

API Specification Doc

(OrderM8)

Version	Date	Author	Description
1.0	29.10.2016	Thomas Leiter	CRUD-Functionality

INDEX

1	Producttype.....	3
1.1	Get Producttypes.....	3
1.2	Get Producttype	3
1.3	Add Producttype.....	4
1.4	Update Producttype	5
1.5	Delete Producttype.....	5
2	Bill.....	6
2.1	Get Bills.....	6
2.2	Get Bill	7
2.3	Add Bill.....	7
2.4	Delete Bill.....	8
3	Product.....	9
3.1	Get Products	9
3.2	Get Product.....	9
3.3	Add Product	10
3.4	Delete Product.....	11
4	Table.....	11
4.1	Get Tables	12
4.2	Get Table	12
4.3	Add Table.....	13
4.4	Delete Table.....	14
5	User	14
5.1	Get Users	14
5.2	Get User	15
5.3	Add User	16
5.4	Delete User	17
6	Orderentry	17
6.1	Get Orderentries.....	17
6.2	Get Orderentry	18
6.3	Add Orderentry.....	19
6.4	Delete Orderentry	20
6.5	Conventions.....	21
6.6	Status Codes	21

Conventions

Status Codes

METHODS

1 PRODUCTTYPE

administrate producttypes

1.1 GET PRODUCTTYPES

1.1.1 REQUEST

Method	URL
GET	api/producttype/

1.1.2 RESPONSE

Status	Response
200	<p>Response will be an object containing the list of producttype (array)</p> <p>An example response is:</p> <pre>[{ "idType": 3, "name": "Drink", "vat": 20 }, { "idType": 4, "name": "Food", "vat": 10 }]</pre>
500	<pre>{ "errorCode": 500, "errorMessage": "{error_msg}" }</pre>

1.2 GET PRODUCTTYPE

1.2.1 REQUEST

Method	URL
GET	api/producttype/:id

Type	Params	Values
URL_PARAM	<id>	number

1.2.2 RESPONSE

Status	Response
200	Response will be an object containing the producttype An example response is: <pre>{ "idType": 3, "name": "Drink", "vat": 20 }</pre>
404	An example response is: <pre>{ "errorCode": 404, "errorMessage": "ProductType with id 2 not found" }</pre>
500	<pre>{ "errorCode": 500, "errorMessage": "{error_msg}" }</pre>

1.3 ADD PRODUCTTYPE

1.3.1 REQUEST

Method	URL
POST	api/producttype/

Type	Params	Values
POST	raw	string An example request is: <pre>{ "name": "Food", "vat": 10 }</pre>

1.3.2 RESPONSE

Status	Response
201	Response will be the created Producttype An example response is: <pre>{ "idType": 1, "name": "Food",</pre>

	<pre>"vat": 10 }</pre>
409	<pre>{ "errorCode": 409, "errorMessage": "conflict with another resource" }</pre>
500	<pre>{ "errorCode": 500, "errorMessage": "{error_msg}" }</pre>

1.4 UPDATE PRODUCTTYPE

1.4.1 REQUEST

Method	URL
PUT	api/producttype/

Type	Params	Values
URL_PARAM	<id>	number
POST	raw	string An example request is: <pre>{ "name": "Food_Changed", "vat": 15 }</pre>

1.4.2 RESPONSE

Status	Response
204	-
404	An example response is: <pre>{ "errorCode": 404, "errorMessage": "ProductType with id 2 not found" }</pre>
500	<pre>{ "errorCode": 500, "errorMessage": "{error_msg}" }</pre>

1.5 DELETE PRODUCTTYPE

1.5.1 REQUEST

Method	URL
DELETE	api/producttype/

Type	Params	Values
URL_PARAM	<id>	number

1.5.2 RESPONSE

Status	Response
204	-
404	An example response is: { "errorCode": 404, "errorMessage": "ProductType with id 2 not found" }
500	{ "errorCode": 500, "errorMessage": "{error_msg}" }

