

Documentation for REST API for the Point-of-Sale/Customer Care system

This document intends to describe the REST API serving data/resources for the Point-of-Sale/Customer Care system.

All existing end-points and the methods they support are described along with which data is expected as input.

The URL:s described below are accessible on: <http://127.0.0.1:6399/>

('/', methods=['GET'])

Description:

Root node, expects no input and returns a greeting.

Input:

None

Result:

String: "Welcome to the REST API for the Point-of-Sale/Customer Care system."

('/full_customer/{int:customer_id}', methods=['GET'])

Description:

Get a representation the "complete" customer. Ie all the customer, equipment/product, sim data. All collected in one JSON-object.

Input:

None:

Result:

```
{
  "Age": 39,
  "City": "Lyckeby",
  "Email": "kristoffer.nordstrom@northerntest.se",
  "Firstname": "Kristoffer",
  "ID": 340,
  "IMEIPtr": 101,
  "IMSIPtr": 439,
  "Lastname": "Nordstrom",
  "Nationality": "Swedish",
  "Password": "SuperSecret",
  "Sex": "Male",
  "Street": "Ronnvagen 5",
  "SubscriptionPtr": "",
  "Zip": "37160",
  "equipment": {
    "ID": 101,
    "IMEI": "IMEI_0123456789",
    "ProductPtr": 452,
    "product": {
```

```
        "ID": 452,
        "ImageURL": "/products/images/452.jpeg",
        "Model": "Google Pixel 2 XL",
        "Type": "Phone"
    },
    "sim": {
        "ID": 439,
        "IMSI": "IMSI_0123456789",
        "MSISDN": "+46723580953"
    }
}
```

Customers

`'/customers/{int:customer_id}', methods=['GET']`

Description:

Retrieves the customer with the specified `customer_id`.

Input:

None

Result: Customer

```
{
  "Age": 39,
  "City": "Lyckeby",
  "Email": "kristoffer.nordstrom@northerntest.se",
  "Firstname": "Kristoffer",
  "ID": 340,
  "IMEIPtr": 101,
  "IMSIPtr": 439,
  "Lastname": "Nordstrom",
  "Nationality": "Swedish",
  "Password": "SuperSecret",
  "Sex": "Male",
  "Street": "Ronnvagen 5",
  "SubscriptionPtr": "",
  "Zip": "37160"
}
```

`'/customers/{int:customer_id}', methods=['DELETE']`

Description:

Deletes the customer with the specified `customer_id`.

Input:

None

Result: String:Message

“Failed to find customer ID”, `status.HTTP_404_NOT_FOUND`
“Internal DB error, found multiple customers with same ID”,
`status.HTTP_500_INTERNAL_SERVER_ERROR`

"OK", status.HTTP_200_OK

`‘/customers/{int:customer_id}’, methods=[‘PUT’]`

Description:

Updates a customer with the specified values. Returns the newly updated customer object.

Input:

```
{
  'Firstname': <String>, (Required)
  'Lastname': <String>, (Required)
  'Age': <String>, (Required)
  'Sex': <String [Male|Female]>, (Required)
  'Street': <String>, (Required)
  'Zip': <String>, (Required)
  'City': <String>, (Required)
  'Nationality': <String>,
  'IMSIPtr': <String>, #ID To a sim belonging to the customer
  'IMEIPtr': <String>, #ID To a equipment belonging to the customer
  'Email': <String>,
  'Password': <String>
}
```

Result:

```
{
  "Age": 15,
  "City": "Torhamn",
  "Email": None,
  "Firstname": "Kristoffer",
  "ID": 341,
  "IMEIPtr": None,
  "IMSIPtr": None,
  "Lastname": "Nordstrom",
  "Nationality": "Swedish",
  "Password": "LessSecret",
  "Sex": "Male",
  "Street": "Gisslev
  "SubscriptionPtr": None,
  "Zip": "37042"
}
```

`‘/customers’, methods=[‘POST’]`

Description:

Creates a new customer in the system.

