

Backend API methods

The screenshot shows the Swagger API Documentation interface in a Safari browser. The browser's address bar shows 'localhost'. The Swagger logo is in the top left, and a 'Select a spec' dropdown is in the top right, currently set to 'default'. The main heading is 'Api Documentation' with a version '1.0' badge. Below it, the base URL is '[Base URL: localhost:8081/]' and the full URL is 'http://localhost:8081/v2/api-docs'. There are links for 'Terms of service' and 'Apache 2.0'. A list of API controllers is shown, each with a right-pointing chevron: 'basic-error-controller' (Basic Error Controller), 'location-controller' (Location Controller), 'product-category-controller' (Product Category Controller), 'product-controller' (Product Controller), and 'Models'.

Swagger

Select a spec: default

Api Documentation ^{1.0}

[Base URL: localhost:8081/]
<http://localhost:8081/v2/api-docs>

Api Documentation
[Terms of service](#)
[Apache 2.0](#)

- basic-error-controller** Basic Error Controller
- location-controller** Location Controller
- product-category-controller** Product Category Controller
- product-controller** Product Controller
- Models

This block shows a detailed view of the API endpoints for two controllers. The 'location-controller' section lists four endpoints: POST /api/location/add (addLocation), DELETE /api/location/delete (deleteLocation), GET /api/location/retrieve (retrieveAllLocation), and PUT /api/location/update (updateLocation). The 'product-category-controller' section lists four endpoints: POST /api/product-category/add (addProductCategory), DELETE /api/product-category/delete (deleteProductCategory), GET /api/product-category/retrieve (retrieveAllProductCategory), and PUT /api/product-category/update (updateProductCategory). Each endpoint is color-coded: green for POST, red for DELETE, blue for GET, and orange for PUT.

location-controller Location Controller

- POST** /api/location/add addLocation
- DELETE** /api/location/delete deleteLocation
- GET** /api/location/retrieve retrieveAllLocation
- PUT** /api/location/update updateLocation

product-category-controller Product Category Controller

- POST** /api/product-category/add addProductCategory
- DELETE** /api/product-category/delete deleteProductCategory
- GET** /api/product-category/retrieve retrieveAllProductCategory
- PUT** /api/product-category/update updateProductCategory

product-controller Product Controller

POST	/api/products/add	addProduct
DELETE	/api/products/delete	deleteProduct
GET	/api/products/retrieve	retrieveAllProduct
GET	/api/products/retrieveById	retrieveProductById
GET	/api/products/search/{productName}	searchProduct
PUT	/api/products/update	updateProduct

Spring Boot Actuator: Monitor and manage application.

```
localhost:8081/actuator
{"_links":{"self":{"href":"http://localhost:8081/actuator","templated":false},"beans":{"href":"http://localhost:8081/actuator/beans","templated":false},"caches-cache":{"href":"http://localhost:8081/actuator/caches/{cache}","templated":true},"caches":{"href":"http://localhost:8081/actuator/caches","templated":false},"health":{"href":"http://localhost:8081/actuator/health/{path}","templated":true},"info":{"href":"http://localhost:8081/actuator/info","templated":false},"conditions":{"href":"http://localhost:8081/actuator/conditions","templated":false},"configprops-prefix":{"href":"http://localhost:8081/actuator/configprops/{prefix}","templated":true},"configprops":{"href":"http://localhost:8081/actuator/configprops","templated":false},"env":{"href":"http://localhost:8081/actuator/env","templated":false},"env-toMatch":{"href":"http://localhost:8081/actuator/env/{toMatch}","templated":true},"loggers":{"href":"http://localhost:8081/actuator/loggers","templated":false},"loggers-name":{"href":"http://localhost:8081/actuator/loggers/{name}","templated":true},"heapdump":{"href":"http://localhost:8081/actuator/heapdump","templated":false},"threaddump":{"href":"http://localhost:8081/actuator/threaddump","templated":false},"metrics-requiredMetricName":{"href":"http://localhost:8081/actuator/metrics/{requiredMetricName}","templated":true},"metrics":{"href":"http://localhost:8081/actuator/metrics","templated":false},"scheduledtasks":{"href":"http://localhost:8081/actuator/scheduledtasks","templated":false},"mappings":{"href":"http://localhost:8081/actuator/mappings","templated":false}}}
```

