CUSTOMER API INTERFACE GUIDE
IOT EASY CONNECT

T-MOBILE ONDERNEMEN
Netherlands

Version: 1.0

TABLE OF CONTENTS

INTRODUCTION

The Customer Web Service API is an interface to integrate the IoT Easy Connect portal functionality with your IT backend applications. You can monitor and administrate your SIM cards over the internet.
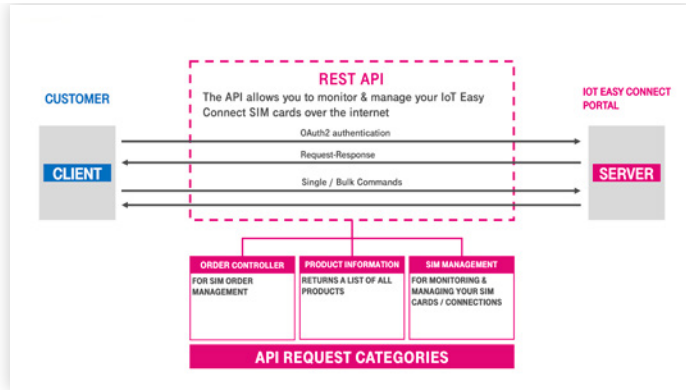

Figure 1: Customer API for monitoring and management

The Web Service API is REST-based (Representational State Transfer with JSON (JavaScript Object Notation) payload).

REST is a very simple and secure protocol based on HTTP(S). It is a standard protocol supported by all net-worked programming environments ranging from very simple IT-backend applications to large-scale IT.

This guide explains how to use the Customer-API Interface.

1. USING THE CUSTOMER API-INTERFACE

1.1 API Overview

A convenient way to explore the REST interface is to follow the link to the API:
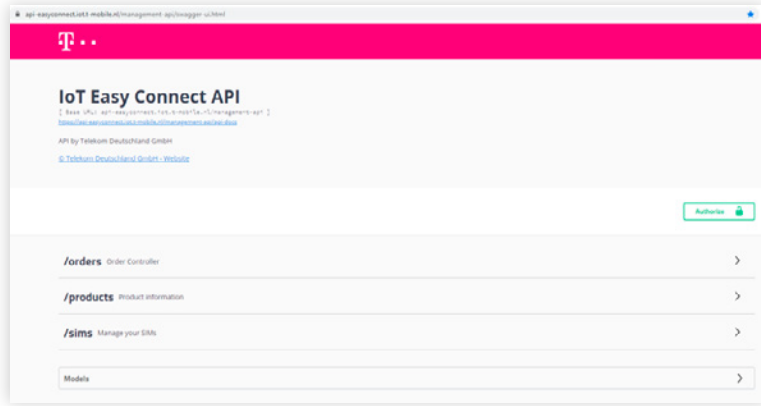
api-easyconnect.iot.t-mobile.nl

Figure 2: Customer API

You will see an overview of the available API requests, such as for ordering SIM cards, retrieving product information, and managing your SIM cards.

A) Explore API Requests
The arrow behind the text depicts an expandable content section. Click on the section header to expand or collapse the section. In the expanded state, all available API-requests are shown in a table.
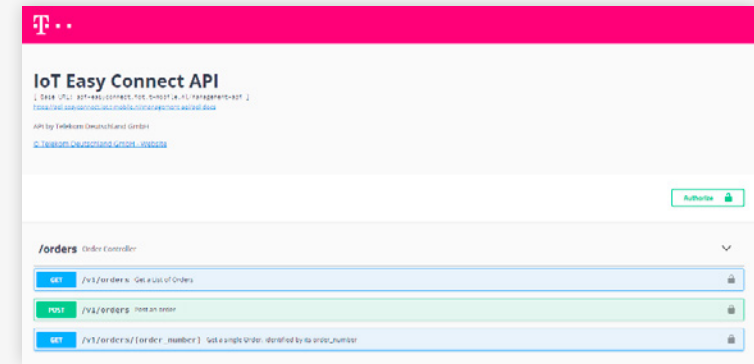

Figure 3: Example Action page using "Order Controller"

By clicking on each request, you will find an expanded section that details how to use each request and the parameter (see section 2.3. for an overview of all available API requests).

B) API test tool login-in
When you have authenticated yourself, you can try-out API requests with the test tool. The test tool is connected to the IoT Easy Connect Portal and allows you to get familiar with each API-request, the parameter and you can send requests for test purposes.

