

ExpensesManager

# Table of Contents

1. Overview .....	1
1.1. Version information .....	1
1.2. License information .....	1
1.3. URI scheme .....	1
1.4. Tags .....	1
2. Resources .....	2
2.1. BillOfSale .....	2
2.1.1. Create bill of sale. ....	2
2.1.2. Retrieve all bills of sale.....	3
2.1.3. Update bill of sale.....	4
2.1.4. Find bill of sale by id.....	5
2.1.5. Update bill of sale by id. ....	6
2.1.6. Remove bill of sale.....	7
2.1.7. Retrieve all budgets.....	8
2.2. Budget .....	9
2.2.1. Create budget. ....	9
2.2.2. Update budget.....	11
2.2.3. Find budget by id. ....	13
2.2.4. Update budget by id. ....	15
2.2.5. Remove budget.....	17
2.3. Order .....	18
2.3.1. Create order. ....	18
2.3.2. Retrieve all product orders. ....	19
2.3.3. Update order.....	20
2.3.4. Find Order by id.....	21
2.3.5. Update order by id.....	22
2.3.6. Remove order. ....	23
2.4. Product .....	24
2.4.1. Create product. ....	24
2.4.2. Retrieve all products.....	25
2.4.3. Update product.....	26
2.4.4. Find product by id. ....	27
2.4.5. Update Product by id.....	28
2.4.6. Remove product.....	29
3. Definitions .....	30
3.1. BillOfSale .....	30
3.2. Budget .....	30
3.3. Product .....	30

3.4. ProductOrder.....	31
------------------------	----

# Chapter 1. Overview

ExpensesManager is a service to expenses management.

## 1.1. Version information

*Version* : 0.5.0

## 1.2. License information

*License* : Proprietary

*Terms of service* : null

## 1.3. URI scheme

*Host* : localhost:8080

*BasePath* : /

*Schemes* : HTTPS

## 1.4. Tags

- BillOfSale
- Budget
- Order
- Product

# Chapter 2. Resources

## 2.1. BillOfSale

### 2.1.1. Create bill of sale.

POST /billofsales

#### Description

Method allow to create a new bill of sale.

#### Responses

HTTP Code	Description	Schema
201	BillOfSale was created.	<a href="#">BillOfSale</a>
409	BillOfSale bought date cannot be null.	No Content

#### Example HTTP request

##### Request path

/billofsales

#### Example HTTP response

##### Response 201

```
{
  "id" : "string",
  "productList" : [ {
    "id" : "string",
    "product" : {
      "id" : "string",
      "name" : "string",
      "price" : 0.0
    },
    "quantity" : 0
  } ],
  "boughtDate" : 0,
  "description" : "string"
}
```

## 2.1.2. Retrieve all bills of sale.

GET /billofsales

### Description

Method allow to find all bills of sale.

### Responses

HTTP Code	Description	Schema
200	BillOfSales were found.	< <a href="#">BillOfSale</a> > array
404	BillOfSales are not found.	No Content

### Example HTTP request

#### Request path

/billofsales

### Example HTTP response

#### Response 200

```
[ {
  "id" : "string",
  "productList" : [ {
    "id" : "string",
    "product" : {
      "id" : "string",
      "name" : "string",
      "price" : 0.0
    },
    "quantity" : 0
  } ],
  "boughtDate" : 0,
  "description" : "string"
} ]
```

### 2.1.3. Update bill of sale.

PUT /billofsales

#### Description

Method allow to update a bill of sale.

#### Responses

HTTP Code	Description	Schema
200	BillOfSale was updated.	<a href="#">BillOfSale</a>
404	BillOfSale not found.	No Content
409	ID has invalid format.	No Content

#### Example HTTP request

##### Request path

/billofsales

#### Example HTTP response

##### Response 200

```
{
  "id" : "string",
  "productList" : [ {
    "id" : "string",
    "product" : {
      "id" : "string",
      "name" : "string",
      "price" : 0.0
    },
    "quantity" : 0
  } ],
  "boughtDate" : 0,
  "description" : "string"
}
```

## 2.1.4. Find bill of sale by id.

```
GET /billofsales/{id}
```

### Description

Method allow to find a bill of sale by id.

