

API documentation

Base URL

<http://localhost:8000/>

Authentication

Null

Endpoints

`GET` /customers`

Get all the customers.

Response

```
`200 done` : [
  [
    "Name" (string),
    "Email" (string),
    "Address" (string)
  ]
]
`404 Not found` : returns "No data found"
```

`POST` /customers`

Add new customer to database

Request body

```
`name (string)` : (required) name of the customer
`email (string)` : (required,primary key) email id of the customer
`address (string)` : (required) address of the customer
```

Response

```
`200 done` : returns "Customer added"
`400 Bad Request` : if the request body is missing any required fields.
```

`PUT` /customers/:email`

Update name,address of existing customer

Path parameter

```
`email (string)` (required) : email of the customer
```

Request body

`name (string)` : (required) name of the customer
`address (string)` : (required) address of the customer

Response

`200 done` : returns "Your details are updated"
`404 Not found` : returns "Email not found"
`400 Bad Request` : if the request body is missing any required fields.

`DELETE /customers/:email`

Delete the customer

Path parameter

`email (string)` (required) : email of the customer

Response

`200 done` : returns "Customer deleted"
`404 Not found` : returns "Customer not found ".

`GET /products`

Get all the Products if found else get "No data found" message.

Response

```
`200 done` : [
  [
    Id (int),
    "Name" (string),
    "Customer_email" (string),
    Price (int),
    "Timestamp" (string),
    Active (boolean)
  ]
]
```

`404 Not found` : returns "No data found"

`POST /products`

Add new product to database

Request body

`name (string)` : (required) name of the product
`customer_email (string)` : (required) email id of the customer
`price (int)` : (required) price of the product

Response

`200 done` : returns "Product added"
`404 Not found` : returns "Customer email not found in database".
`400 Bad Request` : if the request body is missing any required fields.

`PUT /products/:id`

Update name,price of existing product

Path parameter

`id (int)` (required) : id of the product

Request body

`name (string)` : (required) name of the product
`price (int)` : (required) price of the product

Response

`200 done` : returns "Your details are updated"
`404 Not found` : returns "Product not found"
`400 Bad Request` : if the request body is missing any required fields.

`DELETE /products/:id`

Delete the Product

Path parameter

`id (int)` (required) : id of the product

Response

`200 done` : returns "Product deleted"
`404 Not found` : returns "Product not found ".

`PUT` `/products/:id/:status``

Update visibility status of any existing product

Path parameter

 `id (int)` (required) : id of the product

 `status (boolean)` (required) : 1 for activation / 0 for deactivation

Response

 `200 done` : returns "Product activated / deactivated"

 `403 forbidden` : returns "Product can only be deactivated after _ days"

 `404 Not found` : returns "Product not found"