localhost

localhost:8081/actuator warehouse-ui

```
{
  "links": {
    "self": {
      "href": "http://localhost:8081/actuator",
      "templated": false
    },
    "beans": {
      "href": "http://localhost:8081/actuator/beans",
      "templated": false
    },
    "caches-cache": {
      "href": "http://localhost:8081/actuator/caches/{cache}",
      "templated": true
    },
    "caches": {
      "href": "http://localhost:8081/actuator/caches",
      "templated": false
    },
    "health": {
      "href": "http://localhost:8081/actuator/health",
      "templated": false
    },
    "health-path": {
      "href": "http://localhost:8081/actuator/health/{*path}",
      "templated": true
    },
    "info": {
      "href": "http://localhost:8081/actuator/info",
      "templated": false
    },
    "conditions": {
      "href": "http://localhost:8081/actuator/conditions",
      "templated": false
    },
    "configprops-prefix": {
      "href": "http://localhost:8081/actuator/configprops/{prefix}",
      "templated": true
    },
    "configprops": {
      "href": "http://localhost:8081/actuator/configprops",
      "templated": false
    },
    "env": {
      "href": "http://localhost:8081/actuator/env",
      "templated": false
    },
    "env-toMatch": {
      "href": "http://localhost:8081/actuator/env/{toMatch}",
      "templated": true
    },
    "loggers": {
      "href": "http://localhost:8081/actuator/loggers",
      "templated": false
    },
    "loggers-name": {
      "href": "http://localhost:8081/actuator/loggers/{name}",
      "templated": true
    },
    "heapdump": {
      "href": "http://localhost:8081/actuator/heapdump",
      "templated": false
    },
    "threaddump": {
      "href": "http://localhost:8081/actuator/threaddump",
      "templated": false
    },
    "metrics-requiredMetricName": {
      "href": "http://localhost:8081/actuator/metrics/{requiredMetricName}",
      "templated": true
    },
    "metrics": {
      "href": "http://localhost:8081/actuator/metrics",
      "templated": false
    },
    "scheduledtasks": {
      "href": "http://localhost:8081/actuator/scheduledtasks",
      "templated": false
    },
    "mappings": {
      "href": "http://localhost:8081/actuator/mappings",
      "templated": false
    }
  }
}
```

localhost

localhost:8081/actuator/health warehouse-ui

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

Frontend Screens

Product list (screen calls before mount api/products/retrieve methos it return all products.)

The screenshot shows a web browser window with the URL 'localhost'. The page title is 'WAREHOUSE PROJECT'. Below the title, there is a search bar with the placeholder text 'Search Products' and a green 'SEARCH' button. The main content is a table with 6 columns: 'Product Id ↑', 'Product Name', 'Product Category Name', 'Price', 'Quantity', and 'Location Information'. The table contains 5 rows of data. Below the table, there is a pagination bar showing 'Rows per page: 5' and '1-5 of 18'. At the bottom of the page, there are five green buttons: 'ADD LOCATION', 'ADD CATEGORY', 'ADD PRODUCT', 'ALL LOCATION', and 'ALL PRODUCT CATEGORY'.

Product Id ↑	Product Name	Product Category Name	Price	Quantity	Location Information
1	Desktop Computer	Elektronik	35000	5	İzzettin Çalışlar Street Oğul Apt. No:34
2	Laptop	Elektronik	25000	15	İzzettin Çalışlar Street Oğul Apt. No:34
3	Macbook Pro	Elektronik	45000	7	Tokoğlu District Kemal Street.No:47
4	Macbook Air	Elektronik	38500	3	Tokoğlu District Kemal Street.No:47
5	Face Cream	Cosmetics	700	120	İzzettin Çalışlar Street Oğul Apt. No:34

Product list another page

The screenshot shows a web browser window with the URL 'localhost'. The page title is 'WAREHOUSE PROJECT'. Below the title, there is a search bar with the placeholder text 'Search Products' and a green 'SEARCH' button. The main content is a table with 6 columns: 'Product Id ↑', 'Product Name', 'Product Category Name', 'Price', 'Quantity', and 'Location Information'. The table contains 5 rows of data. Below the table, there is a pagination bar showing 'Rows per page: 5' and '6-10 of 18'. At the bottom of the page, there are five green buttons: 'ADD LOCATION', 'ADD CATEGORY', 'ADD PRODUCT', 'ALL LOCATION', and 'ALL PRODUCT CATEGORY'.

