

CRM SYSTEM API DOCUMENTATION

Base URL

All API requests are made to the base URL: <https://pair4project.com/api/>

Resources

Your API includes the following 3 main resources:

- **/login**: Handles user authentication.
 - **/customer**: Manages customer information (listing, adding, updating, deleting).
 - **/product**: Manages product operations (listing, adding, updating, deleting).
 - **/order**: Manages order information (listing, adding, updating, deleting).
-

Routes

All HTTP methods are supported. You can send requests using either **http** or **https**.

Identity Route

1. POST /login

Performs user login.

The request includes the username and password

Customer Routes

1. GET /customer

Returns a list of all **customers**.

2. GET /customer/{id}

Returns a specific **customer** with the given ID.

3. POST /customer

Creates a new **customer**.

4. **PUT /customer/{id}**

Fully updates a specific **customer** by ID.

5. **DELETE /customer/{id}**

Deletes a specific **customer** by ID.

Product Routes

6. **GET / product**

Returns a list of all products.

7. **GET /product/{id}**

Returns a specific **product** with the given ID.

8. **POST /product**

Creates a new product.

9. **PUT /product /{id}**

Fully updates a specific **product** by ID.

10. **DELETE /customer/{id}**

Deletes a specific **product** by ID.

Order Routes

11. **GET / orders**

Returns a list of all orders.

12. **GET /orders/{id}**

Returns a specific **order** with the given ID.

13. **POST /order**

Creates a new order.

14. PUT /order /{id}

Fully updates a specific **order** by ID.

15. DELETE /order/{id}

Deletes a specific **order** by ID.

DETAIL USAGE

1. Authentication: Login

- **Endpoint:** /login
- **Method:** POST
- **Description:** Authenticate a user by providing credentials (username and password).

Request:

- **URL:** /login
- **Method:** POST
- **Headers:**
 - Content-Type: application/json/

Request Body:

```
{  
  "username": "string",  
  "password": "string"  
}
```

Response:

- **200 OK** (Success)

```
{  
  "token": "jwt_token_string"  
}
```

- **400 Bad Request** (Invalid credentials)

```
{  
  "error": "Invalid username or password"  
}
```

2. Customer Management: Customer

GET /customer

- **Description:** Retrieve a list of all customers or a specific customer by ID.

Request:

- **URL:** /customer or /customer/{id}
- **Method:** GET
- **Headers:**
 - Authorization: Bearer {token}

Response:

- **200 OK** (List of customers)

```
[  
  {  
    "address": "String",  
    "city": "String",  
    "contact": "string",  
    "corporateCustomer": "string",  
    "customer": "string",  
    "customerAccount": "string",  
    "district": "string",  
    "gender": "string",  
    "individualCustomer": "string",  
    "type": "string",  
  }  
]
```

- **200 OK** (Single customer by ID)

```
{
```

```
"address": "String",
"city": "String",
"contact!": "string",
"corporateCustomer": "string",
"customer": "string",
"customerAccount": "string",
"district": "string",
"gender": "string",
"induvualCustomer": "string",
"type": "string",
}
```

- **404 Not Found** (Customer not found)

```
{
  "error": "Customer not found"
}
```

POST /customer

- **Description:** Create a new customer.

Request:

- **URL:** /customer
- **Method:** POST
- **Headers:**
 - Authorization: Bearer {token}
 - Content-Type: application/json/

Request Body:

```
{
  "address": "String",
  "city": "String",
  "contact!": "string",
```

```
"corporateCustomer": "string",  
"customer": "string",  
"customerAccount": "string",  
"district": "string",  
"gender": "string",  
"induvualCustomer": "string",  
"type": "string",  
}
```

Response:

- **201 Created** (Customer created)

```
{  
  "address": "String",  
  "city": "String",  
  "contact": "string",  
  "corporateCustomer": "string",  
  "customer": "string",  
  "customerAccount": "string",  
  "district": "string",  
  "gender": "string",  
  "induvualCustomer": "string",  
  "type": "string",  
}
```

- **400 Bad Request** (Invalid request)

```
{  
  "error": "Invalid customer data"
```

```
}
```

PUT /customer/{id}

- **Description:** Update an existing customer's data.

