

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. POST-endpoint to create BillOfSale	2
2.1.2. GET-endpoint to find BillOfSale by bought date	3
2.1.3. PUT-endpoint to update BillOfSale	4
2.1.4. GET-endpoint to find BillOfSale by id	5
2.1.5. PUT-endpoint to update BillOfSale of id	6
2.1.6. DELETE-endpoint to removeById BillOfSale	7
2.2. Budget	8
2.2.1. POST-endpoint to create Budget	8
2.2.2. GET-endpoint to find Budget by name	9
2.2.3. PUT-endpoint to update Budget	10
2.2.4. GET-endpoint to find Budget by id	11
2.2.5. PUT-endpoint to update Budget of id	12
2.2.6. DELETE-endpoint to removeById Budget	13
2.3. Order	14
2.3.1. POST-endpoint to create Order	14
2.3.2. GET-endpoint to find Order by name and price	15
2.3.3. PUT-endpoint to update Order	16
2.3.4. GET-endpoint to find Order by id	17
2.3.5. PUT-endpoint to update Order of id	18
2.3.6. DELETE-endpoint to removeById Order	19
2.4. Product	20
2.4.1. POST-endpoint to create Product	20
2.4.2. GET-endpoint to find Product by name	21
2.4.3. PUT-endpoint to update Product	22
2.4.4. GET-endpoint to find Product by id	23
2.4.5. PUT-endpoint to update Product of id	24
2.4.6. DELETE-endpoint to removeById Product	25
3. Definitions	26
3.1. BillOfSale	26
3.2. Budget	26
3.3. Product	26

3.4. ProductOrder.....	27
------------------------	----

Chapter 1. Overview

ExpensesManager is a service to expenses management.

1.1. Version information

Version : 0.3.5

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. POST-endpoint to create BillOfSale.

POST /billofsales

Description

Method allow to create a new bill of sale.

Responses

HTTP Code	Description	Schema
200	successful operation	BillOfSale

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"
}
```

2.1.2. GET-endpoint to find BillOfSale by bought date.

GET /billofsales

Description

Method allow to find a bill of sale by bought date.

Responses

HTTP Code	Description	Schema
200	successful operation	< BillOfSale > array

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. PUT-endpoint to update BillOfSale.

PUT /billofsales

Description

Method allow to update a bill of sale.

Responses

HTTP Code	Description	Schema
200	successful operation	BillOfSale

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. GET-endpoint to find BillOfSale by id.

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

Description

Method allow to find a bill of sale by id.

Responses

HTTP Code	Description	Schema
200	successful operation	BillOfSale

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. PUT-endpoint to update BillOfSale of id.

PUT /billofsales/{id}

Description

Method allow to update a bill of sale by id.

Responses

HTTP Code	Description	Schema
200	successful operation	BillOfSale

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. DELETE-endpoint to removeById BillOfSale.

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

Description

Method allow to removeById a bill of sale.

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Example HTTP request

Request path

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

2.2. Budget

2.2.1. POST-endpoint to create Budget.

POST /budgets

Description

Method allow to create a new budget.

Responses

HTTP Code	Description	Schema
200	successful operation	Budget

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"
  } ]
}
```

2.2.2. GET-endpoint to find Budget by name.

GET /budgets

Description

Method allow to find a budget by name.

Responses

HTTP Code	Description	Schema
200	successful operation	Budget

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. PUT-endpoint to update Budget.

PUT /budgets

Description

Method allow to update a budget.

Responses

HTTP Code	Description	Schema
200	successful operation	Budget

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"
  } ]
}
```

2.2.4. GET-endpoint to find Budget by id.

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

Description

Method allow to find a budget by id.

Responses

HTTP Code	Description	Schema
200	successful operation	Budget

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. PUT-endpoint to update Budget of id.

PUT /budgets/{id}

Description

Method allow to update a budget by id.

Responses

HTTP Code	Description	Schema
200	successful operation	Budget

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"
  } ]
}
```

2.2.6. DELETE-endpoint to removeById Budget.

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

Description

Method allow to removeById a budget.

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Example HTTP request

Request path

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


2.3. Order

2.3.1. POST-endpoint to create Order.

POST /orders

Description

Method allow to create a new order.

Responses

HTTP Code	Description	Schema
200	successful operation	ProductOrder

Example HTTP request

Request path

/orders

Example HTTP response

Response 200

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

2.3.2. GET-endpoint to find Order by name and price.

GET /orders

Description

Method allow to find a order by name and price.

Responses

HTTP Code	Description	Schema
200	successful operation	ProductOrder

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. PUT-endpoint to update Order.

PUT /orders

Description

Method allow to update a order.

Responses

HTTP Code	Description	Schema
200	successful operation	ProductOrder

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. GET-endpoint to find Order by id.

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

Description

Method allow to find a order by id.

Responses

HTTP Code	Description	Schema
200	successful operation	ProductOrder

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. PUT-endpoint to update Order of id.

PUT /orders/{id}

Description

Method allow to update a order by id.

Responses

HTTP Code	Description	Schema
200	successful operation	ProductOrder

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. DELETE-endpoint to removeById Order.

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

Description

Method allow to removeById a order.

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Example HTTP request

Request path

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

2.4. Product

2.4.1. POST-endpoint to create Product.

POST /products

Description

Method allow to create a new product.

Responses

HTTP Code	Description	Schema
200	successful operation	Product

Example HTTP request

Request path

/products

Example HTTP response

Response 200

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

2.4.2. GET-endpoint to find Product by name.

GET /products

Description

Method allow to find a product by name.

Responses

HTTP Code	Description	Schema
200	successful operation	< Product > array

Example HTTP request

Request path

/products

Example HTTP response

Response 200

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


2.4.3. PUT-endpoint to update Product.

PUT /products

Description

Method allow to update a product.

Responses

HTTP Code	Description	Schema
200	successful operation	Product

Example HTTP request

Request path

/products

Example HTTP response

Response 200

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

2.4.4. GET-endpoint to find Product by id.

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

Description

Method allow to find a product by id.

Responses

HTTP Code	Description	Schema
200	successful operation	Product

Example HTTP request

Request path

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

Example HTTP response

Response 200

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

2.4.5. PUT-endpoint to update Product of id.

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

Description

Method allow to update a product by id.

Responses

HTTP Code	Description	Schema
200	successful operation	Product

Example HTTP request

Request path

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

Example HTTP response

Response 200

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

2.4.6. DELETE-endpoint to removeById Product.

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

Description

Method allow to removeById a product.

Responses

HTTP Code	Description	Schema
default	successful operation	No Content

Example HTTP request

Request path

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

Chapter 3. Definitions

3.1. BillOfSale

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

3.2. Budget

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

3.3. Product

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

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

3.4. ProductOrder

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