### Responses

HTTP Code	Description	Schema
200	BillOfSale was found.	<a href="#">BillOfSale</a>
404	BillOfSale not found.	No Content
409	ID has invalid format.	No Content

### Example HTTP request

#### Request path

```
/billofsales/{id}
```

### Example HTTP response

#### Response 200

```
{
  "id" : "string",
  "productList" : [ {
    "id" : "string",
    "product" : {
      "id" : "string",
      "name" : "string",
      "price" : 0.0
    },
    "quantity" : 0
  } ],
  "boughtDate" : 0,
  "description" : "string"
}
```



## 2.1.5. Update bill of sale by id.

```
PUT /billofsales/{id}
```

### Description

Method allow to update a bill of sale by id.

### Responses

HTTP Code	Description	Schema
200	BillOfSale was updated.	<a href="#">BillOfSale</a>
404	BillOfSale not found.	No Content
409	ID has invalid format.	No Content

### Example HTTP request

#### Request path

```
/billofsales/{id}
```

### Example HTTP response

#### Response 200

```
{
  "id" : "string",
  "productList" : [ {
    "id" : "string",
    "product" : {
      "id" : "string",
      "name" : "string",
      "price" : 0.0
    },
    "quantity" : 0
  } ],
  "boughtDate" : 0,
  "description" : "string"
}
```

## 2.1.6. Remove bill of sale.

```
DELETE /billofsales/{id}
```

### Description

Method allow to remove a bill of sale.

### Responses

HTTP Code	Description	Schema
204	BillOfSale was removed.	No Content
409	ID has invalid format.	No Content

### Example HTTP request

#### Request path

```
/billofsales/{id}
```

## 2.1.7. Retrieve all budgets.

GET /budgets

### Description

Method allow to find all budgets.

### Responses

HTTP Code	Description	Schema
200	Budgets were found.	< <a href="#">Budget</a> > array
404	Budgets are not found.	No Content

### Example HTTP request

#### Request path

/budgets

### Example HTTP response

#### Response 200

```
[ {
  "id" : "string",
  "name" : "string",
  "budgetValue" : 0.0,
  "billsOfSaleList" : [ {
    "id" : "string",
    "productList" : [ {
      "id" : "string",
      "product" : {
        "id" : "string",
        "name" : "string",
        "price" : 0.0
      },
      "quantity" : 0
    } ],
    "boughtDate" : 0,
    "description" : "string"
  } ]
} ]
```

## 2.2. Budget

### 2.2.1. Create budget.

POST /budgets

#### Description

Method allow to create a new budget.

#### Responses

HTTP Code	Description	Schema
201	Budget was created.	<a href="#">Budget</a>
409	Budget value cannot be null or is infinite.	No Content

#### Example HTTP request

##### Request path

/budgets

#### Example HTTP response

##### Response 201

```
{
  "id" : "string",
  "name" : "string",
  "budgetValue" : 0.0,
  "billsOfSaleList" : [ {
    "id" : "string",
    "productList" : [ {
      "id" : "string",
      "product" : {
        "id" : "string",
        "name" : "string",
        "price" : 0.0
      },
      "quantity" : 0
    } ],
    "boughtDate" : 0,
    "description" : "string"
  } ]
}
```

### 2.2.2. Update budget.

PUT /budgets

#### Description

Method allow to update a budget.

#### Responses

HTTP Code	Description	Schema
200	Budget was updated.	<a href="#">Budget</a>
404	Budget not found.	No Content
409	ID has invalid format.	No Content

#### Example HTTP request

##### Request path

/budgets

#### Example HTTP response

##### Response 200

```
{
  "id" : "string",
  "name" : "string",
  "budgetValue" : 0.0,
  "billsOfSaleList" : [ {
    "id" : "string",
    "productList" : [ {
      "id" : "string",
      "product" : {
        "id" : "string",
        "name" : "string",
        "price" : 0.0
      },
      "quantity" : 0
    } ],
    "boughtDate" : 0,
    "description" : "string"
  } ]
}
```

### 2.2.3. Find budget by id.

```
GET /budgets/{id}
```

#### Description

Method allow to find a budget by id.