Request:

- **URL:** /customer/{id}
- **Method:** PUT
- **Headers:**
 - Authorization: Bearer {token}
 - Content-Type: application/json/

Request Body:

```
{  
  
  "address": "String",  
  
  "city": "String",  
  
  "contactl": "string",  
  
  "corporateCustomer": "string",  
  
  "customer": "string",  
  
  "customerAccount": "string",  
  
  "district": "string",  
  
  "gender": "string",  
  
  "induvualCustomer": "string",  
  
  "type": "string"  
}
```

Response:

- **200 OK** (Customer updated)

```
{  
  
  "address": "String",  
  
  "city": "String",  
  
  "contactl": "string",  
  
  "corporateCustomer": "string",  
  
  "customer": "string",  
  
  "customerAccount": "string",  
  
  "district": "string",  
  
  "gender": "string",  
  
  "induvualCustomer": "string",  
  
  "type": "string"  
}
```

```
"corporateCustomer": "string",  
"customer": "string",  
"customerAccount": "string",  
"district": "string",  
"gender": "string",  
"induvualCustomer": "string",  
"type": "string",  
}
```

- **404 Not Found** (Customer not found)

```
{  
  "error": "Customer not found"  
}
```

DELETE /customer/{id}

- **Description:** Delete a customer by ID.

Request:

- **URL:** /customer/{id}
- **Method:** DELETE
- **Headers:**
 - Authorization: Bearer {token}

Response:

- **200 OK** (Customer deleted)

```
{  
  "message": "Customer deleted successfully"  
}
```

- **404 Not Found** (Customer not found)

```
{  
  "error": "Customer not found"  
}
```

3. Product Management: Product

GET /product

- **Description:** Retrieve a list of all products or a specific customer by ID.

Request:

- **URL:** /product or /product /{id}
- **Method:** GET
- **Headers:**
 - Authorization: Bearer {token}

Response:

- **200 OK** (List of customers)

```
[  
  {  
    "campaign": "String",  
    "campaignProduct": "String",  
    "catalog": "string",  
    "catalogProductOffer": "string",  
    "characteristic": "string",  
    "charastericticValue": "string",  
    "customerOffer": "string",  
    "product": "string",  
    "productCharachteristicValue": "string",  
    "productOffer": "string",  
  }  
]
```

- **200 OK** (Single customer by ID)

```
{  
  "campaign": "String",
```

```
"campaignProduct": "String",  
"catalog": "string",  
"catalogProductOffer": "string",  
"characteristic": "string",  
"characteristicValue": "string",  
"customerOffer": "string",  
"product": "string",  
"productCharacteristicValue": "string",  
"productOffer": "string",  
  
}
```

- **404 Not Found** (Customer not found)

```
{  
  "error": "Customer not found"  
}
```

POST /product

- **Description:** Create a new product.

Request:

- **URL:** /product
- **Method:** POST
- **Headers:**
 - Authorization: Bearer {token}
 - Content-Type: application/json/

Request Body:

```
{  
  
  "campaign": "String",  
  "campaignProduct": "String",  
  "catalog": "string",  
  "catalogProductOffer": "string",  
  "characteristic": "string",  
  "characteristicValue": "string",  
  "customerOffer": "string",  
  "product": "string",  
  "productCharacteristicValue": "string",  
  "productOffer": "string",  
}
```

```
"catalogProductOffer": "string",
"characteristic": "string",
"charastericticValue": "string",
"customerOffer": "string",
"product": "string",
"productCharachteristicValue": "string",
"productOffer": "string",

}
```

Response:

- **201 Created** (Customer created)

```
{
"campaign": "String",
"campaignProduct": "String",
"catalog": "string",
"catalogProductOffer": "string",
"characteristic": "string",
"charastericticValue": "string",
"customerOffer": "string",
"product": "string",
"productCharachteristicValue": "string",
"productOffer": "string",

}
```

- **400 Bad Request** (Invalid request)

```
{  
  "error": "Invalid customer data"  
}
```

PUT /product /{id}

- **Description:** Update an existing customer's data.

