mock Servers'. The 'Cliente' collection is expanded, showing methods: 'GET GET ONE', 'GET GET ALL', 'POST CREATE', 'PUT UPDATE', 'DEL DELETE', 'GET SEARCH', 'GET GET ALL PAGED', 'GET SEARCH PAGED'. The 'POST CREATE' method is selected. The main panel displays a POST request to 'http://localhost:9000/api/v1/clientes/'. The 'Body' tab is active, showing JSON input:

```

1
2   ...
3   ...
4   ...
5   ...
6   ...
7   ...
8
  
```

The response status is 200 OK, with a response body:

```

1
2
3
4
5
6
7
8
  
```

The 'Body' tab also includes Pretty, Raw, Preview, Visualize, and JSON dropdowns.

This screenshot shows the same Postman interface as the previous one, but with different JSON input in the 'Body' tab:

```

1
2   ...
3   ...
4   ...
5   ...
6   ...
7   ...
8
  
```

The response status is 200 OK, with a response body:

```

1
2
3
4
5
6
7
8
  
```

The 'Body' tab includes Pretty, Raw, Preview, Visualize, and JSON dropdowns.

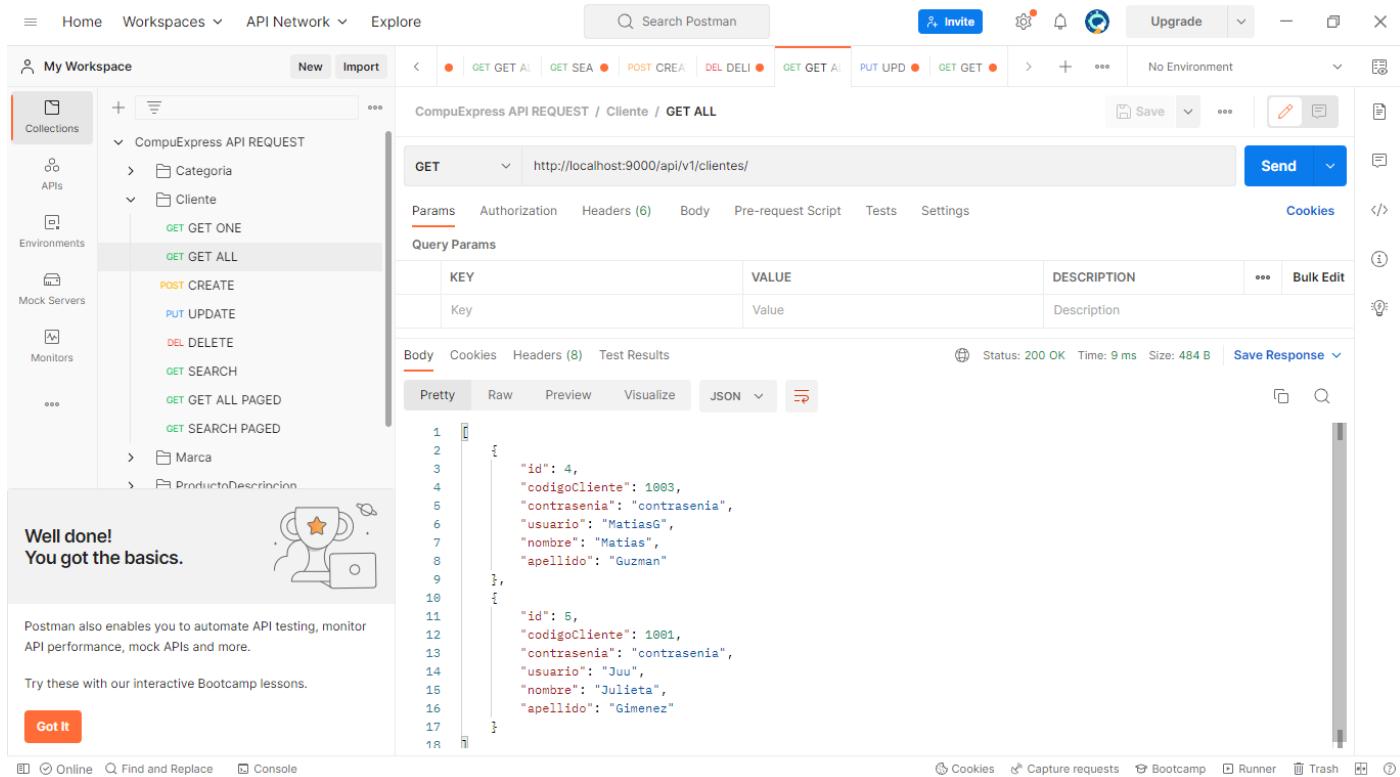
## *Update*

## *GetOne*

The screenshot shows the Postman application interface. On the left, the sidebar includes sections for Collections, APIs, Environments, Mock Servers, Monitors, and more. The main workspace displays a request for 'GET ONE' of a client. The request URL is `http://localhost:9000/api/v1/clientes/5`. The 'Params' tab is selected, showing a single query parameter 'Key' with value 'Value'. Below the request details, the response body is shown as JSON:

```
1  {
2     "id": 5,
3     "codigoCliente": 1001,
4     "contrasenia": "contrasenia",
5     "usuario": "Juu",
6     "nombre": "Julietta",
7     "apellido": "Gimenez"
8 }
```

## GetAll



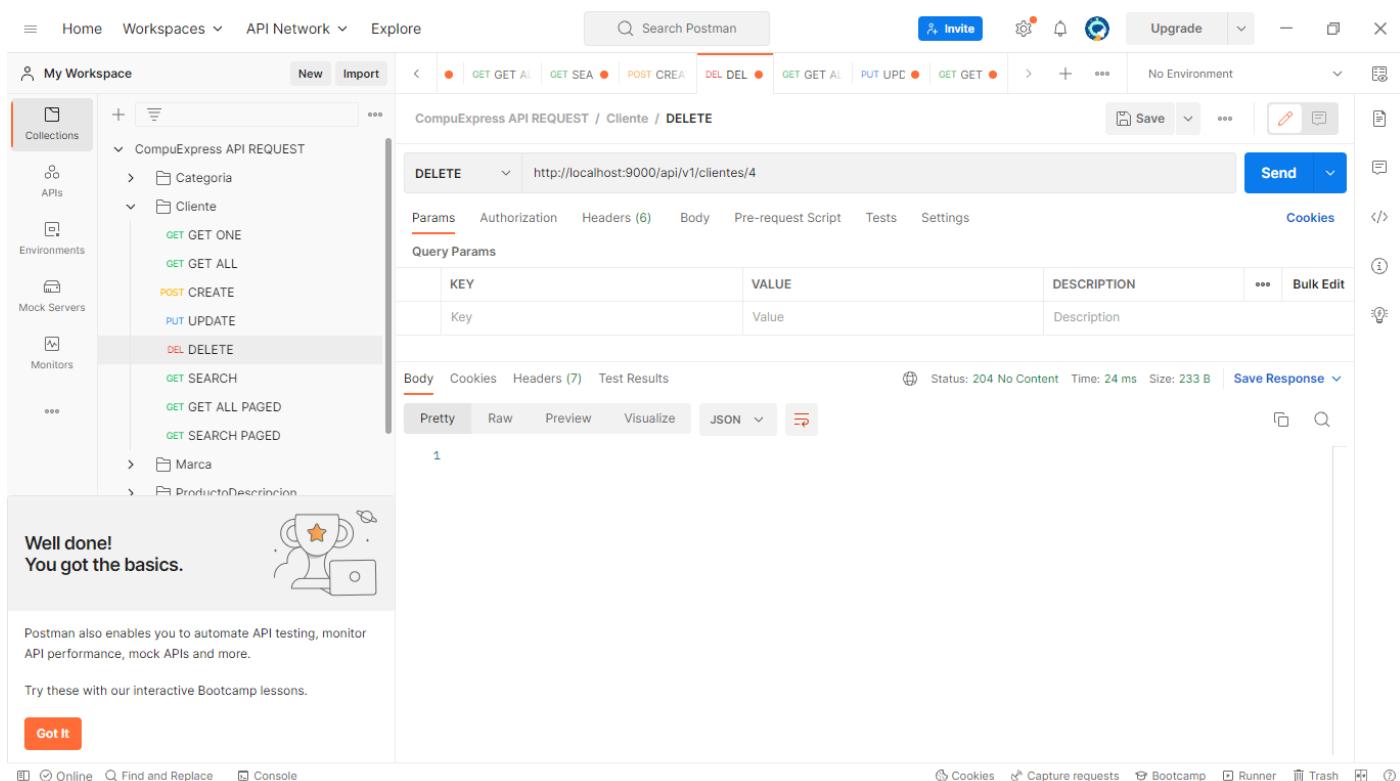
The screenshot shows the Postman interface with a successful API call. The request method is GET, directed at <http://localhost:9000/api/v1/clientes/>. The response status is 200 OK, with a time of 9 ms and a size of 484 B. The response body is a JSON array containing two client records:

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
{
  "id": 4,
  "codigoCliente": 1003,
  "contrasenia": "contrasenia",
  "usuario": "MatiasG",
  "nombre": "Matias",
  "apellido": "Guzman"
},
{
  "id": 5,
  "codigoCliente": 1001,
  "contrasenia": "contrasenia",
  "usuario": "Juu",
  "nombre": "Julietta",
  "apellido": "Gimenez"
}

```

## Delete



The screenshot shows the Postman interface with a successful API call. The request method is DELETE, directed at <http://localhost:9000/api/v1/clientes/4>. The response status is 204 No Content, with a time of 24 ms and a size of 233 B.

The screenshot shows the Postman interface with the following details:

- Collection:** CompuExpress API REQUEST / Cliente
- Request Type:** GET
- URL:** http://localhost:9000/api/v1/clientes/
- Params:** Authorization, Headers (6), Body, Pre-request Script, Tests, Settings
- Query Params:** Key, Value, Description
- Body:** Pretty, Raw, Preview, Visualize, JSON
- Response:** Status: 200 OK, Time: 16 ms, Size: 368 B
- JSON Response:**

```

1
2   {
3     "id": 5,
4     "codigoCliente": 1001,
5     "contrasenia": "contrasenia",
6     "usuario": "Juu",
7     "nombre": "Julietta",
8     "apellido": "Gimenez"
9
10

```

**Well done!**  
You got the basics.

Postman also enables you to automate API testing, monitor API performance, mock APIs and more.

Try these with our interactive Bootcamp lessons.

**Got It**

## Search

The screenshot shows the Postman interface with the following details:

- Collection:** CompuExpress API REQUEST / Cliente
- Request Type:** GET
- URL:** http://localhost:9000/api/v1/clientes/search?filtro=Juu
- Params:** Authorization, Headers (6), Body, Pre-request Script, Tests, Settings
- Query Params:** filtro, Juu
- Body:** Pretty, Raw, Preview, Visualize, JSON
- Response:** Status: 200 OK, Time: 19 ms, Size: 368 B
- JSON Response:**

```

1
2   {
3     "id": 5,
4     "codigoCliente": 1001,
5     "contrasenia": "contrasenia",
6     "usuario": "Juu",
7     "nombre": "Julietta",
8     "apellido": "Gimenez"
9
10

```

**Well done!**  
You got the basics.

Postman also enables you to automate API testing, monitor API performance, mock APIs and more.

Try these with our interactive Bootcamp lessons.

**Got It**

## Get All Paged

The screenshot shows the Postman interface with the following details:

- Collection:** CompuExpress API REQUEST / Cliente
- Request Type:** GET
- URL:** http://localhost:9000/api/v1/clientes/paged?page=0&size=5&sort=id,asc
- Params:**
  - page: 0
  - size: 5
  - sort: id,asc
- Body:** JSON response (Pretty Print) showing the results of the paged query.
- Status:** 200 OK

```

1
2   "content": [
3     {
4       "id": 5,
5       "codigoCliente": 1001,
6       "contrasenia": "contrasenia",
7       "usuaxio": "Juu",
8       "nombre": "Julietta",
9       "apellido": "Gimenez"
10    },
11    "pageable": {
12      "sort": {
13        "empty": false,
14        "unsorted": false,
15        "sorted": true
16      }
17    },
18    "offset": 0,
19    "pageSize": 5,
20    "pageNumber": 0,
21    "paged": true,
22    "unpaged": false
23  },
24  "totalPages": 1,
25  "totalElements": 1,
26  "last": true,
27  "size": 5,
28  "number": 0,
29  "sort": {
30    "empty": false,
31    "unsorted": false,
32    "sorted": true
33  },
34  "numberOfElements": 1,
35  "first": true,
36  "empty": false
37

```

**Well done!**  
You got the basics.

The screenshot shows the Postman interface with the following details:

- Collection:** CompuExpress API REQUEST / Cliente
- Request Type:** GET
- URL:** http://localhost:9000/api/v1/clientes/paged?page=0&size=5&sort=id,asc
- Params:**
  - page: 0
  - size: 5
  - sort: id,asc
- Body:** JSON response (Pretty Print) showing the results of the paged query.
- Status:** 200 OK

```

14
15   "empty": false,
16   "unsorted": false,
17   "sorted": true
18 },
19   "offset": 0,
20   "pageSize": 5,
21   "pageNumber": 0,
22   "paged": true,
23   "unpaged": false
24 },
25   "totalPages": 1,
26   "totalElements": 1,
27   "last": true,
28   "size": 5,
29   "number": 0,
30   "sort": {
31     "empty": false,
32     "unsorted": false,
33     "sorted": true
34   },
35   "numberOfElements": 1,
36   "first": true,
37   "empty": false
38

```

**Well done!**  
You got the basics.

## Search Paged

The screenshot shows the Postman interface with the following details:

- Request Method:** GET
- URL:** <http://localhost:9000/api/v1/clientes/searchPaged?filtro=Juu&page=0&size=5&sort=id,desc>
- Query Params:**

KEY	VALUE
filtro	Juu
page	0
size	5
sort	id,desc
- Body:** (Pretty) JSON response showing a single client record.

```

1
2   "content": [
3     {
4       "id": 5,
5       "codigoCliente": 1001,
6       "contrasenia": "contrasenia",
7       "usuario": "Juu",
8       "nombre": "Julietta",
9       "apellido": "Gimenez"
10    },
11  ],

```

**Postman UI Elements:**

- Left sidebar: Collections, APIs, Environments, Mock Servers, Monitors.
- Middle section: My Workspace, New, Import, CompuExpress API REQUEST, Cliente, GET SEARCH PAGED.
- Bottom section: Well done! You got the basics., Trophy icon, Got It button.
- Bottom navigation: Online, Find and Replace, Console, Cookies, Capture requests, Bootcamp, Runner, Trash.

The screenshot shows the Postman interface with the following details:

- Request Method:** GET
- URL:** <http://localhost:9000/api/v1/clientes/searchPaged?filtro=Juu&page=0&size=5&sort=id,desc>
- Query Params:**

KEY	VALUE
filtro	Juu
page	0
size	5
sort	id,desc
- Body:** (Pretty) JSON response showing a pageable object.

```

12  "pageable": {
13    "sort": {
14      "empty": false,
15      "unsorted": false,
16      "sorted": true
17    },
18    "offset": 0,
19    "pageSize": 5,
20    "pageNumber": 0,
21    "paged": true,
22    "unpaged": false
23  },
24  "totalPages": 1,
25  "totalElements": 1,
26  "last": true,
27  "size": 5,
28  "number": 0,
29  "sort": {
30    "empty": false,
31    "unsorted": false,
32    "sorted": true
33  },
34  "numberOfElements": 1,
35  "first": true

```

**Postman UI Elements:**

- Left sidebar: Collections, APIs, Environments, Mock Servers, Monitors.
- Middle section: My Workspace, New, Import, CompuExpress API REQUEST, Cliente, GET SEARCH PAGED.
- Bottom section: Well done! You got the basics., Trophy icon, Got It button.
- Bottom navigation: Online, Find and Replace, Console, Cookies, Capture requests, Bootcamp, Runner, Trash.

## Entidad: Marca

### Create

The screenshot shows the Postman application interface. On the left, the sidebar lists 'My Workspace' with a collection named 'Marca' containing various endpoints like 'GET ONE', 'GET ALL', and 'POST CREATE'. The main workspace shows a POST request to 'http://localhost:9000/api/v1/marcas/'. The 'Body' tab is selected, displaying a JSON payload:

```
1 {"codigoMarca": 1001,
2     "nombreMarca": "Lenovo"}  
3  
4
```

Below the request, the response is shown in the 'Pretty' tab:

```
1 {  
2     "id": 2,  
3     "codigoMarca": 1001,  
4     "nombreMarca": "Lenovo"  
5 }
```

The status bar at the bottom indicates: Status: 200 OK Time: 38 ms Size: 303 B.

This screenshot shows the same Postman interface after another creation request. The 'Marca' collection now has an additional entry: 'POST CREATE' with the payload:

```
1 {"codigoMarca": 1002,
2     "nombreMarca": "HP"}  
3  
4
```

The response in the 'Pretty' tab is:

```
1 {  
2     "id": 3,  
3     "codigoMarca": 1002,  
4     "nombreMarca": "HP"  
5 }
```

The status bar at the bottom indicates: Status: 200 OK Time: 58 ms Size: 299 B.

## Update

The screenshot shows the Postman interface with the following details:

- Header Bar:** Home, Workspaces, API Network, Explore, Search Postman, Invite, Upgrade, etc.
- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors, etc. Under Collections, there is a folder named "Marca" containing "GET GET ONE", "GET GET ALL", "POST CREATE", "PUT UPDATE", "DEL DELETE", "GET SEARCH", "GET GET ALL PAGED", and "GET SEARCH PAGED".
- Request Panel:** Method: PUT, URL: http://localhost:9000/api/v1/marcas/1. Headers tab is selected. Body tab is selected, showing JSON input:

```
1
2   ...
3     ...
4       ...
5         ...
```

Body content:

```
1
2   "id": 2,
3   "codigoMarca": 1001,
4   "nombreMarca": "Dell"
```
- Response Panel:** Status: 200 OK, Time: 30 ms, Size: 301 B. Body tab is selected, showing the response content:

```
1
2   "id": 2,
3   "codigoMarca": 1001,
4   "nombreMarca": "Dell"
```
- Bottom Navigation:** Online, Find and Replace, Console, Cookies, Capture requests, Bootcamp, Runner, Trash.

## GetOne

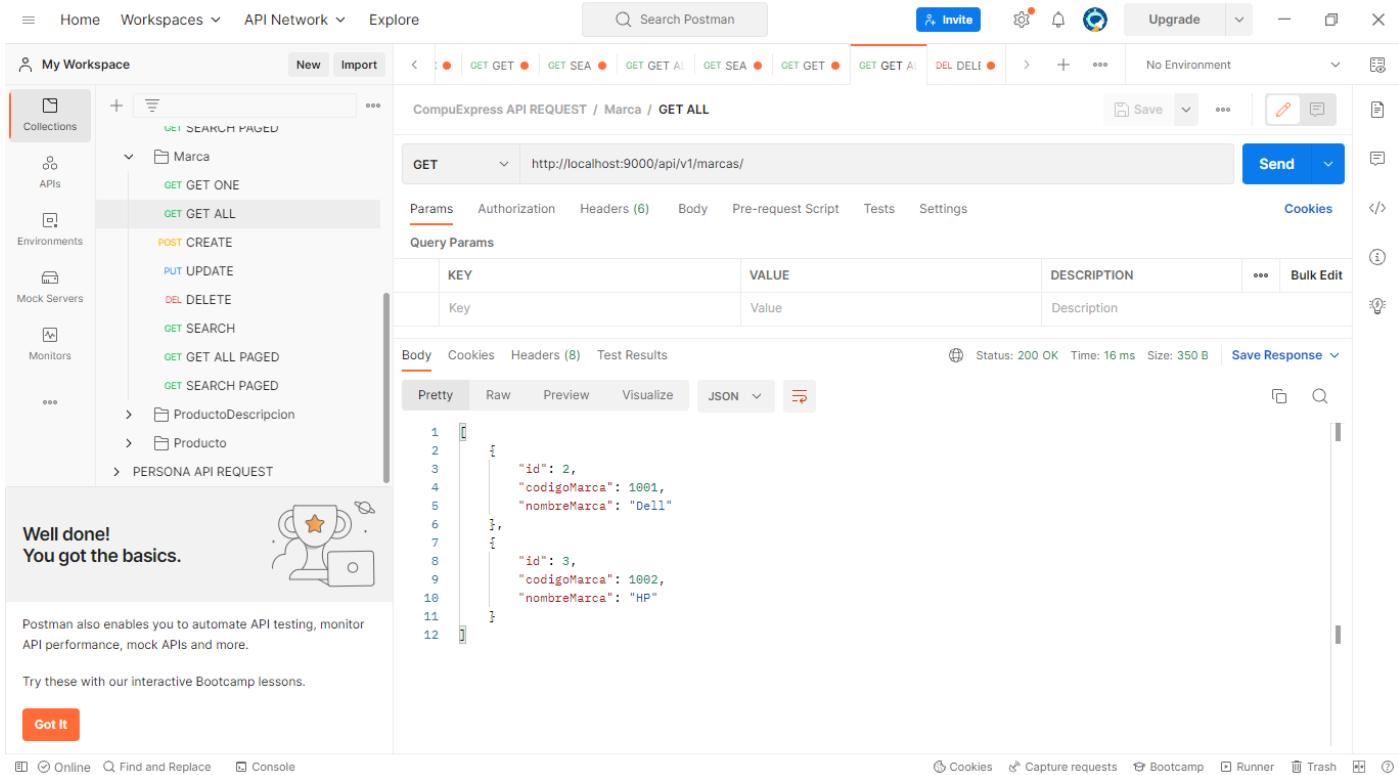
The screenshot shows the Postman interface with the following details:

- Header Bar:** Home, Workspaces, API Network, Explore, Search Postman, Invite, Upgrade, etc.
- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors, etc. Under Collections, there is a folder named "Marca" containing "GET GET ONE", "GET GET ALL", "POST CREATE", "PUT UPDATE", "DEL DELETE", "GET SEARCH", "GET GET ALL PAGED", and "GET SEARCH PAGED".
- Request Panel:** Method: GET, URL: http://localhost:9000/api/v1/marcas/2. Headers tab is selected. Query Params tab is selected, showing a table:

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description	...	
- Response Panel:** Status: 200 OK, Time: 23 ms, Size: 301 B. Body tab is selected, showing the response content:

```
1
2   "id": 2,
3   "codigoMarca": 1001,
4   "nombreMarca": "Dell"
```
- Bottom Navigation:** Online, Find and Replace, Console, Cookies, Capture requests, Bootcamp, Runner, Trash.

## GetAll



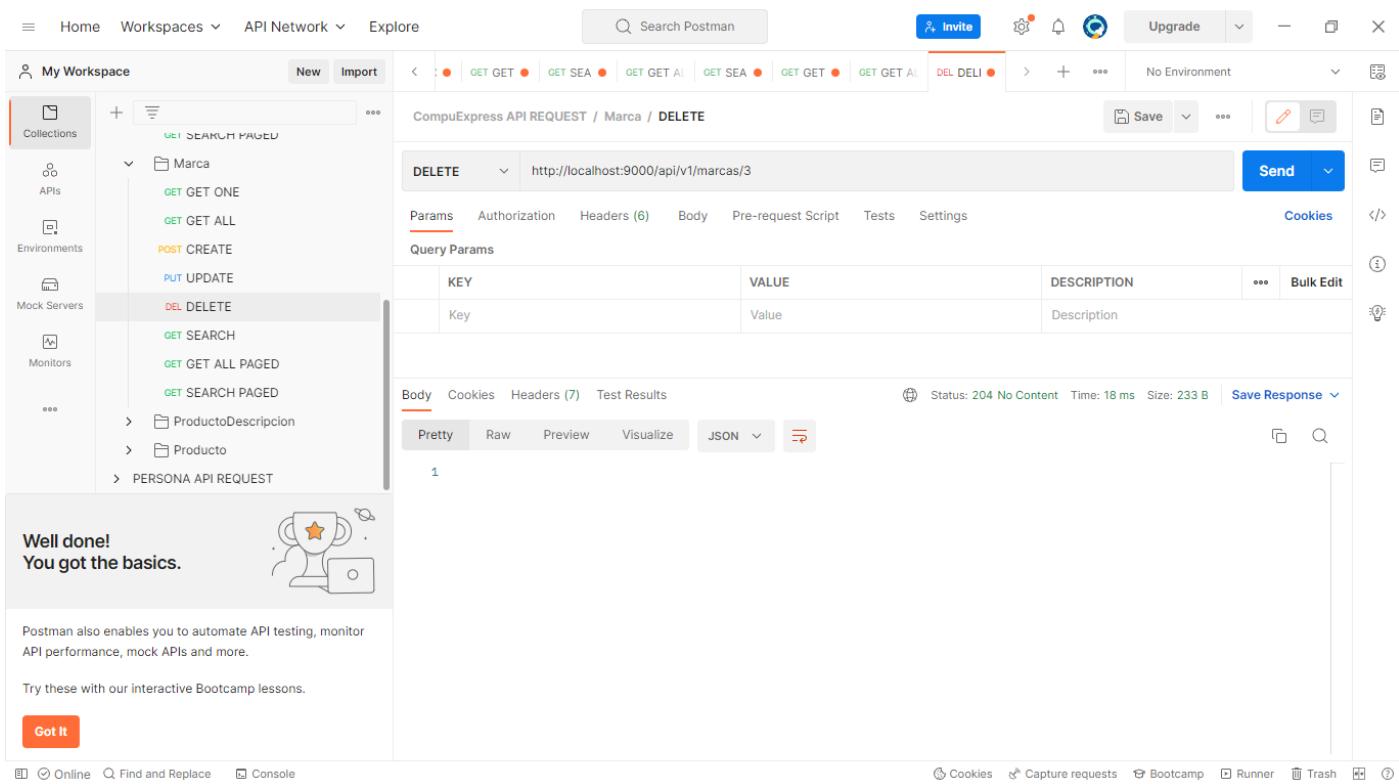
The screenshot shows the Postman interface with the following details:

- Request Type:** GET
- URL:** http://localhost:9000/api/v1/marcas/
- Headers:** (6)
- Body:** (Pretty) JSON response showing two objects representing brands:
 