#### Responses

HTTP Code	Description	Schema
200	Budget was found.	<a href="#">Budget</a>
404	Budget not found.	No Content
409	ID has invalid format.	No Content

#### Example HTTP request

##### Request path

```
/budgets/{id}
```

#### Example HTTP response

##### Response 200



```
{
  "id" : "string",
  "name" : "string",
  "budgetValue" : 0.0,
  "billsOfSaleList" : [ {
    "id" : "string",
    "productList" : [ {
      "id" : "string",
      "product" : {
        "id" : "string",
        "name" : "string",
        "price" : 0.0
      },
      "quantity" : 0
    } ],
    "boughtDate" : 0,
    "description" : "string"
  } ]
}
```

## 2.2.4. Update budget by id.

PUT /budgets/{id}

### Description

Method allow to update a budget by id.

### Responses

HTTP Code	Description	Schema
200	Budget was updated.	<a href="#">Budget</a>
404	Budget not found.	No Content
409	ID has invalid format.	No Content

### Example HTTP request

#### Request path

/budgets/{id}

### Example HTTP response

#### Response 200

```
{
  "id" : "string",
  "name" : "string",
  "budgetValue" : 0.0,
  "billsOfSaleList" : [ {
    "id" : "string",
    "productList" : [ {
      "id" : "string",
      "product" : {
        "id" : "string",
        "name" : "string",
        "price" : 0.0
      },
      "quantity" : 0
    } ],
    "boughtDate" : 0,
    "description" : "string"
  } ]
}
```

### 2.2.5. Remove budget.

```
DELETE /budgets/{id}
```

#### Description

Method allow to remove a budget.

#### Responses

HTTP Code	Description	Schema
204	Budget was removed.	No Content
409	ID has invalid format.	No Content

#### Example HTTP request

##### Request path

```
/budgets/{id}
```

## 2.3. Order

### 2.3.1. Create order.

POST /orders

#### Description

Method allow to create a new order.

#### Responses

HTTP Code	Description	Schema
201	Order was created.	<a href="#">ProductOrder</a>
409	Order quantity cannot be null or is infinite.	No Content

#### Example HTTP request

##### Request path

/orders

#### Example HTTP response

##### Response 201

```
{
  "id" : "string",
  "product" : {
    "id" : "string",
    "name" : "string",
    "price" : 0.0
  },
  "quantity" : 0
}
```

### 2.3.2. Retrieve all product orders.

GET /orders

#### Description

Method allow to find all product orders.

#### Responses

HTTP Code	Description	Schema
200	Orders were found.	< <a href="#">ProductOrder</a> > array
404	Orders are not found.	No Content

#### Example HTTP request

##### Request path

/orders

#### Example HTTP response

##### Response 200

```
[ {
  "id" : "string",
  "product" : {
    "id" : "string",
    "name" : "string",
    "price" : 0.0
  },
  "quantity" : 0
} ]
```

### 2.3.3. Update order.

PUT /orders

#### Description

Method allow to update an order.

#### Responses

HTTP Code	Description	Schema
200	Order was updated.	<a href="#">ProductOrder</a>
404	Order not found.	No Content
409	ID has invalid format.	No Content

#### Example HTTP request

##### Request path

/orders

#### Example HTTP response

##### Response 200

```
{
  "id" : "string",
  "product" : {
    "id" : "string",
    "name" : "string",
    "price" : 0.0
  },
  "quantity" : 0
}
```

### 2.3.4. Find Order by id.

```
GET /orders/{id}
```

#### Description

Method allow to find a order by id.

#### Responses

HTTP Code	Description	Schema
200	Order was found.	<a href="#">ProductOrder</a>
404	Order not found.	No Content
409	ID has invalid format.	No Content

#### Example HTTP request

##### Request path

```
/orders/{id}
```

#### Example HTTP response

##### Response 200

```
{
  "id" : "string",
  "product" : {
    "id" : "string",
    "name" : "string",
    "price" : 0.0
  },
  "quantity" : 0
}
```



### 2.3.5. Update order by id.

```
PUT /orders/{id}
```

#### Description

Method allow to update a order by id.

#### Responses

HTTP Code	Description	Schema
200	Order was updated.	<a href="#">ProductOrder</a>
404	Order not found.	No Content
409	ID has invalid format.	No Content

#### Example HTTP request

##### Request path

```
/orders/{id}
```

#### Example HTTP response

##### Response 200

```
{
  "id" : "string",
  "product" : {
    "id" : "string",
    "name" : "string",
    "price" : 0.0
  },
  "quantity" : 0
}
```

### 2.3.6. Remove order.

```
DELETE /orders/{id}
```

#### Description

Method allow to remove an order by id.