Request:

- **URL:** /product /{id}
- **Method:** PUT
- **Headers:**
 - Authorization: Bearer {token}
 - Content-Type: application/json/

Request Body:

```
{  
  "campaign": "String",  
  "campaignProduct": "String",  
  "catalog": "string",  
  "catalogProductOffer": "string",  
  "characteristic": "string",  
  "charastericticValue": "string",  
  "customerOffer": "string",  
  "product": "string",  
  "productCharachteristicValue": "string",  
  "productOffer": "string",  
}
```

Response:

- **200 OK** (Customer updated)

```
{
```

```
"campaign": "String",
"campaignProduct": "String",
"catalog": "string",
"catalogProductOffer": "string",
"characteristic": "string",
"characteristicValue": "string",
"customerOffer": "string",
"product": "string",
"productCharacteristicValue": "string",
"productOffer": "string",
}

● 404 Not Found (Customer not found)
```

```
{
  "error": "Product not found"
}
```

DELETE /product /{id}

- **Description:** Delete a customer by ID.

Request:

- **URL:** /product /{id}
- **Method:** DELETE
- **Headers:**
 - Authorization: Bearer {token}

Response:

```
● 200 OK (Customer deleted)

{
  "message": "Product deleted successfully"
}

● 404 Not Found (Customer not found)
```

```
{  
  "error": "Product not found"
```

3. Order Management: Order

GET /order

- **Description:** Retrieve a list of all orders or a specific order by ID.

Request:

- **URL:** /order. or /order /{id}
- **Method:** GET
- **Headers:**
 - Authorization: Bearer {token}

Response:

- **200 OK** (List of customers)

```
[  
  {  
    "order": "String",  
    "orderProductOfferItem": "String",  
  }  
]
```

- **200 OK** (Single customer by ID)

```
{  
  "order": "String",  
  "orderProductOfferItem": "String",  
}
```

- **404 Not Found** (Customer not found)

```
{  
  "error": "Order not found"
```

```
}
```

POST /order

- **Description:** Create a new order.

Request:

- **URL:** /order
- **Method:** POST
- **Headers:**
 - Authorization: Bearer {token}
 - Content-Type: application/json/

Request Body:

```
{  
  
  "order": "String",  
  
  "orderProductOfferItem": "String",  
  
}
```

Response:

- **201 Created** (Customer created)

```
{  
  
  "order": "String",  
  
  "orderProductOfferItem": "String",  
  
}
```

- **400 Bad Request** (Invalid request)

```
{  
  
  "error": "Invalid order data"  
  
}
```

PUT /order /{id}

- **Description:** Update an existing orders data.

Request:

- **URL:** /order /{id}
- **Method:** PUT
- **Headers:**
 - Authorization: Bearer {token}
 - Content-Type: application/json/

Request Body:

```
{  
  "order": "String",  
  "orderProductOfferItem": "String",  
}
```

Response:

- **200 OK** (Customer updated)

```
{  
  "order": "String",  
  "orderProductOfferItem": "String",  
}
```

404 Not Found (Customer not found)

```
{  
  "error": "Order not found"  
}
```

DELETE /order /{id}

- **Description:** Delete a order by ID.

Request:

- **URL:** /order /{id}
- **Method:** DELETE
- **Headers:**
 - Authorization: Bearer {token}

Response:

- **200 OK** (Customer deleted)

```
{  
  "message": "Product deleted successfully"  
}
```

- **404 Not Found** (Customer not found)

```
{  
  "error": "Order not found"  
  
}
```