2 BILL

administrate bills

2.1 GET BILLS

2.1.1 REQUEST

Method	URL
GET	api/bill/

2.1.2 RESPONSE

Status	Response
200	Response will be an object containing the list of bill (array) An example response is: [{ "billDate": "2016-10-29T13:53:47.183+02:00", "idBill": 3 },]

	<pre>{ "billDate": "2016-10-29T13:56:43.653+02:00", "idBill": 4 }</pre>
500	<pre>{ "errorCode": 500, "errorMessage": "{error_msg}" }</pre>

2.2 GET BILL

2.2.1 REQUEST

Method	URL
GET	api/bill/:id

Type	Params	Values
URL_PARAM	<id>	number

2.2.2 RESPONSE

Status	Response
200	<p>Response will be an object containing the bill</p> <p>An example response is:</p> <pre>{ "billDate": "2016-10-29T13:56:43.653+02:00", "idBill": 4 }</pre>
404	<p>An example response is:</p> <pre>{ "errorCode": 404, "errorMessage": "Bill with id 2 not found" }</pre>
500	<pre>{ "errorCode": 500, "errorMessage": "{error_msg}" }</pre>

2.3 ADD BILL

2.3.1 REQUEST

Method	URL
POST	api/Bill/

2.3.2 RESPONSE

Status	Response
201	Response will be the created bill An example response is: <pre>{ "idBill": 28 }</pre>
500	<pre>{ "errorCode": 500, "errorMessage": "{error_msg}" }</pre>

2.4 DELETE BILL

2.4.1 REQUEST

Method	URL
DELETE	api/Bill/

Type	Params	Values
URL_PARAM	<id>	number

2.4.2 RESPONSE

Status	Response
204	-
404	An example response is: <pre>{ "errorCode": 404, "errorMessage": "Bill with id 2 not found" }</pre>
500	<pre>{ "errorCode": 500, "errorMessage": "{error_msg}" }</pre>

3 PRODUCT

administrate products

3.1 GET PRODUCTS

3.1.1 REQUEST

Method	URL
GET	api/product/

3.1.2 RESPONSE

Status	Response
200	<p>Response will be an object containing the list of product (array)</p> <p>An example response is:</p> <pre>[{ "fkType": 1, "idProduct": 6, "name": "schopf", "price": 13, "quantity": 200 }, { "fkType": 1, "idProduct": 5, "name": "schnitzel", "price": 10.2, "quantity": 20 }]</pre>
500	<pre>{ "errorCode": 500, "errorMessage": "{error_msg}" }</pre>

3.2 GET PRODUCT

3.2.1 REQUEST

Method	URL
GET	api/Product/:id

Type	Params	Values
URL_PARAM	<id>	number

3.2.2 RESPONSE

Status	Response
200	Response will be an object containing the Product An example response is: <pre>{ "fkType": 1, "idProduct": 5, "name": "schnitzel", "price": 10.2, "quantity": 20 }</pre>
404	An example response is: <pre>{ "errorCode": 404, "errorMessage": "Product with id 2 not found" }</pre>
500	<pre>{ "errorCode": 500, "errorMessage": "{error_msg}" }</pre>

3.3 ADD PRODUCT

3.3.1 REQUEST

Method	URL
POST	api/product/

Type	Params	Values
POST	raw	string An example request is: <pre>{ "fkType": 1, "name": "schnitzel", "price": 10.2, "quantity": 20 }</pre>

3.3.2 RESPONSE

Status	Response
201	Response will be the created product An example response is:

	<pre>{ "fkType": 1, "idProduct": 5, "name": "schnitzel", "price": 10.2, "quantity": 20 }</pre>
409	<pre>{ "errorCode": 409, "errorMessage": "conflict with another resource" }</pre>
500	<pre>{ "errorCode": 500, "errorMessage": "{error_msg}" }</pre>