Product Id ↑	Product Name	Product Category Name	Price	Quantity	Location Information
6	Body Cream	Cosmetics	200	150	Beethovenstraat 8, 1077 JG
7	Moisturizer	Cosmetics	300	70	Beethovenstraat 8, 1077 JG
8	Vacuum Cleaner	Home and Living	4000	20	Hilal, Turan Güneş Biv. No:64 D:B
9	Tea Machine	Kitchen	1500	110	Beethovenstraat 8, 1077 JG
10	Coffee Machine	Kitchen	5750	185	Beethovenstraat 8, 1077 JG

Searched value call /api/products/search/{productName} method. The application searches the free text word in the product name field and sorts it alphabetically.

WAREHOUSE PROJECT

cream

SEARCH

Product Id ↑	Product Name	Product Category Name	Price	Quantity	Location Information
5	Face Cream	Cosmetics	700	120	Izzettin Çalışlar Street Oğul Apt. No:34
6	Body Cream	Cosmetics	200	150	Beethovenstraat 8, 1077 JG

Rows per page: 5 1-2 of 2 < >

ADD LOCATION

ADD CATEGORY

ADD PRODUCT

ALL LOCATION

ALL PRODUCT CATEGORY

List All Product category. Call /api/product-category/retrieve method.

WAREHOUSE PROJECT

All Product Category

Id ↑	Category Name
1	Elektronik
2	Cosmetics
3	Home and Living
4	Kitchen
5	Sport

Rows per page: 5 1-5 of 8 < >

BACK

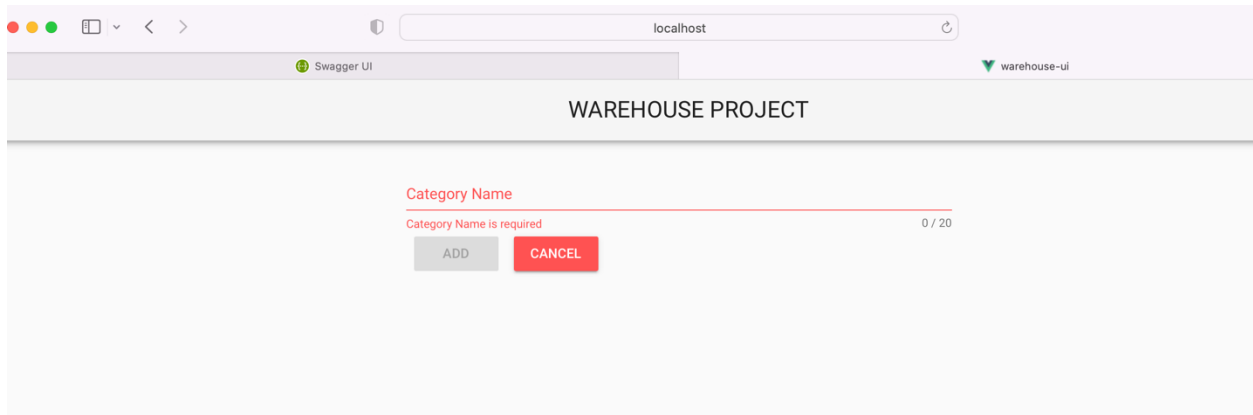
List all location. Call /api/location/retrieve

WAREHOUSE PROJECT			
All Location			
Id ↑	Adress	City	Country
1	İzzettin Çalışlar Street Oğul Apt. No:34	Bahçelievler	İstanbul
2	Tokoğlu District Kemal Street.No:47	Bergama	İzmir
3	Beethovenstraat 8, 1077 JG	Amsterdam	Netherlands
4	Hilal, Turan Güneş Blv. No:64 D:B	Çankaya	Ankara
Rows per page: 5 1-4 of 4 < >			
BACK			

Adding new Product Screen with onscreen controls. Call /api/products/add post method.

WAREHOUSE PROJECT	
Product Name	
Product Name is required	0 / 20
Price	
0	
Price is required	
Quantity	
0	
Product Category Id	
0	
Location Id	
0	
ADD	CANCEL

Adding new Product Category with onscreen controls..



