

# API Documentation

## Traders Model

Property	Description
Id	The unique identifier by which to identify the trader
name	The name of the trader
email	The email of the trader
storeName	The store name of the trader
phone	The phone number of the trader
password	The password of the trader

## Traders endPoint

endpoint	Method
/auth/signup	POST
/auth/signup	POST

## Signup

```
1-
EndPoint: /auth/signup
Method: POST
JSON Format
{
  "name": "Ahmed",
  "email" : "mostafa.ahmed.salah.222@gmail.com",
  "storeName" : "store-1",
  "phone": "01124284915",
  "password" : "12345mM@",
  "cPassword": "12345mM@"
}
```

# Signin

```
2-
EndPoint: /auth/signin
Method: POST
JSON Format
{
  "email" : "mostafa.ahmed.salah.222@gmail.com",
  "password" : "12345mM@"
}
```

**Note :** Don't forget send token with Bearer key (E-C@a\_M\_) in authorization key in headers in all next end points.

Http Headers

<input checked="" type="checkbox"/>	Accept	*/*
<input checked="" type="checkbox"/>	User-Agent	Thunder Client (https://www.thunderclient.com)
<input checked="" type="checkbox"/>	authorization	E-C@a_M_eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTJ5IiwiaWF0IjE5NzUyMjY0MjE2In0=

## Clients Model

Property	Description
Id	The unique identifier by which to identify the client
Code	The code of the client
Email	The email of the client
Name	The name of the client
Address	The address of the client
City	The city of the client

# Clients endPoint

endPoint	Method
/client	POST
/client?page=1&size=10&sortedField=id&orderBy=DESC	GET
/client/search?page=1&size=5&field=Code&key=code1	GET
/client/1000	PUT
/client/45	DELETE

## Add Clients or add client manually

POST

localhost:3000/client

Query

Headers<sup>3</sup>

Auth

Body<sup>1</sup>

Tests

Pre Run<sup>New</sup>

Json

Xml

Text

Form

Form-encode

Graphql

Binary

Form Fields

☒

Code

code1000

☒

Email

user+999@gmail.com

☒

Name

user+1000

☒

Address

address-1000

☒

City

Cairo

☐

field name

value

Files

☒

excel

Choose File

clients.xlsx

**Note :** can send in Files from Excel file or in Form Fields  
Manually , by default sending occurs from Files if exist

## get Clients

GET localhost:3000/client?page=1&size=10&sortedField=id&orderBy=DESC

Query Headers 3 Auth Body Tests Pre Run New

Query Parameters

<input checked="" type="checkbox"/>	page	1
<input checked="" type="checkbox"/>	size	10
<input checked="" type="checkbox"/>	sortedField	id
<input checked="" type="checkbox"/>	orderBy	DESC

## search for clients

GET localhost:3000/client/search?page=1&size=5&field=Code&key=code1

Query Headers 3 Auth Body Tests Pre Run New

Query Parameters

<input checked="" type="checkbox"/>	page	1
<input checked="" type="checkbox"/>	size	5
<input checked="" type="checkbox"/>	field	Code
<input checked="" type="checkbox"/>	key	code1

## Update client by id

```
4-
EndPoint: /client/1000
Method: PUT
JSON Format
{
  "Code": "code1000",
  "Email": "user+1000@gmail.com",
  "Name": "user+1000",
  "Address": "address-1000",
  "City": "Cairo"
}
```

## delete client by id

```
5-
EndPoint: /client/1001
Method: DELETE
```

## Products Model

Property	Description
Id	The unique identifier by which to identify the product
Code	The code of the product
Name	The name of the product
Quantity	Quantity of the product
Description	Description of the product
Category	Category of the product
Trademark	Trademark of the product

## Products endPoint

endPoint	Method
/product	POST
/product?page=10&size=10&sortedField=createdAt&orderBy=DESC	GET
/product/search?field=id&key=5	GET
/product	PATCH
/client/222	PUT
/client/450	DELETE

## Add Clients or add client manually

POST

localhost:3000/product

Query

Headers<sup>3</sup>

Auth

Body<sup>1</sup>

Tests

Pre Run<sup>New</sup>

Json

Xml

Text

Form

Form-encode

Graphql

Binary

Form Fields

<input checked="" type="checkbox"/>	Code	code-1000
<input checked="" type="checkbox"/>	Name	name-1000
<input checked="" type="checkbox"/>	Quantity	22
<input checked="" type="checkbox"/>	Description	description-1000
<input checked="" type="checkbox"/>	Category	category-1000
<input checked="" type="checkbox"/>	Trademark	tardemark-1000
<input type="checkbox"/>	field name	value