```

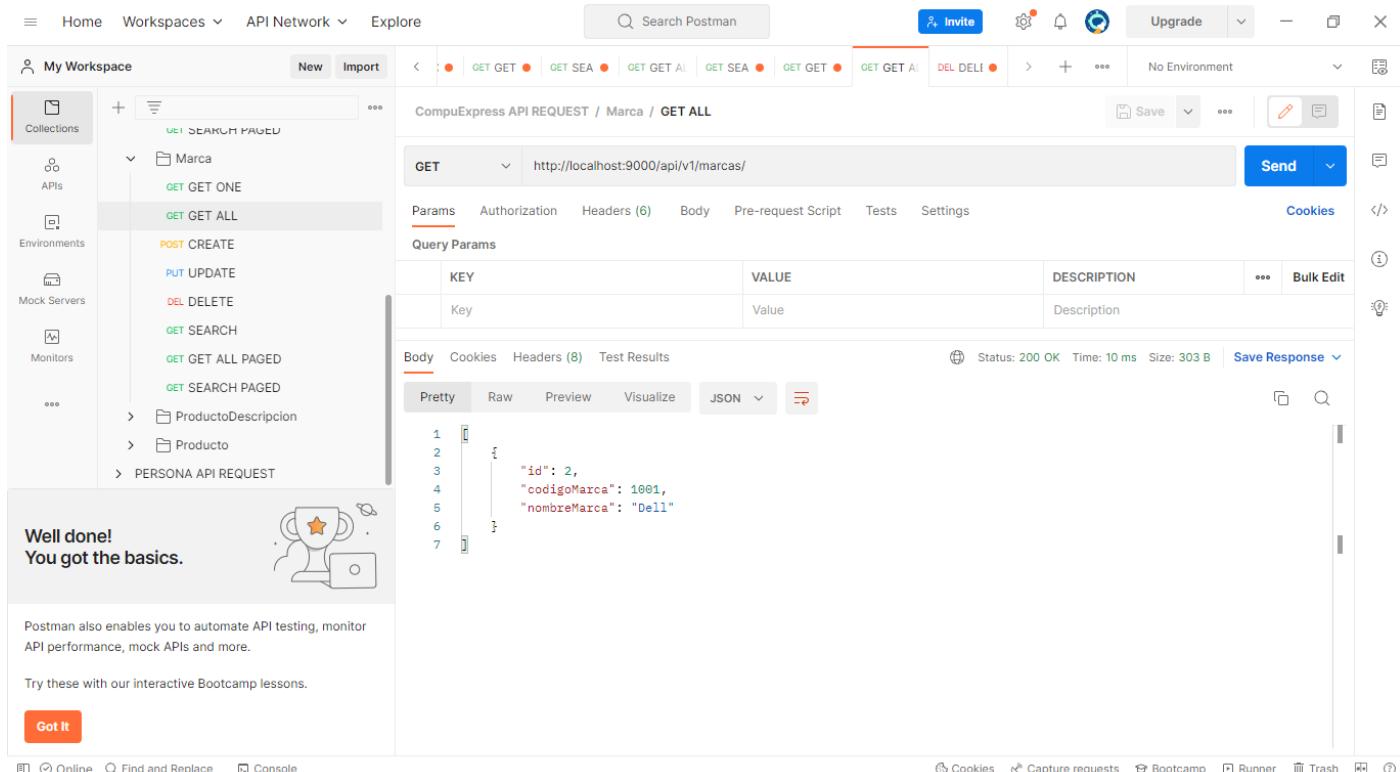
1  {
2   "id": 2,
3   "codigoMarca": 1001,
4   "nombreMarca": "Dell"
5 },
6   {
7     "id": 3,
8     "codigoMarca": 1002,
9     "nombreMarca": "HP"
10 }
11
12
      
```
- Status:** 200 OK
- Time:** 16 ms
- Size:** 350 B

## Delete



The screenshot shows the Postman interface with the following details:

- Request Type:** DELETE
- URL:** http://localhost:9000/api/v1/marcas/3
- Headers:** (6)
- Body:** (Pretty) JSON response showing a single digit '1'.
- Status:** 204 No Content
- Time:** 18 ms
- Size:** 233 B



The screenshot shows the Postman interface with the following details:

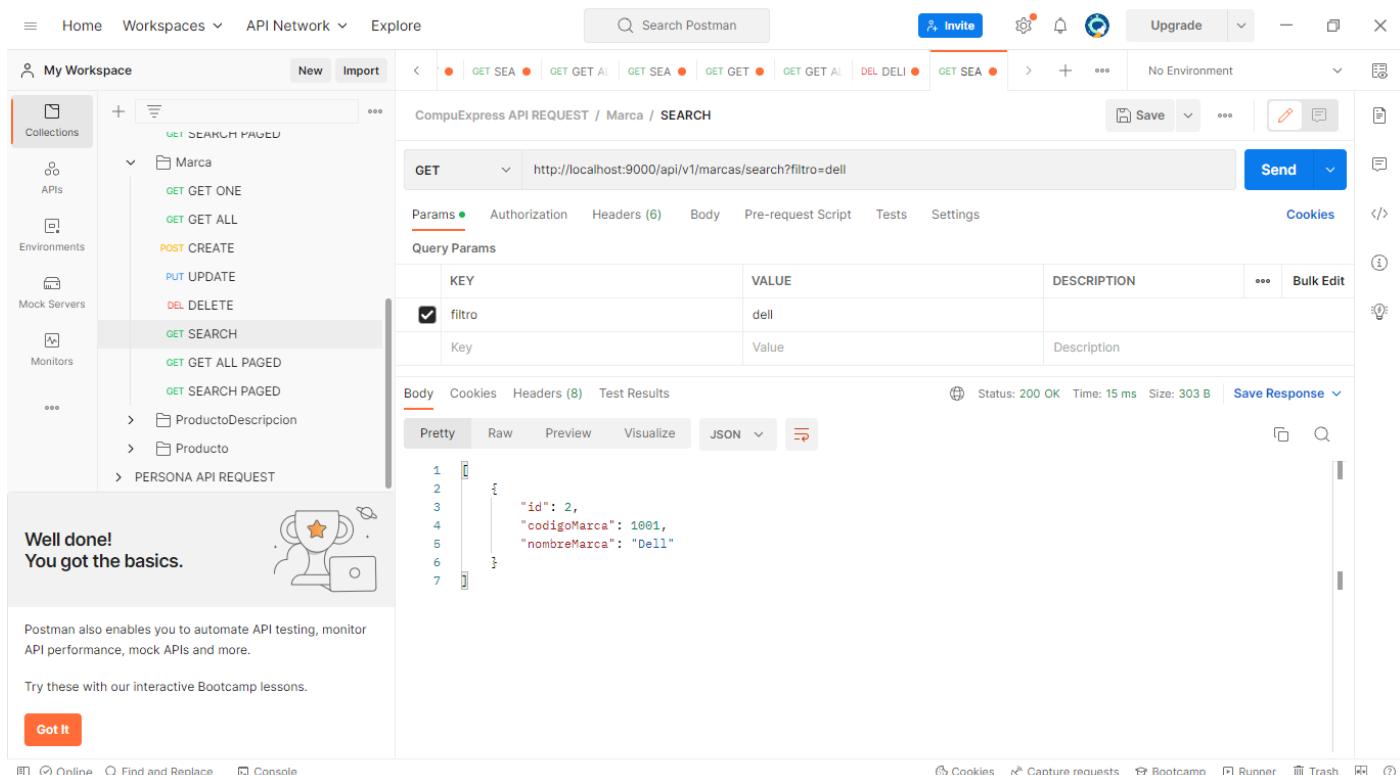
- Collection:** My Workspace
- Request:** GET /api/v1/marcas/
- Response Status:** 200 OK
- Response Body (Pretty JSON):**

```

1  {
2   "id": 2,
3   "codigoMarca": 1001,
4   "nombreMarca": "Dell"
5 }
```

The interface includes a sidebar with collections like 'Marcas' and 'PERSONA API REQUEST', and a bottom section with a 'Well done! You got the basics.' message and a 'Got It' button.

## Search



The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace
- Request:** GET /api/v1/marcas/search?filtro=dell
- Response Status:** 200 OK
- Response Body (Pretty JSON):**