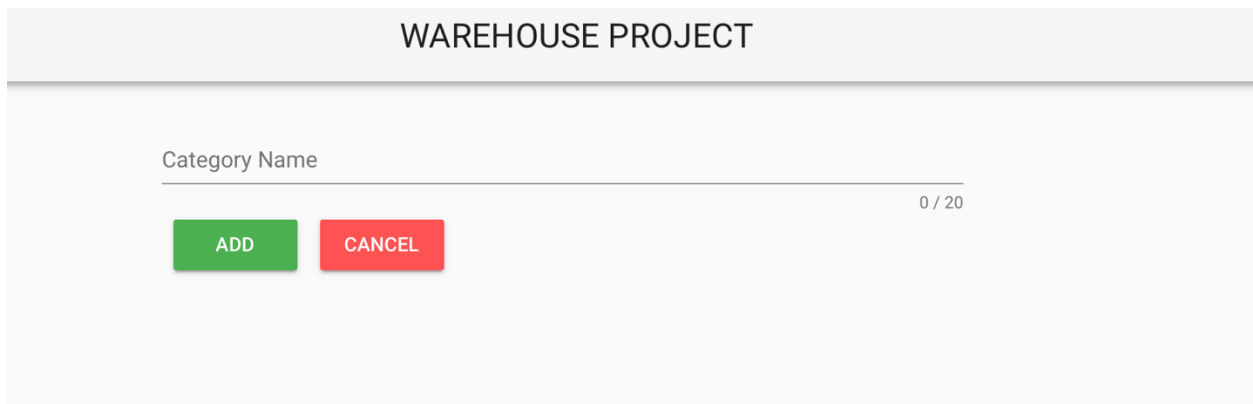
WAREHOUSE PROJECT

Category Name

Category Name is required 0 / 20

ADD CANCEL

call /api/product-category/add post method



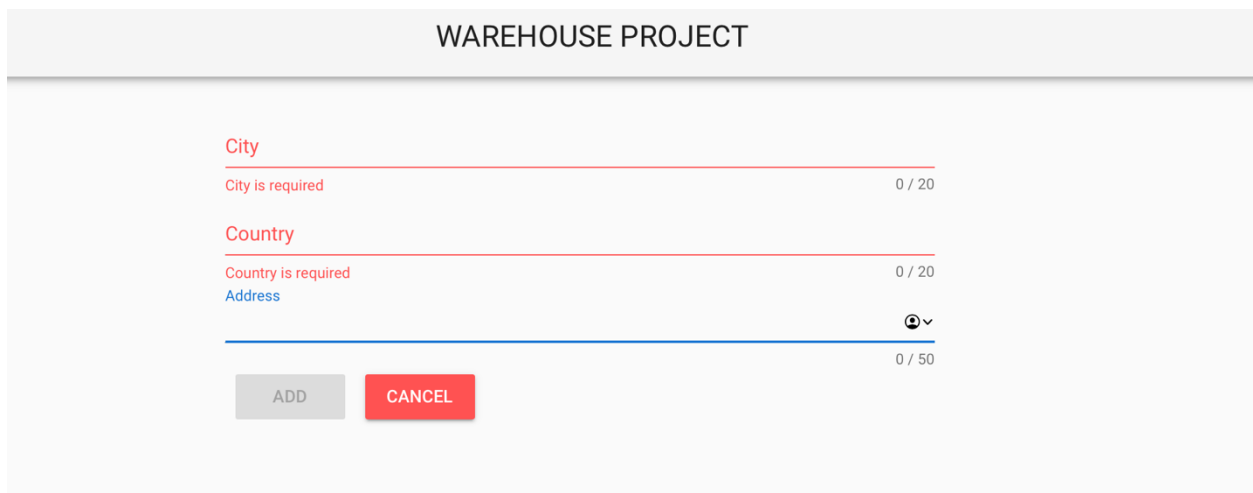
WAREHOUSE PROJECT

Category Name

0 / 20

ADD CANCEL

Adding new Location with onscreen controls.



WAREHOUSE PROJECT

City

City is required 0 / 20

Country

Country is required 0 / 20

Address 0 / 50

ADD CANCEL

Call /api/location/add post api.

The screenshot shows a web browser window with the address bar set to 'localhost'. The browser tabs include 'Swagger UI' and 'warehouse-ui'. The main content area is titled 'WAREHOUSE PROJECT' and contains a form with the following fields:

- City**: Mock City (9 / 20)
- Country**: Mock Country (12 / 20)
- Address**: Mock Address (12 / 50)

The 'Address' field is highlighted with a blue border. Below the form are two buttons: a green 'ADD' button and a red 'CANCEL' button.