**List of resources**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Resource path** | **POST** | **GET** | **PUT** | **DELETE** |
| /products |  | Get most important information about all products that are offered. |  |  |
| /:id(\\d+)/ |  | Get all details about a specific product. |  |  |
| /categories |  | Get product categories that exist. |  |  |
| /prodcat/:id |  | Get most important information about products for a specific category. |  |  |
| /user |  |  | Put the userId and password, then verify by server and if information is right, then get a token send back from the server side |  |
| /user | Post the user information and find if the user registered, if not then register it and return the token. |  |  |  |
| /user |  | Get the user information |  |  |
| /cart |  |  | Create the cart if not exist |  |
| /products | Add the product to the cart by using productid and productNumber |  |  |  |
| /cart |  | Get the information about user’s cart |  |  |
| /cart | Update the whole cart |  |  |  |
| /:id |  |  |  | Delete the product from user’s cart |
| /cart |  |  |  | Delete user’s cart |

**Path: /products**

**Method: GET**

**Summary:** Get most important information about all products that are offered.

**URL Params:** -

**Body:** Customer JSON data

For example: [{ "productId": 1, "productName": "Matcha Bubbletea", "price": 34.99, "description": "Match Bubbletea is perfect for crunch time. Its loaded with antioxidants and sugar. It comes with sweet matcha jelly, tea mixture and red beans", "category": "drinks" }, … ]

**Success Response:**

**Code:** -  
Body Content: -

**Error Response:**

**Code:** -  
Body Content: -

**Sample Call:**

let response = await fetch('/products,

{ method: 'GET',

headers: {'Content-Type': 'application/json;charset=utf-8'},

});

**Path: /:id(\\d+)/**

**Method: GET**

**Summary:** Get all details about a specific product.

**URL Params:**

Id(\\d+): number, required in path

**Body:** Customer JSON data

For example: { "productId": 1, "productName": "Matcha Bubbletea", "price": 34.99, "description": "Match Bubbletea is perfect for crunch time. Its loaded with antioxidants and sugar. It comes with sweet matcha jelly, tea mixture and red beans", "category": "drinks" }

**Success Response:**

**Code:** -  
Body Content: -

**Error Response:**

**Code:** -  
Body Content: -

**Sample Call:**

let id = 4

let response = await fetch(´/:id(\\d+)/´,

{ method: 'GET',

headers: {'Content-Type': 'application/json;charset=utf-8'},

});

**Path: /categories**

**Method: GET**

**Summary:** Get product categories that exist.

**URL Params:** -

**Body:** Customer JSON data

For example: {"drinks", "snacks", "meals"}

**Success Response:**

**Code:** -  
Body Content: -

**Error Response:**

**Code:** -  
Body Content: -

**Sample Call:**

let response = await fetch('/categories,

{ method: 'GET',

headers: {'Content-Type': 'application/json;charset=utf-8'},

});

**Path: /prodcat/:id**

**Method: GET**

**Summary:** Get most important information about products for a specific category.

**URL Params:**

id: number, required in path

**Body:** Customer JSON data

For example: [{ "productId": 1, "productName": "Matcha Bubbletea", "price": 34.99, "description": "Match Bubbletea is perfect for crunch time. Its loaded with antioxidants and sugar. It comes with sweet matcha jelly, tea mixture and red beans", "category": "drinks" }…]

**Success Response:**

**Code:** -  
Body Content: -

**Error Response:**

**Code:** -  
Body Content: -

**Sample Call:**

let response = await fetch('/prodcat/:id,

{ method: 'GET',

headers: {'Content-Type': 'application/json;charset=utf-8'},

});

**Path: /user**

**Method: PUT**

**Summary:** Put the userId and password, then verify by server and if information is right, then get a token send back from the server side

**URL Params:**

**Body:** Customer JSON data

For example: {

"Id":4,

"Password": "123"

}

**Success Response:**

**Code:** -200  
Body Content: -{

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOjQsImlhdCI6MTY1MDQ4NzkzMiwiZXhwIjoxNjUwNDkxNTMyfQ.vGozWbe3hNYOCimlxat27eu2lbtr7K\_E1FtPPQk-eYI"

}

**Error Response:**

**Code:** -401  
Body Content: -User not found or password is incorrect

**Sample Call:**

let response = await fetch('/user,

{ method: 'PUT',

headers: {'Content-Type': 'application/json;charset=utf-8'},

body: JSON.stringify({

"Id":4,

"Password": "123"

})

});

**Path: /user**

**Method: GET**

**Summary:**  Get the user information

**URL Params:**

**Body:** none

For example:

**Success Response:**

**Code:** -200  
Body Content: - {

"userInformation": {

"Id": 4,

"Email": "Ethan3",

"FirstName": "cat",

"LastName": "more",

"Password": "123"

}

}

**Error Response:**

**Code:** -401  
Body Content: Invalid Token

**Sample Call:**

let response = await fetch('/user,

{

method: 'GET',

headers: {

'Authorization': 'Bearer ' + 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOjQsImlhdCI6MTY1MDQ4NzkzMiwiZXhwIjoxNjUwNDkxNTMyfQ.vGozWbe3hNYOCimlxat27eu2lbtr7K\_E1FtPPQk-eYI',

},

});

**Path: /user**

**Method: POST**

**Summary:** Post the user information and find if the user registered, if not then register it and return the token.

**URL Params:**

**Body:** Customer JSON data

For example:

{

"Id":78,

"Password": "123",

"Email": "taiguanran@gmail.com",

"FirstName":"Guanran",

"LastName" :"Tai"

