

# Simple Grocery Store API

This API allows you to place a grocery order which will be ready for pick-up in the store.

The API is available at <https://simple-grocery-store-api.glitch.me>

Alternative URL: <http://simple-grocery-store-api.online/> (HTTP only!)

## Endpoints

- Status
- Products
  - Get all products
  - Get a product
- Cart
  - Get a cart
  - Get cart items
  - Create a new cart
  - Add an item to cart
  - Modify an item in the cart
  - Replace an item in the cart
  - Delete an item in the cart
- Orders
  - Get all orders
  - Get a single order
  - Create a new order
  - Update an order
  - Delete an order
- API Authentication
  - Register a new API client

## Status

**GET /status**

Returns the status of the API. Example response:

```
{  
  "status": "UP"  
}
```



Status `UP` indicates that the API is running as expected.

No response or any other response indicates that the API is not functioning correctly.

# 'Products

## 'Get all products

**GET** /products

Returns a list of products from the inventory.

### Parameters

Name	Type	In	Required	Description
category	string	query	No	Specifies the category of products you want to be returned. It can be one of: meat-seafood, fresh-produce, candy, bread-bakery, dairy, eggs, coffee.
results	integer	query	No	Specifies the number of results you want. Must be number between 1 and 20. By default, only 20 products will be displayed.
available	boolean	query	No	Specifies the availability of the products. By default, all products will be displayed.

### Status codes

Status code	Description
200 OK	Indicates a successful response.
400 Bad Request	Indicates that the parameters provided are invalid.

Example response:

```
[
  {
    "id": 4643,
    "category": "coffee",
    "name": "Starbucks Coffee Variety Pack, 100% Arabica",
    "inStock": true
  },
  {
    "id": 4646,
    "category": "coffee",
    "name": "Ethical Bean Medium Dark Roast, Espresso",
    "inStock": true
  },
  ...
]
```

Copy

## 'Get a product

**GET** /products/:productId

Returns a single product from the inventory.

#### Parameters

Name	Type	In	Required	Description
productId	integer	path	Yes	Specifies the product's id you wish to retrieve.
product-label	boolean	query	No	Returns the product label in PDF format.

#### Status codes

Status code	Description
200 OK	Indicates a successful response.
404 Not found	Indicates that there is no product with the specified id.

## ' Cart

### ' Get a cart

**GET /carts/:cartId**

Returns a cart.

#### Parameters

Name	Type	In	Required	Description
cartId	string	path	Yes	Specifies the id of the cart you wish to retrieve.

#### Status codes

Status code	Description
200 OK	Indicates a successful response.
404 Not found	Indicates that there is no cart with the specified id.

### ' Get cart items

Returns the items in a cart.

**GET /carts/:cartId/items**

#### Parameters

Name	Type	In	Required	Description
cartId	string	path	Yes	Specifies the id of the cart for which you wish to retrieve the items.

#### Status codes

Status code	Description
200 OK	Indicates a successful response.
404 Not found	Indicates that there is no cart with the specified id.

## ’ Create a new cart

To create a new cart, submit an empty POST request to the `/carts` endpoint.

### POST `/carts`

Creates a new cart and returns the id in the response body.

### Parameters

No parameters are accepted for this request.

### Status codes

Status code	Description
201 Created	Indicates that the cart has been created successfully.

Example response body:

```
{
  "created": true,
  "cartId": "bx0-ycNjqIm5IvufuuZ09"
}
```



## ’ Add an item to cart

Allows the addition of items to an existing cart. Only one item can be added at a time.

### POST `/carts/:cartId/items`

The request body needs to be in JSON format.

### Parameters

Name	Type	In	Required	Description
<code>cartId</code>	string	path	Yes	Specifies the cart id.
<code>productId</code>	string	body	Yes	Specifies the product id
<code>quantity</code>	integer	body	No	If no quantity is provided, the default value is 1.

Example request body:

```
{
  "productId": 1234
}
```



}

## Status codes

Status code	Description
201 Created	Indicates that the item has been added successfully.
400 Bad Request	Indicates that the parameters provided are invalid.

## ’ Modify an item in the cart

Allows modifying information about an item in the cart.

**PATCH /cars/:cartId/items/:itemId**

The request body needs to be in JSON format.

## Parameters

Name	Type	In	Required	Description
cartId	string	path	Yes	Specifies the cart id.
itemId	string	path	Yes	Specifies the item id.
quantity	integer	body	Yes	Quantity

## Status codes

Status code	Description
204 No Content	Indicates that the item has been updated successfully.
400 Bad Request	Indicates that the parameters provided are invalid or missing.
404 Not found	The cart or the item could not be found.

## ’ Replace an item in the cart

Replace an item in the cart.

**PUT /cars/:cartId/items/:itemId**

The request body needs to be in JSON format.