Files

<input checked="" type="checkbox"/>	excel	<div>Choose File</div> products.xlsx
-------------------------------------	-------	--------------------------------------

**Note :** Can send in Files from Excel file or in Form Fields  
Manually , by default sending occurs from Files if exist

GET

localhost:3000/product?page=10&size=10&sortedField=createdAt&orderBy=DESC

Query

Headers<sup>3</sup>

Auth

Body

Tests

Pre Run<sup>New</sup>

Query Parameters

<input checked="" type="checkbox"/>	page	10
<input checked="" type="checkbox"/>	size	10
<input checked="" type="checkbox"/>	sortedField	createdAt
<input checked="" type="checkbox"/>	orderBy	DESC

get Clients

## search for products

GET localhost:3000/product/search?page=1&size=3&field=id&key=5

Query Headers<sup>3</sup> Auth Body Tests Pre Run <sup>New</sup>

Query Parameters

<input checked="" type="checkbox"/>	page	1
<input checked="" type="checkbox"/>	size	3
<input checked="" type="checkbox"/>	field	id
<input checked="" type="checkbox"/>	key	5

## Update products quantities from excel sheet

PATCH localhost:3000/product

Query Headers<sup>3</sup> Auth Body<sup>1</sup> Tests Pre Run <sup>New</sup>

Json Xml Text Form Form-encode Graphql Binary

Form Fields

<input type="checkbox"/>	field name	value
--------------------------	------------	-------

Files

☒ excel 

Choose File

 product-update.xlsx

## Update product by id

```
4-
EndPoint: /product/300
Method: PUT
JSON Format
{
  "Code": "code-1000",
  "Name": "name-1000",
  "Quantity": 222,
  "Description": "description-1000",
  "Category": "category-1000",
  "Trademark": "tardemark-1000"
}
```

## delete product by id

```
6-  
EndPoint: /product/212  
Method: DELETE
```

## Orders Model

property	Description
Id	The unique identifier by which to identify the order
clientID	The ID of the client
productID	The ID of the product
Quantity	The ID of the Quantity

## Orders endPoint

endPoint	Method
/order	POST
/order?page=1&size=3&sortedField=clientCode	GET
/order/search?page=1&size=2&field=City&key=c	GET
/order/12	DELETE

## create Order

```
1-  
EndPoint: /order  
Method: POST  
JSON Format  
{  
  "clientID" : 1145,  
  "productID": 1364,  
  "Quantity": 4  
}
```



## get Orders

GET localhost:3000/order?page=1&size=3&sortedField=clientCode&orderBy=DESC

Query Headers 3 Auth Body Tests Pre Run [New](#)

Query Parameters

<input checked="" type="checkbox"/>	page	1
<input checked="" type="checkbox"/>	size	3
<input checked="" type="checkbox"/>	sortedField	clientCode
<input checked="" type="checkbox"/>	orderBy	DESC

## search for orders

GET localhost:3000/order/search?page=1&size=2&field=City&key=c

Query Headers 3 Auth Body Tests Pre Run [New](#)

Query Parameters

<input checked="" type="checkbox"/>	page	1
<input checked="" type="checkbox"/>	size	2
<input checked="" type="checkbox"/>	field	City
<input checked="" type="checkbox"/>	key	c

## delete order by id

```
4-
EndPoint: /order/500
Method: DELETE
```

# Notifications Model

property	Description
Id	The unique identifier by which to identify the notification
notification	notification or offer which defined by trader

## Notifications endPoint

endPoint	Method
/notification	POST
/notification?sortedField=createdAt&orderBy=DESC	GET
/notification/56	PATCH
/notification/22	DELETE

### add notification

```
1-
EndPoint: /notification
Method: POST
JSON Format
{
    "notification": "offer-1"
}
```

### get notifications

GET

localhost:3000/notification?sortedField=createdAt&orderBy=DESC

Query

Headers<sup>3</sup>

Auth

Body

Tests

Pre Run

New

Query Parameters

☒sortedField

createdAt

☒orderBy

DESC

## update notification by id

```
3-  
EndPoint: /notification/222  
Method: PATCH  
JSON Format  
{  
    "notification": "notification-5"  
}
```

## delete notification by id

```
4-  
EndPoint: /notification/799  
Method: DELETE
```

**BASE\_URL:** <http://localhost:3000/>