```

1  {
2   "id": 2,
3   "codigoMarca": 1001,
4   "nombreMarca": "Dell"
5 }
```

The interface includes a sidebar with collections like 'Marcas' and 'PERSONA API REQUEST', and a bottom section with a 'Well done! You got the basics.' message and a 'Got It' button.

## Get All Paged

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace / Marca / GET SEARCH PAGED
- Method:** GET
- URL:** http://localhost:9000/api/v1/marcas/paged?page=0&size=5&sort=id,asc
- Params:**
  - page: 0
  - size: 5
  - sort: id,asc
- Body:** JSON response (Pretty) showing the first page of results for 'Dell'.
- Status:** 200 OK, Time: 23 ms, Size: 617 B

Postman also enables you to automate API testing, monitor API performance, mock APIs and more. Try these with our interactive Bootcamp lessons.

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace / Marca / GET SEARCH PAGED
- Method:** GET
- URL:** http://localhost:9000/api/v1/marcas/paged?page=0&size=5&sort=id,asc
- Params:**
  - page: 0
  - size: 5
  - sort: id,asc
- Body:** JSON response (Pretty) showing the first page of results for 'Dell'.
- Status:** 200 OK, Time: 23 ms, Size: 617 B

Postman also enables you to automate API testing, monitor API performance, mock APIs and more. Try these with our interactive Bootcamp lessons.

## Search Paged

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace > Marca
- Request:** GET /api/v1/marcas/searchPaged?filtro=dell&page=0&size=5&sort=id,asc
- Params:**
  - filtro: dell
  - page: 0
  - size: 5
  - sort: id,asc
- Body:** JSON response (Pretty) showing the first page of search results for 'Dell'.
- Status:** 200 OK, Time: 28 ms, Size: 617 B

Message: Well done! You got the basics.

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace > Marca
- Request:** GET /api/v1/marcas/searchPaged?filtro=dell&page=0&size=5&sort=id,asc
- Params:**
  - filtro: dell
  - page: 0
  - size: 5
  - sort: id,asc
- Body:** JSON response (Pretty) showing the first page of search results for 'Dell' with detailed pageable information.
- Status:** 200 OK, Time: 28 ms, Size: 617 B

Message: Well done! You got the basics.

## Entidad: ProductoDescripcion

Create

The screenshot shows the Postman application interface. The left sidebar is titled "My Workspace" and contains sections for Collections, APIs, Environments, Mock Servers, and Monitors. Under "Collections", there is a "CompuExpress API REQUEST / ProductoDescripcion / CREATE" collection. The "Body" tab is selected in the main request editor, which shows a POST request to "http://localhost:9000/api/v1/productodescripciones/". The body content is a JSON object with fields "id", "codigoProductoDescripcion", and "descripcionPD". The response status is 200 OK, and the response body is identical to the request body.

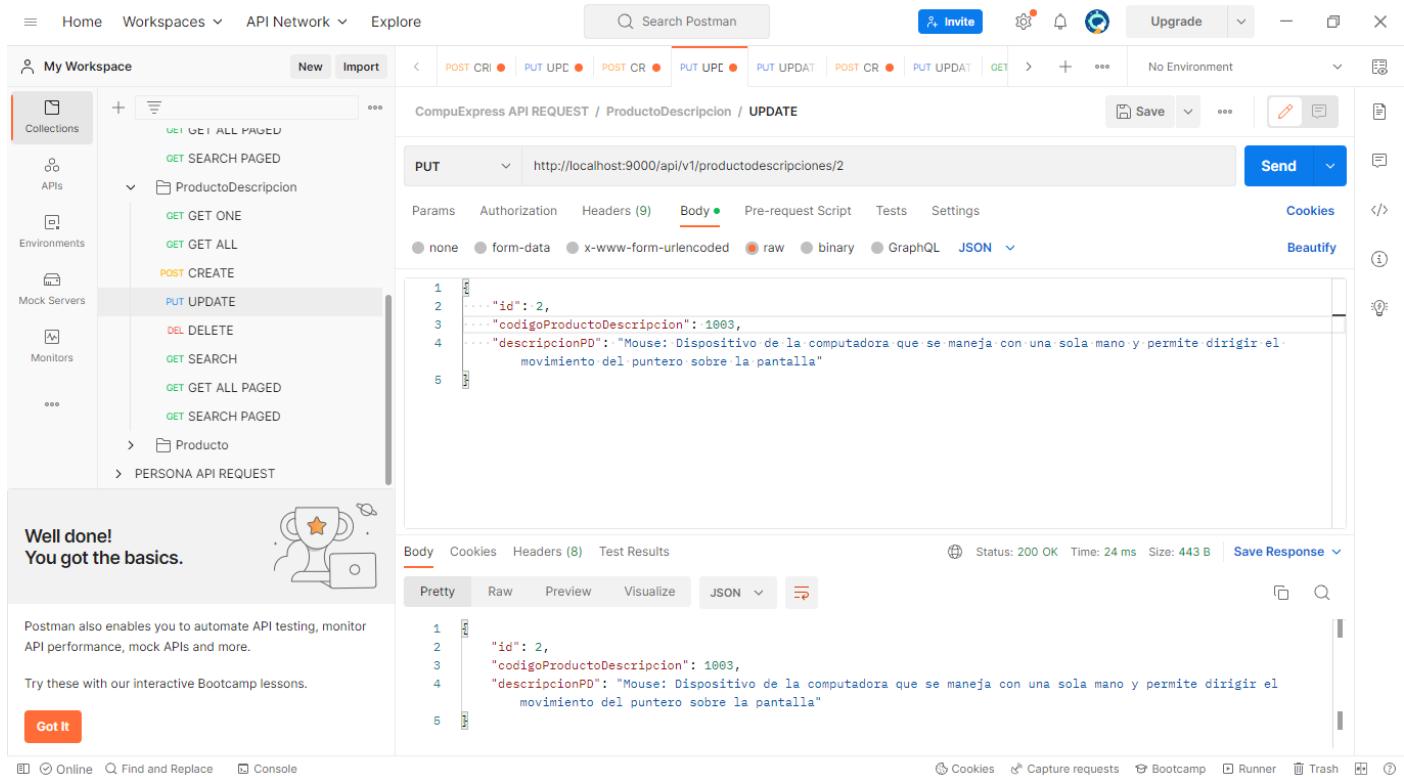
Well done!  
You got the basics.

Postman also enables you to automate API testing, monitor API performance, mock APIs and more.

Try these with our interactive Bootcamp lessons.

**Got It**

## Update



The screenshot shows the Postman interface with the following details:

- Request Type:** PUT
- URL:** http://localhost:9000/api/v1/productodescripciones/2
- Body (JSON):**

```

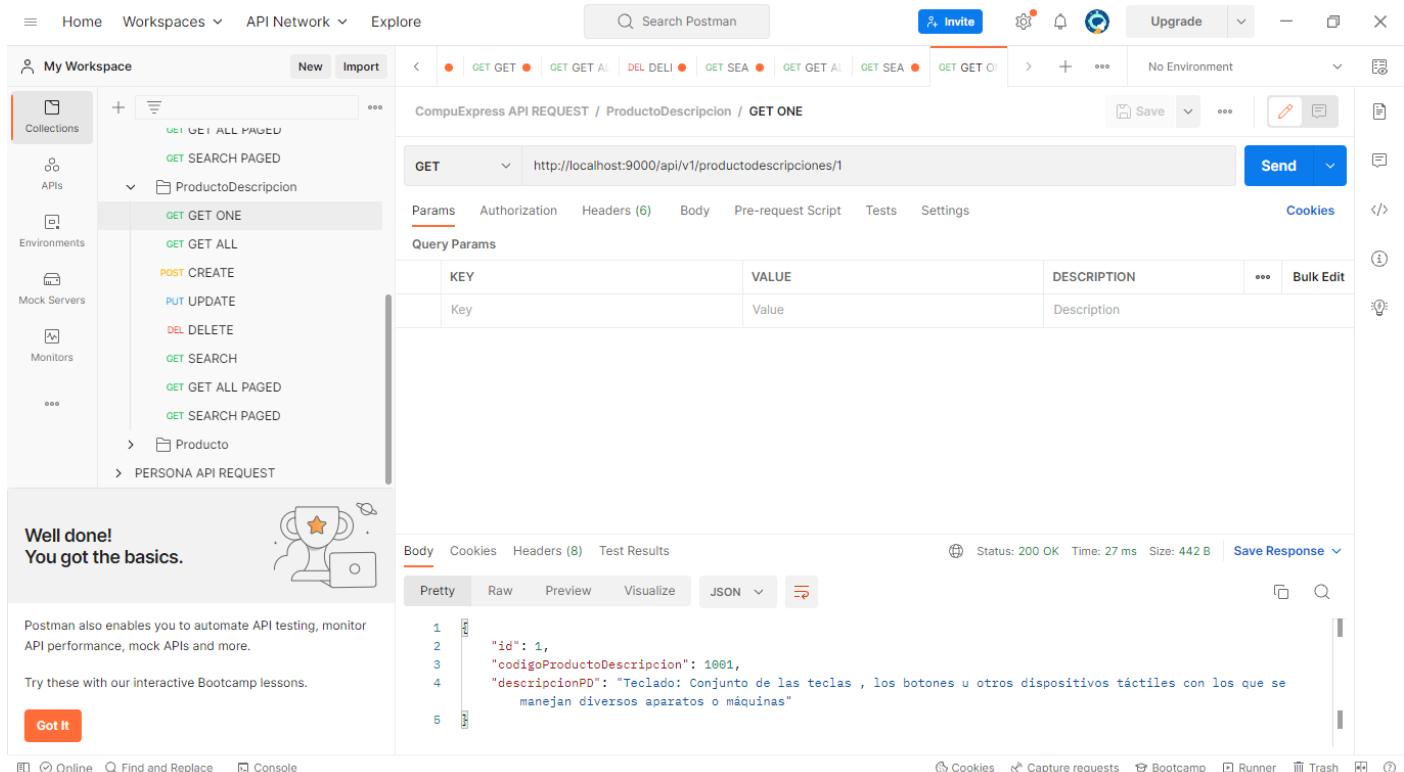
1 ...
2 ... "id": 2,
3 ... "codigoProductoDescripcion": 1003,
4 ... "descripcionPD": "Mouse: Dispositivo de la computadora que se maneja con una sola mano y permite dirigir el movimiento del puntero sobre la pantalla"
5 ...
    