## Parameters

Name	Type	In	Required	Description
cartId	string	path	Yes	Specifies the cart id.
itemId	string	path	Yes	Specifies the item id.
productId	integer	body	Yes	Specifies the product id.
quantity	integer	body	No	Quantity

## Status codes

Status code	Description
204 No Content	Indicates that the item has been updated successfully.
400 Bad Request	Indicates that the parameters provided are invalid or missing.
404 Not found	The cart or the item could not be found.

## ’ Delete an item in the cart

Deletes an item in the cart.

**DELETE /cars/:cartId/items/:itemId**

### Parameters

Name	Type	In	Required	Description
cartId	string	path	Yes	Specifies the cart id.
itemId	string	path	Yes	Specifies the item id.

### Status codes

Status code	Description
204 No Content	Indicates that the item has been deleted successfully.
404 Not found	The cart or the item could not be found.

## ’ Orders

### ’ Get all orders

Returns all orders created by the API client.

**GET /orders**

### Parameters

Name	Type	In	Required	Description
Authorization	string	header	Yes	Specifies the bearer token of the API client.

### Status codes

Status code	Description
200 OK	Indicates a successful response.
401 Unauthorized	Indicates that the request has not been authenticated. Check the response body for additional details.

### ’ Get a single order

Returns a single order.

**GET /orders/:orderId**

**Parameters**

Name	Type	In	Required	Description
Authorization	string	header	Yes	The bearer token of the API client.
orderId	string	path	Yes	The order id.
invoice	boolean	query	No	Show the PDF invoice.

**Status codes**

Status code	Description
200 OK	Indicates a successful response.
401 Unauthorized	Indicates that the request has not been authenticated. Check the response body for additional details.
404 Not found	Indicates that there is no order with the specified id associated with the API client.

**Create a new order**

**POST /orders**

The request body needs to be in JSON format. Once the order has been successfully submitted, the cart is deleted.

**Parameters**

Name	Type	In	Required	Description
Authorization	string	header	Yes	The bearer token of the API client.
cartId	string	body	Yes	The cart id
customerName	string	body	Yes	The name of the customer.
comment	string	body	No	A comment associated with the order.

Example request body:

```
{
  "cartId": "ZFe4yhG5qNhmuNyrbLWa4",
  "customerName": "John Doe"
}
```



**Status codes**

Status code	Description
201 Created	Indicates that the order has been created successfully.

Status code	Description
400 Bad Request	Indicates that the parameters provided are invalid.
401 Unauthorized	Indicates that the request has not been authenticated. Check the response body for additional details.

## Update an order

**PATCH /orders/:orderId**

The request body needs to be in JSON format.

### Parameters

Name	Type	In	Required	Description
Authorization	string	header	Yes	The bearer token of the API client.
orderId	string	path	Yes	The order id.
customerName	string	body	No	The name of the customer.
comment	string	body	No	A comment associated with the order.

### Status codes

Status code	Description
204 No Content	Indicates that the order has been updated successfully.
400 Bad Request	Indicates that the parameters provided are invalid.
401 Unauthorized	Indicates that the request has not been authenticated. Check the response body for additional details.
404 Not found	Indicates that there is no order with the specified id associated with the API client.

Example request body:

<pre>{   "customerName": "Joe Doe" }</pre>	
--	---

## Delete an order

**DELETE /orders/:orderId**

### Parameters

Name	Type	In	Required	Description
Authorization	string	header	Yes	The bearer token of the API client.



Name	Type	In	Required	Description
orderId	string	path	Yes	The order id.

## Status codes

Status code	Description
204 No Content	Indicates that the order has been deleted successfully.
400 Bad Request	Indicates that the parameters provided are invalid.
401 Unauthorized	Indicates that the request has not been authenticated. Check the response body for additional details.
404 Not found	Indicates that there is no order with the specified id associated with the API client.

## API Authentication

Some endpoints may require authentication. To submit or view an order, you need to register your API client and obtain an access token.

The endpoints that require authentication expect a bearer token sent in the `Authorization` header.

Example:

```
Authorization: Bearer YOUR TOKEN
```

## Register a new API client

**POST** `/api-clients`

### Parameters

The request body needs to be in JSON format.

Name	Type	In	Required	Description
clientName	string	body	Yes	The name of the API client.
clientEmail	string	body	Yes	The email address of the API client. *

- The email address DOES NOT need to be real. The email will not be stored on the server.

## Status codes

Status code	Description
201 Created	Indicates that the client has been registered successfully.
400 Bad Request	Indicates that the parameters provided are invalid.
409 Conflict	Indicates that an API client has already been registered with this email address.

Example request body:

```
{  
  "clientName": "Postman on Valentin's computer",  
  "clientEmail": "valentin@example.com"  
}
```



The response body will contain the access token.