Input:

```
{'Firstname': <String>, (Required)
  'Lastname': <String>, (Required)
  'Age': <String>, (Required)
  'Sex': <String [Male|Female]>, (Required)
  'Street': <String>, (Required)
  'Zip': <String>, (Required)
  'City': <String>, (Required)
  'Nationality': <String>,
  'IMSIPtr': <String>, #ID To a sim belonging to the customer
  'IMEIPtr': <String>, #ID To a equipment belonging to the customer
```

```
'Email': <String>,  
'Password': <String>  
}
```

Result: Returns back the newly created customer with its ID

```
{  
  "Age": 39,  
  "City": "Lyckeby",  
  "Email": "kristoffer.nordstrom@northerntest.se",  
  "Firstname": "Kristoffer",  
  "ID": 340,  
  "IMEIPtr": 101,  
  "IMSIPtr": 439,  
  "Lastname": "Nordstrom",  
  "Nationality": "Swedish",  
  "Password": "SuperSecret",  
  "Sex": "Male",  
  "Street": "Ronnvagen 5",  
  "SubscriptionPtr": "",  
  "Zip": "37160"  
}
```

‘/customers’, methods=[‘GET’]

Description:

Retrieves all the customers in the system. Only includes customer information, and not equipment/products, sim.

Input:

None

Result:

```
[  
  {  
    "Age": 39,  
    "City": "Lyckeby",  
    "Email": "kristoffer.nordstrom@northerntest.se",  
    "Firstname": "Kristoffer",  
    "ID": 340,  
    "IMEIPtr": 101,  
    "IMSIPtr": 439,  
    "Lastname": "Nordstrom",  
    "Nationality": "Swedish",  
    "Password": "SuperSecret",  
    "Sex": "Male",  
    "Street": "Ronnvagen 5",  
    "SubscriptionPtr": "",  
    "Zip": "37160"  
  },  
  {  
    "Age": 15,  
    "City": "Torhamn",  
    "Email": None,  
    "Firstname": "Kristoffer",  
    "ID": 341,  
    "IMEIPtr": None,  
    "IMSIPtr": None,  
    "Lastname": "Nordstrom",  
    "Nationality": Swedish,  
    "Password": "SuperSecret",  
    "Sex": "Male",  
    "Street": "Ronnvagen 5",  
    "SubscriptionPtr": "",  
    "Zip": "37160"  
  }  
]
```

```
        "Password": "LessSecret",
        "Sex": "Male",
        "Street": "Gisslevik",
        "SubscriptionPtr": None,
        "Zip": "37042"
    }
]
```

SIM Cards

`‘/sims’, methods=[‘GET’]`

Description:

Retrieves all the sim cards in the system.

Input:

None

Result:

```
[
  {
    "ID": 439,
    "IMSI": "IMSI_0123456789",
    "MSISDN": "+46723580953"
  },
  {
    "ID": 440,
    "IMSI": "IMSI_9876543210",
    "MSISDN": "+46723580954"
  },
]
```

`‘/sims’, methods=[‘POST’]`

Description:

Creates a new sim card in the system. Returns the newly created sim with the ID.

Input:

```
indata = {
    "IMSI": "IMSI_0123456789",
    "MSISDN": "+46723580953"
}
```

Result:

```
{
  "ID": 439,
  "IMSI": "IMSI_0123456789",
  "MSISDN": "+46723580953"
}
```

`‘/sims/{int:sim_id}’, methods=[‘PUT’]`

Description:

Updates an existing SIM card with new details. Returns the updated SIM card.

Input:

```
indata = {  
    "IMSI": "NEW_IMSI_336699",  
    "MSISDN": "+46708662212"  
}
```

Result:

```
{  
    "ID": 439,  
    "IMSI": "NEW_IMSI_336699",  
    "MSISDN": "+46708662212"  
}
```

`‘/sims/{int:sim_id}’, methods=[‘GET’]`

Description:

Retrieves the SIM card with the specified ID.

Input:

None

Result:

```
{  
    "ID": 439,  
    "IMSI": "NEW_IMSI_336699",  
    "MSISDN": "+46708662212"  
}
```

`‘/sims/{int:sim_id}’, methods=[‘DELETE’]`

Description:

Deletes the SIM card with the specified ID.

Input:

None

Result: String:Message

“Failed to find sim ID”, status.HTTP_404_NOT_FOUND
“Internal DB error, found multiple sims with same ID”,
status.HTTP_500_INTERNAL_SERVER_ERROR
“OK”

Products

`‘/products’, methods=[‘GET’]`

Description:

Retrieves all the products in the system. Contains URL to the image for the product.