```

- Response Status:** 200 OK
- Body (Pretty):**

```

1 ...
2 ...
3 ...
4 ...
5 ...
    
```

## GetOne



The screenshot shows the Postman interface with the following details:

- Request Type:** GET
- URL:** http://localhost:9000/api/v1/productodescripciones/1
- Headers (6):**
- Query Params:**

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

- Response Status:** 200 OK
- Body (Pretty):**

```

1 ...
2 ...
3 ...
4 ...
5 ...
    
```

## GetAll

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors.
- Request URL:** http://localhost:9000/api/v1/productodescripciones/
- Method:** GET
- Headers:** (6)
- Body:** JSON response (Pretty) showing two product descriptions:
 

```

1  {
2   "id": 1,
3   "codigoProductoDescripcion": 1001,
4   "descripcionPD": "Teclado: Conjunto de las teclas , los botones u otros dispositivos táctiles con los que se manejan diversos aparatos o máquinas"
5 },
6   {
7   "id": 2,
8   "codigoProductoDescripcion": 1003,
9   "descripcionPD": "Mouse: Dispositivo de la computadora que se maneja con una sola mano y permite dirigir el movimiento del puntero sobre la pantalla"
10 }
11
12 
```
- Status:** 200 OK, Time: 18 ms, Size: 635 B

## Delete

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors.
- Request URL:** http://localhost:9000/api/v1/productodescripciones/2
- Method:** DELETE
- Headers:** (6)
- Body:** JSON response (Pretty) showing a single digit '1'.
- Status:** 204 No Content, Time: 27 ms, Size: 233 B

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors.
- Request URL:** http://localhost:9000/api/v1/productodescripciones/
- Method:** GET
- Query Params:** Key: Key, Value: Value
- Body:** JSON response (Pretty):
 

```

1 {
2   "id": 1,
3   "codigoProductoDescripcion": 1001,
4   "descripcionPD": "Teclado: Conjunto de las teclas , los botones u otros dispositivos táctiles con los que se manejan diversos aparatos o máquinas"
5 }
6
7 
```
- Status:** 200 OK, Time: 12 ms, Size: 444 B

## Search

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors.
- Request URL:** http://localhost:9000/api/v1/productodescripciones/search?filtro=teclado
- Method:** GET
- Query Params:** filtro: teclado
- Body:** JSON response (Pretty):
 

```

1 {
2   "id": 1,
3   "codigoProductoDescripcion": 1001,
4   "descripcionPD": "Teclado: Conjunto de las teclas , los botones u otros dispositivos táctiles con los que se manejan diversos aparatos o máquinas"
5 }
6
7 
```
- Status:** 200 OK, Time: 9 ms, Size: 444 B

## *Get All Paged*

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors.
- Request URL:** http://localhost:9000/api/v1/productodescripciones/paged?page=0&size=5&sort=id,asc
- Method:** GET
- Params:** page (0), size (5), sort (id,asc)
- Body:** JSON response (Pretty View) showing the first few items of a pageable list of product descriptions.
- Status:** 200 OK, Time: 12 ms, Size: 758 B

**Success Message:** Well done! You got the basics.

**Description:** Postman also enables you to automate API testing, monitor API performance, mock APIs and more.

**Call To Action:** Try these with our interactive Bootcamp lessons. Got It.

The screenshot shows the Postman application interface. The left sidebar contains navigation links: Home, Workspaces, API Network, Explore, Collections, APIs, Environments, Mock Servers, Monitors, and a 'Well done! You got the basics.' message with a coffee cup icon. The main workspace displays a 'CompuExpress API REQUEST / ProductoDescripcion / GET ALL PAGED' collection. A specific GET request is selected, showing the URL `http://localhost:9000/api/v1/productodescripciones/paged?page=0&size=5&sort=id,asc`. The response status is 200 OK, time 12 ms, size 758 B. The response body is a JSON object representing pagination parameters:

```
11     "empty": false,
12     "unsorted": false,
13     "sorted": true
14 },
15     "offset": 0,
16     "pageSize": 5,
17     "pageNumber": 0,
18     "paged": true,
19     "unpaged": false
20 },
21     "totalPages": 1,
22     "totalElements": 1,
23     "last": true,
24     "size": 5,
25     "number": 0,
26     "sort": {
27         "empty": false,
28         "unsorted": false,
29         "sorted": true
30     },
31     "numberOfElements": 1,
32     "first": true,
33     "empty": false
34 }
```

At the bottom, there are buttons for 'Online', 'Find and Replace', 'Console', 'Cookies', 'Capture requests', 'Bootcamp', 'Runner', 'Trash', and 'Upgrade'.

## Search Paged

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace
- Request:** GET /api/v1/productodescripciones/searchPaged?filtro=teclado&page=0&size=5&sort=id,desc
- Params:**

KEY	VALUE	DESCRIPTION
filtro	teclado	
page	0	
size	5	
sort	id,desc	
- Body:** JSON response (Pretty view) showing the search results.
- Response Status:** 200 OK, Time: 15 ms, Size: 758 B
- Message:** "Well done! You got the basics."

The screenshot shows the Postman interface with the following details:

- Collection:** My Workspace
- Request:** GET /api/v1/productodescripciones/searchPaged?filtro=teclado&page=0&size=5&sort=id,desc
- Params:**

KEY	VALUE	DESCRIPTION
filtro	teclado	
page	0	
size	5	
sort	id,desc	
- Body:** JSON response (Pretty view) showing a more detailed search results object.
- Response Status:** 200 OK, Time: 15 ms, Size: 758 B
- Message:** "Well done! You got the basics."

## Entidad: Producto

Create

The screenshot shows the Postman application interface. The left sidebar is titled "My Workspace" and contains sections for Collections, APIs, Environments, Mock Servers, Monitors, and Persona API REQUEST. Under "Collections", there is a "Producto" folder containing "GET GET ONE", "GET GET ALL", "POST CREATE", "PUT UPDATE", "DEL DELETE", "GET SEARCH", "GET GET ALL PAGED", and "GET SEARCH PAGED". The "POST CREATE" item is currently selected. The main workspace shows a "CompuExpress API REQUEST / Producto / CREATE" collection. A POST request is being prepared to the URL `http://localhost:9000/api/v1/products/`. The "Body" tab is selected, showing a JSON payload:

```
2   "codigoProducto": 1001,
3   "nombreProducto": "Teclado",
4   "precioProducto": 10000.00,
5   "marca": [
6     {
7       "id": 2,
8       "codigoMarca": 1001,
9       "nombreMarca": "Dell"
10    }
11  ],
12  "productoDescripcion": null,
13  "tipoProducto": null
```

The "Body" tab also displays the response status: Status: 200 OK, Time: 29 ms, Size: 439 B, and a "Save Response" button. Below the body, there are tabs for Pretty, Raw, Preview, Visualize, and JSON.

The screenshot shows the Postman application interface. The left sidebar contains sections for Collections, APIs, Environments, Mock Servers, Monitors, and various API endpoints. The main workspace displays a POST request to 'http://localhost:9000/api/v1/productos/'. The 'Body' tab is selected, showing a JSON payload for creating a product. The response tab shows a successful 200 OK status with a response body identical to the request's body. A 'Well done! You got the basics.' message is displayed at the bottom.

My Workspace

New Import

POST CR ● GET GET AL POST CREA POST CR ● DEL DELI ● GET GET ● GET GET AL PUT > + ... No Environment

CompuExpress API REQUEST / Producto / CREATE

Save Send

POST http://localhost:9000/api/v1/productos/

Params Authorization Headers (9) Body Tests Settings Cookies Beautify

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

```
1
2     "codigoProducto": 1002,
3     "nombreProducto": "Mouse",
4     "precioProducto": 8000.00,
5     "marca": {
6         "id": 2,
7         "codigoMarca": 1001,
8         "nombreMarca": "Dell"
9     }
```

Body Cookies Headers (8) Test Results Status: 200 OK Time: 19 ms Size: 436 B Save Response

Pretty Raw Preview Visualize JSON

