

SALES TRACKING SOLUTION API DOCUMENTATION

Incourage Sale Tracking Solution API (1.0.0)

VERSION	UPDATED	DOCUMENT AUTHOR
1.0.0	20 th May 2024	Austin Mwasi

INTRODUCTION

The Sales Tracking Solution API allows a third-party to access and request services.

In order to make requests, your application must authenticate using an authorization token which the API will assign to you. The Authorization token (Token ID) will be valid for a period of 15minutes on successful login, thereafter, the user will be required to re-login again.

You will only be able to make requests on the services using the provided Authorization Token.

Generally, we will stick to HTTP API guidelines.

- GET to retrieve resource from the API.
- POST to create new resource.
- PUT to update a resource.
- DELETE to delete a resource.

ERROR RESPONSES AND DESCRIPTION

Error Code/Status Code	Descriptions/Possibilities
404	- Unauthorized Access
405	- Invalid Token
501	- User Not Registered
200	- Success Response
11	- Invalid login credentials Username/Password
10	- Parameters errors
12	- Failed Event
13	- User Not Deleted
15	- Invalid User

Description: User login.

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

Accept: application/json

Authentication: Token Required

BaseUrl: http://localhost:3000/

Request

Endpoint	/user/login
Method	POST
Body Params	Username
	Password
	TokenID

Example

http://localhost:3000/POST/user/login

Username="austinmwasi@gmail.com"

Password="M*****24"

TokenID="*****"

Response

Returns a JSON object response string in both success and fail.

On Successful login

```
{
  "StatusCode": 200,
  "Message": "Auth Successful"
}
```

On Failure

```
{
  "StatusCode": 405,
  "Message": "Auth Failed"
}
```

Description: User Register.

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

Accept: application/json

BaseUrl: http://localhost:3000/

Request

Endpoint	/user/signup
Method	POST
Body Params	Username
	Password

Example

http://localhost:3000/POST/user/signup

Username="austinmwasi@gmail.com"

Password="M*****24"

Response

Returns a JSON object response string in both success and fail.

On Successful signup

```
{
  "StatusCode": 200,
  "Message": "User Created Successful"
}
```

On Failure

```
{
  "StatusCode": 12,
  "Message": "Email Exists"
}
```

Description: User Delete.

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

Accept: application/json

BaseUrl: http://localhost:3000/

Request

Endpoint	/user/userId
Method	DELETE
Body Params	UserId

Example

http://localhost:3000/DELETE/user/userId

UserId=5

Response

Returns a JSON object response string in both success and fail.

On **Successful** delete

```
{
  "StatusCode": 200,
  "Message": "User Deleted Successful"
}
```

On **Failure**

```
{
  "StatusCode": 13,
  "Message": "Failed to delete"
}
```

Description: Get All Products.

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

Accept: application/json

BaseUrl: http://localhost:3000/

Request

Endpoint	/products/
Method	GET

Example

http://localhost:3000/GET/products/

Response

Returns a JSON object response string in both success and fail.

On Successful

```
{
  "StatusCode": "200",
  "ProductsInfo":
  [
    {
      "ProductName": "Prod1",
      "ProductId": "1",
      "ProductImage": ""
    },
    {
      "ProductName": "Prod2",
      "ProductId": "2",
      "ProductImage": ""
    },
    {
      "ProductName": "Prod3",
      "ProductId": "3",
      "ProductImage": ""
    }
  ]
}
```

On Failure

```
{
  "StatusCode": 13,
  "Message": "Failed"
}
```

Description: Get Products By Id.

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

Accept: application/json

BaseUrl: http://localhost:3000/

Request

Endpoint	/products/productId
Method	GET
Body Params	ProductId

Example

http://localhost:3000/GET/products/productId

ProductId=5

Response

Returns a JSON object response string in both success and fail.

On Successful

```
{
  "StatusCode": 200,
  "ProductName": "Test Prod",
  "ProductId": "5",
  "ProductImage": "Test Prod"
}
```

On Failure

```
{
  "StatusCode": 13,
  "Message": "Failed"
}
```

Description: Create Products.

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

Accept: application/json

BaseUrl: http://localhost:3000/

Request

Endpoint	/products/
Method	POST
Body Params	ProductName
	ProductPrice
	ProductImage

Example

http://localhost:3000/POST/products/

ProductName="Test Prod2"

ProductPrice="50"

ProductImage=""

Response

Returns a JSON object response string in both success and fail.