}

**Success Response:**

**Code:** -200  
Body Content: - {

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOjM4LCJpYXQiOjE2NTA0ODkyNDksImV4cCI6MTY1MDQ5Mjg0OX0.5HqGr3Z1btvm\_pjJmoIkFBy2\_hSD87qrkcPtwfzJDJI"

}

**Error Response:**

**Code:** -401  
Body Content: - User already exists

**Sample Call:**

let response = await fetch('/user,

{ method: 'POST',

headers: {'Content-Type': 'application/json;charset=utf-8'},

body: JSON.stringify({

"Id":78,

"Password": "123",

"Email": "taiguanran@gmail.com",

"FirstName":"Guanran",

"LastName" :"Tai"

}),

});

**Path: /products**

**Method: POST**

**Summary:** Add the product to the cart by using productid and productNumber

**URL Params:** -

**Body:** Customer JSON data

For example:

{

"productId":3,

"productNumber": 1

}

**Success Response:**

**Code:** -200  
Body Content: Product added to cart

**Error Response:**

**Code:** -401  
Body Content: - Invalid Token

**Sample Call:**

**let** response = await fetch("http://localhost:3000/products", {

method: "POST",

headers: {

'Content-Type': 'application/json;charset=utf-8',

Authorization:

"Bearer " +

"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOjQsImlhdCI6MTY1MDQ4NzkzMiwiZXhwIjoxNjUwNDkxNTMyfQ.vGozWbe3hNYOCimlxat27eu2lbtr7K\_E1FtPPQk-eYI",

},

body: JSON.stringify({

productId: **3**,

productNumber: **1**,

}),

});

**Path: /cart**

**Method: GET**

**Summary:** Get the information about user’s cart

**URL Params:** -

**Body:** none

For example:

**Success Response:**

**Code:** -200  
Body Content:

{

"cart": [

{

"productId": 3,

"productNumber": 1

}

]

}

**Error Response:**

**Code:** -401  
Body Content: - Invalid Token

**Sample Call:**

**let** response = await fetch("http://localhost:3000/cart", {

method: "GET",

headers: {

Authorization:

"Bearer " +

"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOjQsImlhdCI6MTY1MDQ4NzkzMiwiZXhwIjoxNjUwNDkxNTMyfQ.vGozWbe3hNYOCimlxat27eu2lbtr7K\_E1FtPPQk-eYI",

},

});

**Path: /cart**

**Method: POST**

**Summary:** Update the whole cart

**URL Params:** -

**Body:** Customer JSON data

For example:

{"cart": [

{

"productId": 1,

"productNumber": 2

},

{

"productId": 2,

"productNumber": 2

}

]

}

**Success Response:**

**Code:** -200  
Body Content:

Cart updated

**Error Response:**

**Code:** -401  
Body Content: - Invalid Token

**Sample Call:**

**let** response = await fetch("http://localhost:3000/cart", {

method: "POST",

headers: {

'Content-Type': 'application/json;charset=utf-8',

Authorization:

"Bearer " +

"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOjQsImlhdCI6MTY1MDQ4NzkzMiwiZXhwIjoxNjUwNDkxNTMyfQ.vGozWbe3hNYOCimlxat27eu2lbtr7K\_E1FtPPQk-eYI",

},

body: JSON.stringify({"cart": [

{

"productId": 1,

"productNumber": 2

},

{

"productId": 2,

"productNumber": 2

}

]

})

,});

**Path: /cart**

**Method: PUT**

**Summary:** Create the cart if not exist

**URL Params:** -

**Body:** none

For example:

**Success Response:**

**Code:** -200  
Body Content:

Cart created

**Error Response:**

**Code:** -401  
Body Content: - Invalid Token

**Sample Call:**

**let** response = await fetch("http://localhost:3000/cart", {

method: "PUT",

headers: { Authorization:

"Bearer " +

"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOjQsImlhdCI6MTY1MDQ4NzkzMiwiZXhwIjoxNjUwNDkxNTMyfQ.vGozWbe3hNYOCimlxat27eu2lbtr7K\_E1FtPPQk-eYI",

},

});

**Path: /cart**

**Method: DELETE**

**Summary:** Delete user’s cart

**URL Params:** -

**Body:** none

For example:

**Success Response:**

**Code:** -200  
Body Content:

Cart deleted

**Error Response:**

**Code:** -401  
Body Content: - Invalid Token

**Sample Call:**

**let** response = await fetch("http://localhost:3000/cart", {

method: "DELETE",

headers: { Authorization:

"Bearer " +

"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOjQsImlhdCI6MTY1MDQ4NzkzMiwiZXhwIjoxNjUwNDkxNTMyfQ.vGozWbe3hNYOCimlxat27eu2lbtr7K\_E1FtPPQk-eYI",

},

});

**Path: /:id**

**Method: GET**

**Summary:** Get all details about a specific product.

**URL Params:**

Id: number, required in path

**Body:none**

**Success Response:**

**Code:** -200  
Body Content: - Product deleted from cart

**Error Response:**

**Code:** -401  
Body Content: - Invalid Token

**Sample Call:**

let id = 4

let response = await fetch(`http://localhost:3000/${id}`,

{ method: 'DELETE',

headers: { Authorization:

"Bearer " +

"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOjQsImlhdCI6MTY1MDQ4OTA5NCwiZXhwIjoxNjUwNDkyNjk0fQ.SpVt-auMfFS8ZkU6WWV4PP7IvAor9fyEPptDib2Bzkk",

},

});