3.4 DELETE PRODUCT

3.4.1 REQUEST

Method	URL
DELETE	api/product/

Type	Params	Values
URL_PARAM	<id>	number

3.4.2 RESPONSE

Status	Response
204	-
404	<p>An example response is:</p> <pre>{ "errorCode": 404, "errorMessage": "Product with id 2 not found" }</pre>
500	<pre>{ "errorCode": 500, "errorMessage": "{error_msg}" }</pre>

4 TABLE

administrate tables

4.1 GET TABLES

4.1.1 REQUEST

Method	URL
GET	api/table/

4.1.2 RESPONSE

Status	Response
200	<p>Response will be an object containing the list of table (array)</p> <p>An example response is:</p> <pre>[{ "idTable": 9, "x": 3, "y": 2 }, { "idTable": 10, "x": 3, "y": 2 }, { "idTable": 11, "x": 3, "y": 5 }]</pre>
500	<pre>{ "errorCode": 500, "errorMessage": "{error_msg}" }</pre>

4.2 GET TABLE

4.2.1 REQUEST

Method	URL
GET	api/table/:id

Type	Params	Values
URL_PARAM	<id>	number

4.2.2 RESPONSE

Status	Response
200	<p>Response will be an object containing the table</p> <p>An example response is:</p> <pre>{ "idTable": 10, "x": 3, "y": 2 }</pre>
404	<p>An example response is:</p> <pre>{ "errorCode": 404, "errorMessage": "Table with id 2 not found" }</pre>
500	<pre>{ "errorCode": 500, "errorMessage": "{error_msg}" }</pre>

4.3 ADD TABLE

4.3.1 REQUEST

Method	URL
POST	api/table/

Type	Params	Values
POST	raw	<pre>string</pre> <p>An example request is:</p> <pre>{ "x":10, "y":5 }</pre>

4.3.2 RESPONSE

Status	Response
201	<p>Response will be the created table</p> <p>An example response is:</p> <pre>{ "idTable": 10, "x": 3, "y": 2 }</pre>

409	{ "errorCode": 409, "errorMessage": "conflict with another resource" }
500	{ "errorCode": 500, "errorMessage": "{error_msg}" }

4.4 DELETE TABLE

4.4.1 REQUEST

Method	URL
DELETE	api/table/

Type	Params	Values
URL_PARAM	<id>	number

4.4.2 RESPONSE

Status	Response
204	-
404	An example response is: { "errorCode": 404, "errorMessage": "Table with id 2 not found" }
500	{ "errorCode": 500, "errorMessage": "{error_msg}" }

5 USER

administrate users

5.1 GET USERS

5.1.1 REQUEST

Method	URL
GET	api/user/

5.1.2 RESPONSE

Status	Response
200	<p>Response will be an object containing the list of user (array)</p> <p>An example response is:</p> <pre>[{ "idUser": 2, "password": "test", "uType": 1, "username": "tomi" }, { "idUser": 3, "password": "evil", "uType": 2, "username": "andi" }]</pre>
500	<pre>{ "errorCode": 500, "errorMessage": "{error_msg}" }</pre>

5.2 GET USER

5.2.1 REQUEST

Method	URL
GET	api/user/:id

Type	Params	Values
URL_PARAM	<id>	number

5.2.2 RESPONSE

Status	Response
200	<p>Response will be an object containing the user</p> <p>An example response is:</p> <pre>{ "idUser": 3, "password": "evil", "uType": 2, "username": "andi" }</pre>

404	An example response is: <pre>{ "errorCode": 404, "errorMessage": "User with id 2 not found" }</pre>
500	<pre>{ "errorCode": 500, "errorMessage": "{error_msg}" }</pre>

5.3 ADD USER

5.3.1 REQUEST

Method	URL
POST	api/user/

Type	Params	Values
POST	raw	<p>string</p> <p>An example request is:</p> <pre>{ "username": "tomi", "password": "test", "uType": 1 }</pre> <p><i>uType: 1=Waiter;2=Organisers</i></p>