```
1
2     "id": 3,
3     "nombreProducto": "Mouse",
4     "codigoProducto": 1002,
5     "precioProducto": 8000.0,
6     "marca": [
7         {
8             "id": 2,
9             "codigoMarca": 1001,
10            "nombreMarca": "Dell"
11        },
12        "productoDescripcion": null,
```

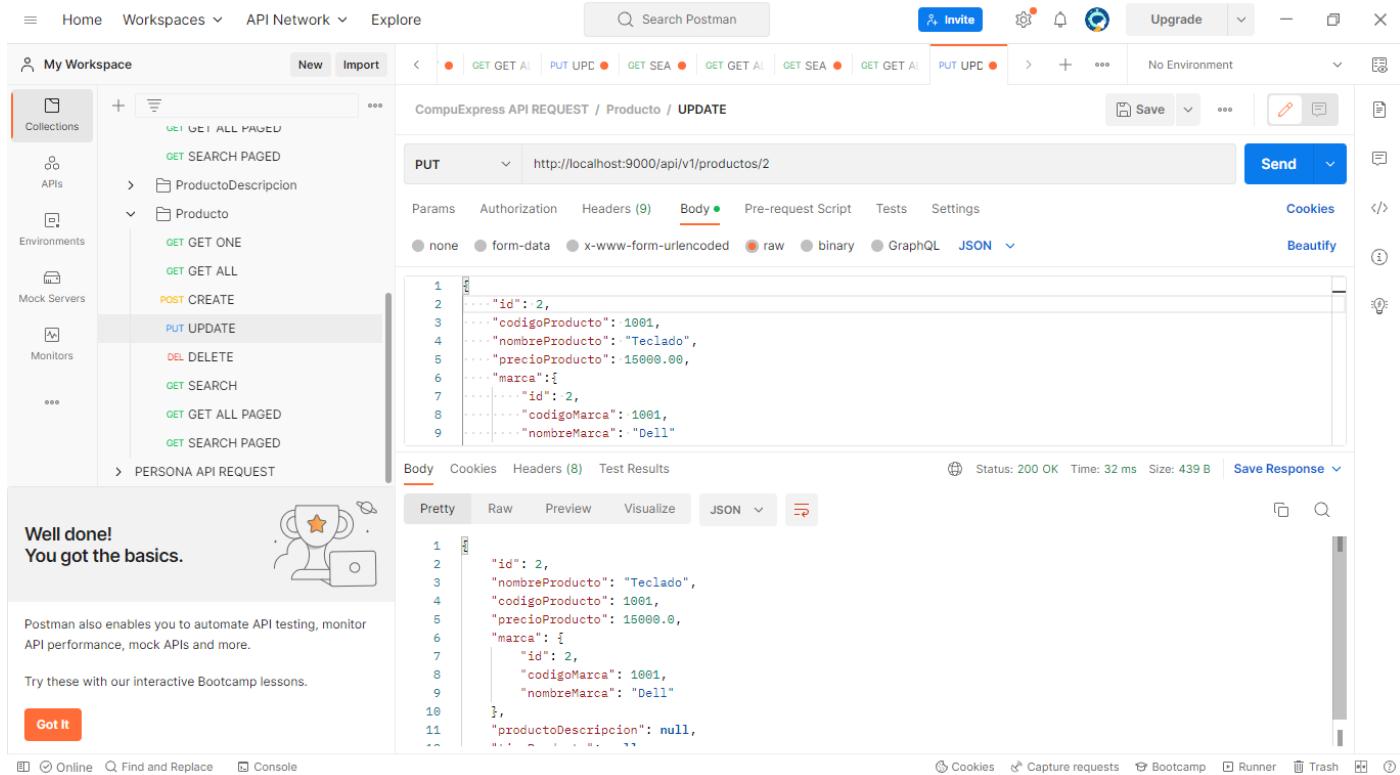
Well done!  
You got the basics.

Postman also enables you to automate API testing, monitor API performance, mock APIs and more.

Try these with our interactive Bootcamp lessons.

Got It

## Update



The screenshot shows the Postman interface with the following details:

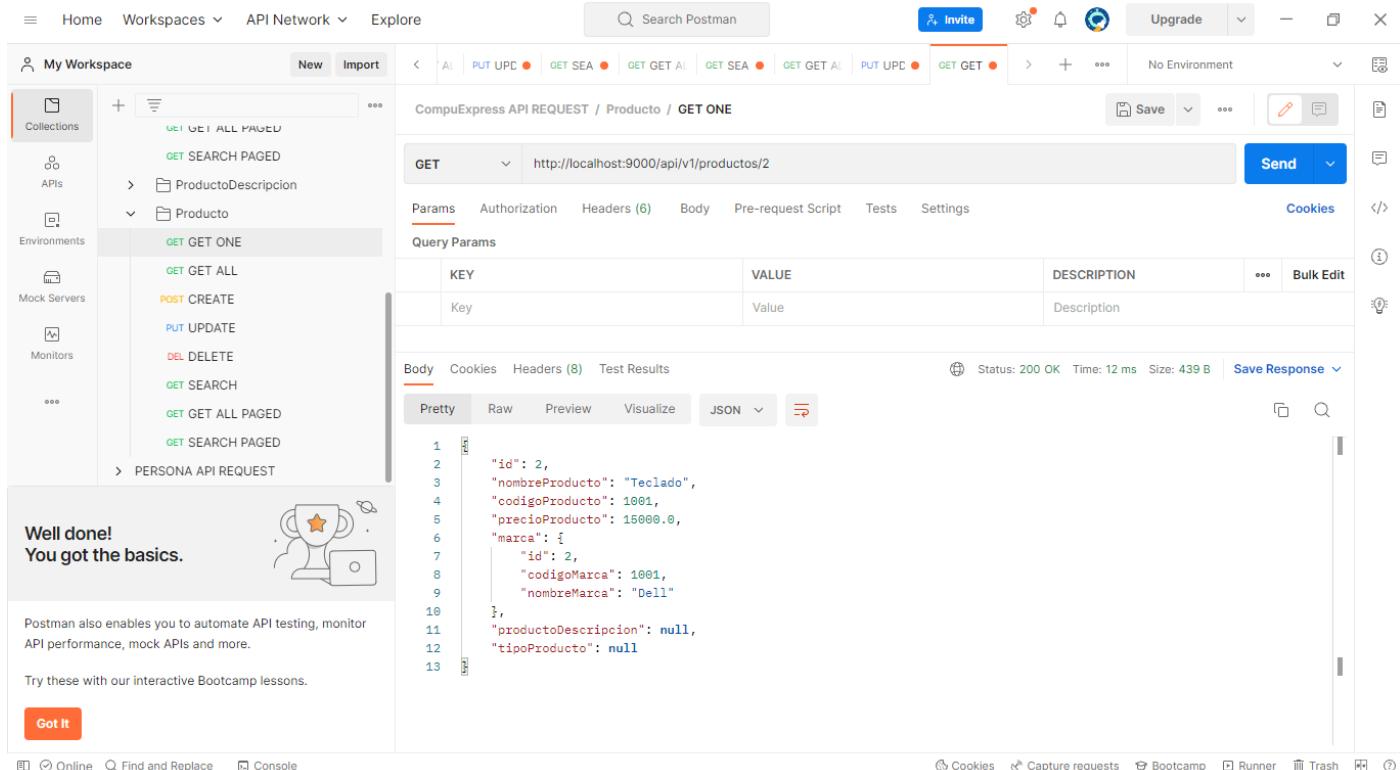
- Header:** PUT http://localhost:9000/api/v1/productos/2
- Body (JSON):**