On Successful

```
{
  "StatusCode": 200,
  "Message": "Product Created Successful",
  "ProductName": "Test Prod2",
  "ProductId": "7",
  "ProductPrice": "50",
  "ProductImage": "Test Prod2"
}
```

On Failure

```
{
  "StatusCode": 13,
  "Message": "Failed to create product"
}
```

Description: Update Products.

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

Accept: application/json

BaseUrl: http://localhost:3000/

Request

Endpoint	/products/productId
Method	POST
Body Prams	ProductName
	ProductPrice

Example

http://localhost:3000/POST/products/productId

ProductName="Test Prod3"

ProductPrice="70"

Response

Returns a JSON object response string in both success and fail.

On Successful

```
{
  "StatusCode": 200,
  "Message": "Product Updated Successful",
  "ProductName": "Test Prod3",
  "ProductId": "7",
  "ProductPrice": "70"
}
```

On Failure

```
{
  "StatusCode": 13,
  "Message": "Failed to Update product"
}
```


Description: Product Delete.

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

Accept: application/json

BaseUrl: http://localhost:3000/

Request

Endpoint	/products/productId
Method	DELETE
Body Params	ProductId

Example

http://localhost:3000/DELETE/products/productId

ProductId=7

Response

Returns a JSON object response string in both success and fail.

On **Successful** delete

```
{
  "StatusCode": 200,
  "Message": "Product Deleted Successful"
}
```

On **Failure**

```
{
  "StatusCode": 13,
  "Message": "Failed to delete"
}
```

Description: Get All Orders.

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

Accept: application/json

BaseUrl: http://localhost:3000/

Request

Endpoint	/orders/
Method	GET

Example

http://localhost:3000/GET/orders/

Response

Returns a JSON object response string in both success and fail.

On Successful

```
{
  "StatusCode": "200",
  "ProductsInfo":
  [
    {
      "ProductName": "Test Orders1",
      "OrderId": "12",
      "ProductQuantity": "20"
    },
    {
      "ProductName": "Test Orders2",
      "OrderId": "13",
      "ProductQuantity": "40"
    },
    {
      "ProductName": "Test Orders3",
      "OrderId": "14",
      "ProductQuantity": "60"
    }
  ]
}
```

On Failure

```
{
  "StatusCode": 13,
  "Message": "No orders"
}
```

Description: Get Orders By Id.

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

Accept: application/json

BaseUrl: http://localhost:3000/

Request

Endpoint	/orders/orderId
Method	GET
Body Params	OrderId

Example

http://localhost:3000/GET/orders/orderId

OrderId=12

Response

Returns a JSON object response string in both success and fail.

On Successful

```
{
  "StatusCode": 200,
  "ProductName": "Test Order1",
  "OrderId": "12",
  "ProductQuantity": "20"
}
```

On Failure

```
{
  "StatusCode": 13,
  "Message": "No such order found"
}
```

Description: Create Orders.

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

Accept: application/json

BaseUrl: http://localhost:3000/

Request

Endpoint	/orders/
Method	POST
Body Params	ProductName
	ProductQuantity

Example

http://localhost:3000/POST/orders/

ProductName="Test Order40"

ProductQuantity="30"

Response

Returns a JSON object response string in both success and fail.

On Successful

```
{
  "StatusCode": 200,
  "Message": "Order Created Successful",
  "ProductName": "Test Order40",
  "OrderId": "8",
  "ProductQuantity": "30"
}
```

On Failure

```
{
  "StatusCode": 13,
  "Message": "Failed to create order"
}
```

Description: Update Orders.

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

Accept: application/json

BaseUrl: http://localhost:3000/

Request

Endpoint	/orders/orderId
Method	POST
Body Params	ProducName
	ProductQuantity

Example

http://localhost:3000/POST/orders/orderId

ProductName="Test Prod3"

ProductQuantity="90"

Response

Returns a JSON object response string in both success and fail.

On Successful

```
{
  "StatusCode": 200,
  "Message": "Order Updated Successful",
  "ProductName": "Test Prod3",
  "OrderId": "8",
  "ProductQuabtity": "90"
}
```

On Failure

```
{
  "StatusCode": 13,
  "Message": "Failed to Update Order"
}
```

Description: Order Delete.

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

Accept: application/json

BaseUrl: http://localhost:3000/

Request

Endpoint	/orders/orderId
Method	DELETE
Body Params	OrderId

Example

http://localhost:3000/DELETE/orders/orderId

OrderId=7

Response

Returns a JSON object response string in both success and fail.

On **Successful** delete

```
{
  "StatusCode": 200,
  "Message": "Order Deleted Successful"
}
```

On **Failure**

```
{
  "StatusCode": 13,
  "Message": "Failed to delete"
}
```