#### Responses

HTTP Code	Description	Schema
204	Order was removed.	No Content
409	ID has invalid format.	No Content

#### Example HTTP request

##### Request path

```
/orders/{id}
```

## 2.4. Product

### 2.4.1. Create product.

POST /products

#### Description

Method allow to create a new product.

#### Responses

HTTP Code	Description	Schema
201	Product was created.	<a href="#">Product</a>
409	Product price cannot be null or is infinite.	No Content

#### Example HTTP request

##### Request path

/products

#### Example HTTP response

##### Response 201

```
{
  "id" : "string",
  "name" : "string",
  "price" : 0.0
}
```

## 2.4.2. Retrieve all products.

GET /products

### Description

Method allow to find all products.

### Responses

HTTP Code	Description	Schema
200	Products were found.	< <a href="#">Product</a> > array
404	Products are not found.	No Content

### Example HTTP request

#### Request path

/products

### Example HTTP response

#### Response 200

```
[ {  
  "id" : "string",  
  "name" : "string",  
  "price" : 0.0  
} ]
```

### 2.4.3. Update product.

PUT /products

#### Description

Method allow to update a product.

#### Responses

HTTP Code	Description	Schema
200	Product was updated.	<a href="#">Product</a>
404	Product not found.	No Content
409	ID has invalid format.	No Content

#### Example HTTP request

##### Request path

/products

#### Example HTTP response

##### Response 200

```
{
  "id" : "string",
  "name" : "string",
  "price" : 0.0
}
```

## 2.4.4. Find product by id.

```
GET /products/{id}
```

### Description

Method allow to find a product by id.

### Responses

HTTP Code	Description	Schema
200	Product was found.	<a href="#">Product</a>
404	Product not found.	No Content
409	ID has invalid format.	No Content

### Example HTTP request

#### Request path

```
/products/{id}
```

### Example HTTP response

#### Response 200

```
{
  "id" : "string",
  "name" : "string",
  "price" : 0.0
}
```

## 2.4.5. Update Product by id.

```
PUT /products/{id}
```

### Description

Method allow to update a product by id.

### Responses

HTTP Code	Description	Schema
200	Product was updated.	<a href="#">Product</a>
404	Product not found.	No Content
409	ID has invalid format.	No Content

### Example HTTP request

#### Request path

```
/products/{id}
```

### Example HTTP response

#### Response 200

```
{
  "id" : "string",
  "name" : "string",
  "price" : 0.0
}
```

## 2.4.6. Remove product.

```
DELETE /products/{id}
```

### Description

Method allow to remove a product by id.

### Responses

HTTP Code	Description	Schema
204	Product was removed.	No Content
409	ID has invalid format.	No Content

### Example HTTP request

#### Request path

```
/products/{id}
```



# Chapter 3. Definitions

## 3.1. BillOfSale

Name	Description	Schema
<b>boughtDate</b> <i>optional</i>	Example : 0	integer (int64)
<b>description</b> <i>optional</i>	Example : "string"	string
<b>id</b> <i>required</i>	Example : "string"	string
<b>productList</b> <i>optional</i>	Example : [ "ProductOrder" ]	< ProductOrder > array

## 3.2. Budget

Name	Description	Schema
<b>billsOfSaleList</b> <i>optional</i>	Example : [ "BillOfSale" ]	< BillOfSale > array
<b>budgetValue</b> <i>optional</i>	Example : 0.0	number (double)
<b>id</b> <i>required</i>	Example : "string"	string
<b>name</b> <i>optional</i>	Example : "string"	string

## 3.3. Product

Name	Description	Schema
<b>id</b> <i>required</i>	Example : "string"	string
<b>name</b> <i>optional</i>	Example : "string"	string

Name	Description	Schema
<b>price</b> <i>optional</i>	Example : 0.0	number (double)

### 3.4. ProductOrder

Name	Description	Schema
<b>id</b> <i>required</i>	Example : "string"	string
<b>product</b> <i>optional</i>	Example : Product	Product
<b>quantity</b> <i>optional</i>	Example : 0	integer (int32)