Input:

None

Result:

```
[
  {
    "ID": 452,
    "ImageURL": "/products/images/452.jpeg",
    "Model": "Google Pixel 2 XL",
    "Type": "Phone"
  },
  {
    "ID": 453,
    "ImageURL": "/products/images/453.jpeg",
    "Model": "Apple iPad Pro",
    "Type": "Tablet"
  }
]
```

`‘/products/{int:product_id}’, methods=[‘GET’]`

Description:

Retrieves the products with the specified ID in the system. Contains URL to the image for the product.

Input:

None

Result:

```
{
  "ID": 452,
  "ImageURL": "/products/images/452.jpeg",
  "Model": "Google Pixel 2 XL",
  "Type": "Phone"
}
```

`‘/products/images/{string:image_url}’, methods=[‘GET’]`

Description:

Retrieves the specified image in binary format.

Input:

None

Result:

Binary image data ready to store directly to disc or display.

‘/products’, methods=[‘POST’]

Description:

Creates a new product in the system, which has to be referenced by an equipment. Returns the newly created product with the ID. Notice that the image is uploaded separately, and that the imageURL is generated automatically.

Input:

```
indata = {
    'Type': 'Phone',
    'Model': 'Apple Iphone X',
}
```

Result:

```
{
    "ID": 453,
    "ImageURL": "",
    "Model": "Apple Iphone X",
    "Type": "Phone"
}
```

‘/products/{int:product_id}’, methods=[‘PUT’]

Description:

Updates a product with the parameters supplied. Returns the updated product.

Input:

```
indata = {
    'Type': 'Tablet',
    'Model': 'Apple iPad Pro',
}
```

Result:

```
{
    "ID": 453,
    "ImageURL": "",
    "Model": "Apple iPad Pro",
    "Type": "Tablet"
}
```

‘/products/{int:product_id}/image’, methods=[‘PUT’]

Description:

Uploads an image into the system for the product ID specified. Returns the updates product object with ImageURL set.

Input:

Binary image data.

Result:


```
{
  "ID": 453,
  "ImageURL": "/products/images/453.jpeg",
  "Model": "Apple iPad Pro",
  "Type": "Tablet"
}
```

`‘/products/{int:product_id}’, methods=[‘DELETE’]`

Description:

Deletes the product with the specified ID.

Input:

None

Result: String

“Failed to find product ID”, status.HTTP_404_NOT_FOUND
“Internal DB error, found multiple products with same ID”,
status.HTTP_500_INTERNAL_SERVER_ERROR
“OK”, status.HTTP_200_OK

`‘/equipments’, methods=[‘GET’]`

Description:

Retrieves all the equipment (Phones, Tablets...) in the system.

Input:

None

Result:

```
[
  {
    "ID": 101,
    "IMEI": "IMEI_0123456789",
    "ProductPtr": 452
  },
  {
    "ID": 102,
    "IMEI": "IMEI_987654321",
    "ProductPtr": 452
  }
]
```

Equipment

`‘/equipments/{int:equipment_id}’, methods=[‘GET’]`

Description:

Retrieves the equipment with the specified ID (Phones, Tablets...) in the system.

Input:

None

Result: Equipment

```
{
  "ID": 101,
  "IMEI": "IMEI_0123456789",
  "ProductPtr": 452
}
```

`‘/equipments’, methods=[‘POST’]`

Description:

Creates a new equipment. Returns back the newly created equipment with ID attached.

Input:

```
indata = {
  'IMEI': 'IMEI_336699',
  'ProductPtr': 453
}
```

Result:

```
{
  'ID': 103
  'IMEI': 'IMEI_336699',
  'ProductPtr': 453
}
```

`‘/equipments/{int:equipment_id}’, methods=[‘PUT’]`

Description:

Updates an existing equipment with the specified ID. Returns back the updated equipment.

Input:

```
indata = {
  'IMEI': 'IMEI_996633',
  'ProductPtr': 452
}
```

Result:

```
{
  'ID': 103
  'IMEI': 'IMEI_996633',
  'ProductPtr': 452
}
```

`‘/equipments/{int:equipment_id}’, methods=[‘DELETE’]`

Description:

Deletes the product with the specified ID.

Input:

None

Result: String

"Failed to find equipment ID", status.HTTP_404_NOT_FOUND
"Internal DB error, found multiple equipments with same ID",
status.HTTP_500_INTERNAL_SERVER_ERROR
"OK", HTTP_200_OK