5.3.2 RESPONSE

Status	Response
201	<p>Response will be the created user</p> <p>An example response is:</p> <pre>{ "idUser": 2, "password": "test", "uType": 1, "username": "tomi" }</pre>
409	<pre>{ "errorCode": 409, "errorMessage": "conflict with another resource" }</pre>
500	<pre>{</pre>

	<pre> "errorCode": 500, "errorMessage": "{error_msg}" } </pre>
--	--

5.4 DELETE USER

5.4.1 REQUEST

Method	URL
DELETE	api/user/

Type	Params	Values
URL_PARAM	<id>	number

5.4.2 RESPONSE

Status	Response
204	-
404	An example response is: <pre> { "errorCode": 404, "errorMessage": "User with id 2 not found" } </pre>
500	<pre> { "errorCode": 500, "errorMessage": "{error_msg}" } </pre>

6 ORDERENTRY

administrate orderentries

6.1 GET ORDERENTRIES

6.1.1 REQUEST

Method	URL
GET	api/orderentry/

6.1.2 RESPONSE

Status	Response
200	Response will be an object containing the list of orderentry (array)

	<p>An example response is:</p> <pre>[{ "cancelled": false, "coupon": false, "fkBill": 0, "fkProduct": 5, "fkTable": 10, "fkUser": 2, "idOrderEntry": 10, "note": "ohne ketchup" }, { "cancelled": false, "coupon": false, "fkBill": 0, "fkProduct": 6, "fkTable": 11, "fkUser": 2, "idOrderEntry": 11 }]</pre>
500	<pre>{ "errorCode": 500, "errorMessage": "{error_msg}" }</pre>

6.2 GET ORDERENTRY

6.2.1 REQUEST

Method	URL
GET	api/orderentry/:id

Type	Params	Values
URL_PARAM	<id>	number

6.2.2 RESPONSE

Status	Response
200	<p>Response will be an object containing the orderentry</p> <p>An example response is:</p> <pre>{ "cancelled": false, "coupon": false, "fkBill": 0,</pre>

	<pre> "fkProduct": 5, "fkTable": 10, "fkUser": 2, "idOrderEntry": 10, "note": "ohne ketchup" } </pre>
404	An example response is: <pre> { "errorCode": 404, "errorMessage": "Orderentry with id 2 not found" } </pre>
500	<pre> { "errorCode": 500, "errorMessage": "{error_msg}" } </pre>

6.3 ADD ORDERENTRY

6.3.1 REQUEST

Method	URL
POST	api/orderentry/

Type	Params	Values
POST	raw	<pre> string An example request is: { "fkProduct":5, "fkTable":10, "fkUser":2, "note":"ohne ketchup", "cancelled":0, "coupon":0 } </pre>

6.3.2 RESPONSE

Status	Response
201	Response will be the created orderentry <p>An example response is:</p> <pre> { "cancelled": false, "coupon": false, "fkBill": 0, </pre>

	<pre> "fkProduct": 5, "fkTable": 10, "fkUser": 2, "idOrderEntry": 10, "note": "ohne ketchup" } </pre>
500	<pre> { "errorCode": 500, "errorMessage": "{error_msg}" } </pre>

6.4 DELETE ORDERENTRY

6.4.1 REQUEST

Method	URL
DELETE	api/orderentry/

Type	Params	Values
URL_PARAM	<id>	number

6.4.2 RESPONSE

Status	Response
204	-
404	<p>An example response is:</p> <pre> { "errorCode": 404, "errorMessage": "Orderentry with id 2 not found" } </pre>
500	<pre> { "errorCode": 500, "errorMessage": "{error_msg}" } </pre>

GLOSSARY

6.5 CONVENTIONS

- **Client** - Client application.
- **Status** - HTTP status code of response.
- All the possible responses are listed under 'Responses' for each method. Only one of them is issued per request server.
- All response are in JSON format.

6.6 STATUS CODES

All status codes are standard HTTP status codes. The below ones are used in this API.

2XX - Success of some kind

4XX - Error occurred in client's part

5XX - Error occurred in server's part

Status Code	Description
200	OK
201	Created
409	Conflict
503	Service Unavailable