```

1
2   "id": 2,
3   "nombreProducto": "Teclado",
4   "codigoProducto": 1001,
5   "precioProducto": 15000.00,
6   "marca": {
7     "id": 2,
8     "codigoMarca": 1001,
9     "nombreMarca": "Dell"
10 },
11   "productoDescripcion": null,
12   "tipoProducto": null
13

```

## GetOne



The screenshot shows the Postman interface with the following details:

- Header:** GET http://localhost:9000/api/v1/productos/2
- Query Params:**

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

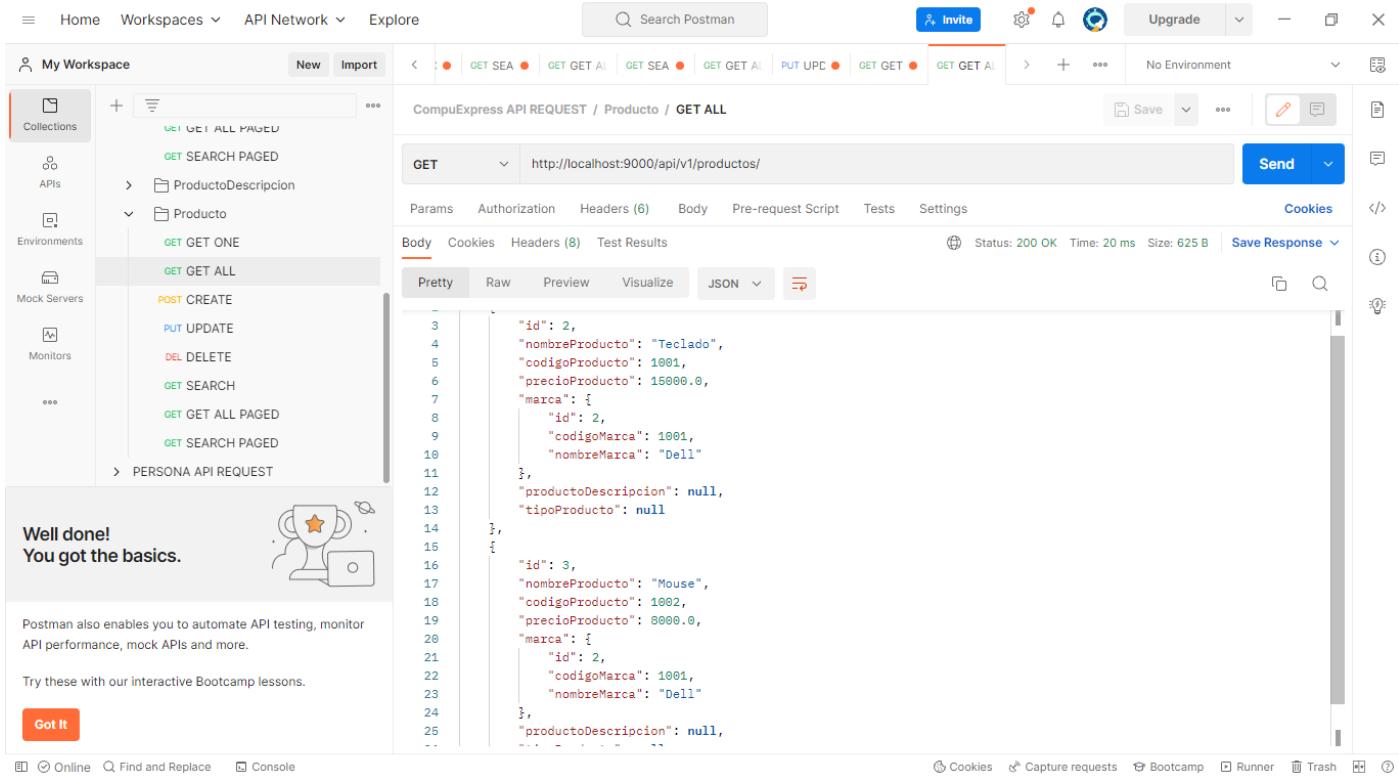
- Body (JSON):**

```

1
2   "id": 2,
3   "nombreProducto": "Teclado",
4   "codigoProducto": 1001,
5   "precioProducto": 15000.00,
6   "marca": {
7     "id": 2,
8     "codigoMarca": 1001,
9     "nombreMarca": "Dell"
10 },
11   "productoDescripcion": null,
12   "tipoProducto": null
13

```

## GetAll



The screenshot shows the Postman interface with the following details:

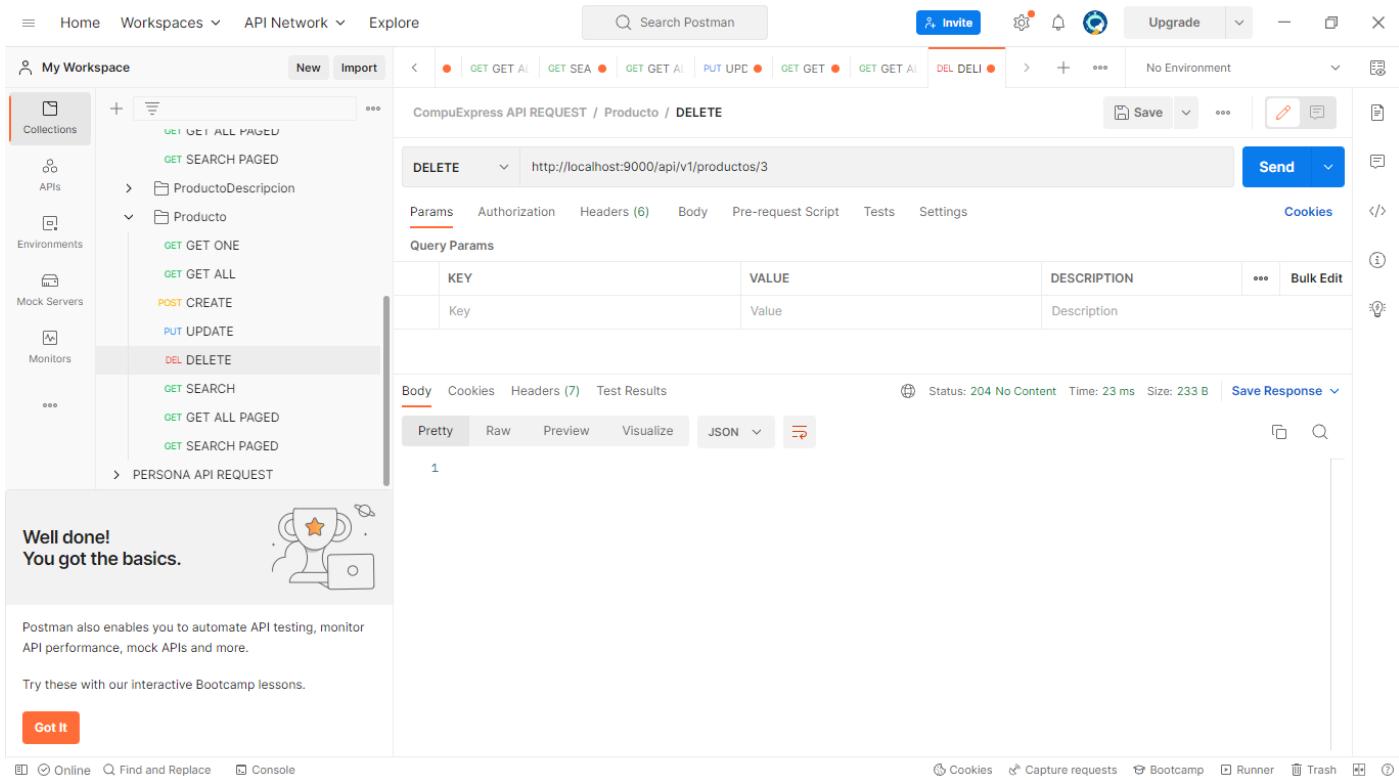
- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors.
- Request URL:** http://localhost:9000/api/v1/productos/
- Method:** GET
- Headers:** (6)
- Body:** Pretty (JSON response shown below)
- Test Results:** Status: 200 OK, Time: 20 ms, Size: 625 B
- Response Body (Pretty JSON):**

```

1  [
2    {
3      "id": 2,
4      "nombreProducto": "Teclado",
5      "codigoProducto": 1001,
6      "precioProducto": 15000.0,
7      "marca": {
8        "id": 2,
9        "codigoMarca": 1001,
10       "nombreMarca": "Dell"
11     },
12     "productoDescripcion": null,
13     "tipoProducto": null
14   },
15   {
16     "id": 3,
17     "nombreProducto": "Mouse",
18     "codigoProducto": 1002,
19     "precioProducto": 8000.0,
20     "marca": {
21       "id": 2,
22       "codigoMarca": 1001,
23       "nombreMarca": "Dell"
24     },
25     "productoDescripcion": null,
26   }
27 ]

```

## Delete



The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors.
- Request URL:** http://localhost:9000/api/v1/productos/3
- Method:** DELETE
- Headers:** (6)
- Query Params:** KEY, VALUE, DESCRIPTION, Bulk Edit
- Body:** Pretty (JSON response shown below)
- Test Results:** Status: 204 No Content, Time: 23 ms, Size: 233 B
- Response Body (Pretty JSON):**

```
1
```

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Collections, APIs, Environments, Mock Servers, and Monitors. A 'Well done! You got the basics.' message with a trophy icon is displayed. The main workspace shows a collection named 'CompuExpress API REQUEST / Producto' containing several requests like 'GET ALL PAGED', 'GET SEARCH PAGED', 'GET GET ONE', and 'GET GET ALL'. A specific request 'GET GET ALL' is selected, showing its details. The request method is 'GET' and the URL is 'http://localhost:9000/api/v1/productos/'. The response body is displayed in JSON format:

```

1  {
2   "id": 2,
3   "nombreProducto": "Teclado",
4   "codigoProducto": 1001,
5   "precioProducto": 15000.0,
6   "marca": {
7     "id": 2,
8     "codigoMarca": 1001,
9     "nombreMarca": "Dell"
10 },
11   "productoDescripcion": null,
12   "tipoProducto": null
13 }
14
15

```

At the bottom, there are buttons for 'Send', 'Save', and 'Cookies'. The status bar at the bottom shows 'Status: 200 OK Time: 14 ms Size: 441 B Save Response'.

## SpringSecurity

The screenshot shows a POST request to 'http://localhost:9000/api/v1/clientes/'. The 'Authorization' tab is selected, showing 'Basic Auth' as the type. A warning message says: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables'.

Params	Authorization	Headers (11)	Body	Pre-request Script	Tests	Settings	Cookies

The 'Authorization' section contains fields for 'Username' (user) and 'Password' (password), with a checked 'Show Password' checkbox.