Login and API access are only possible for users with the role "API-Access". Authenticate yourself by enter-ing your username and password used to access the Portal GUI, in the login form, followed by a click on "Authorize".
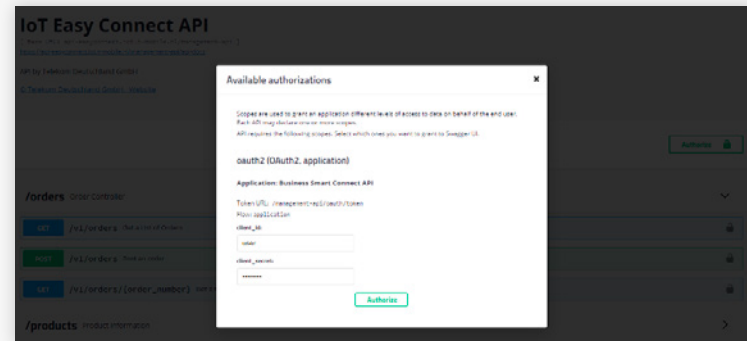

Figure 4: Test tool login window

Should the username or password be invalid, an error message "Auth Error" will appear and the opportunity to re-enter the information will be provided.

Upon successful login you will see the message "authorized" in the login-window. Next, click the exit symbol and you can start testing API requests.
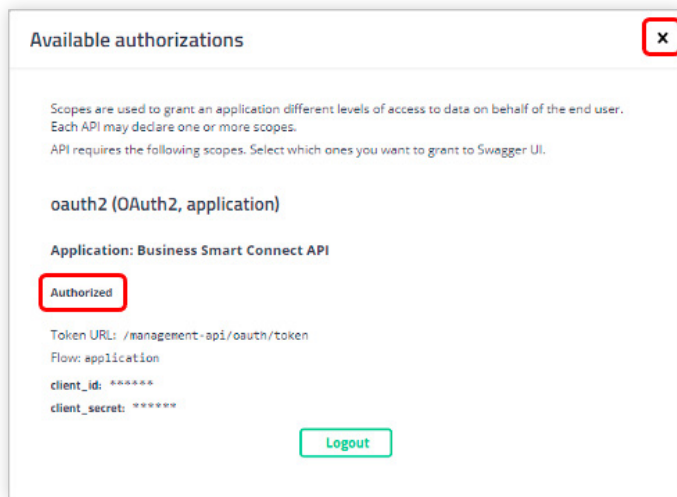

Figure 4: Checking your API authorization status

1.2 Available requests within the web service API

This chapter gives an overview of the API requests. Details and examples for these requests are outlined in chapter 3 "API Documentation".

There are four types of API-methods, as shown below:

GET	Retrieves information from the BSC-platform
POST	Adds information to the BSC-platform or SIM card
PUT	Modifies SIM card attributes
DELETE	Deletes a SMS that has not yet been delivered by the SMSC

Web services are synchronous. The client immediately receives a response or a time-out, in case something went wrong.

Some requests provide bulk support, which means it is possible to apply the action to more than one SIM card.

ORDER CONTROLLER

Request name	Bulk	Description
GET/v1/orders	No	Get a list of orders
POST/v1/orders	No	Post an order
GET/v1/orders/{order_number}	No	Get a single order, identified by its order_number

PRODUCT INFORMATION

Request name	Bulk	Description
GET/v1/products	No	Get a list of products

SIM MANAGEMENT

Request name	Bulk	Description
GET/v1/sims	No	Get a list of SIM cards
POST/v1/sims	No	Change a list of SIM cards for activate, deactivate, label, IMEI lock, etc.
GET/v1/sims/{iccid}	No	Get status information i.e. status, label, MSISDN, IMSI, ICCID, Lifetime, for a device with an ICCID
PUT/v1/sims/{iccid}	No	Modification of a SIM card to activate, deactivate, change label, change IMEI lock, etc.
GET/v1/sims/{iccid}/connectivity_info	No	Retrieve connectivity information and location for a SIM card
GET/v1/sims/{iccid}/events	No	Get diagnostic/event information for a SIM card i.e. net-work events
GET/v1/sims/{iccid}/quota/data	No	Get the data quota of a SIM card

Request name	Bulk	Description
GET/v1/sims/{ic-cid}/quota/sms	No	Get the data quota of a SIM card
POST/v1/sims/{iccid}/reset	No	Trigger a connectivity reset for a given SIM card. A positive response only means that the connectivity reset has been successfully placed into the queue.
GET/v1/sims/{iccid}/sms	No	Get the list of SMSs sent and received by this SIM card
POST/v1/sims/{iccid}/sms	No	Submit SMS to specific SIM card
GET/v1/sims/{iccid}/sms/{id}	No	Get details about the SMS
DELETE/v1/sims/{iccid}/sms/{id}	No	Cancel SMS that is buffered for SIM card and not yet delivered
POST/v1/sims/{iccid}/topup	No	Top up the data volume of a specific SIM card
GET/v1/sims/{iccid}/usage	No	Get the daily usage generated by a SIM card
POST/v1/sims/topup	Yes	Top up the data volume of specific SIM cards

You can find detailed information about how to use each request by following the link to the API **api-easyconnect.iot.t-mobile.nl**.

Login and collapse the section for the respective API request to see detailed information. Click on the "Try-it-out"-button to see API-request-specific parameter and Instructions.
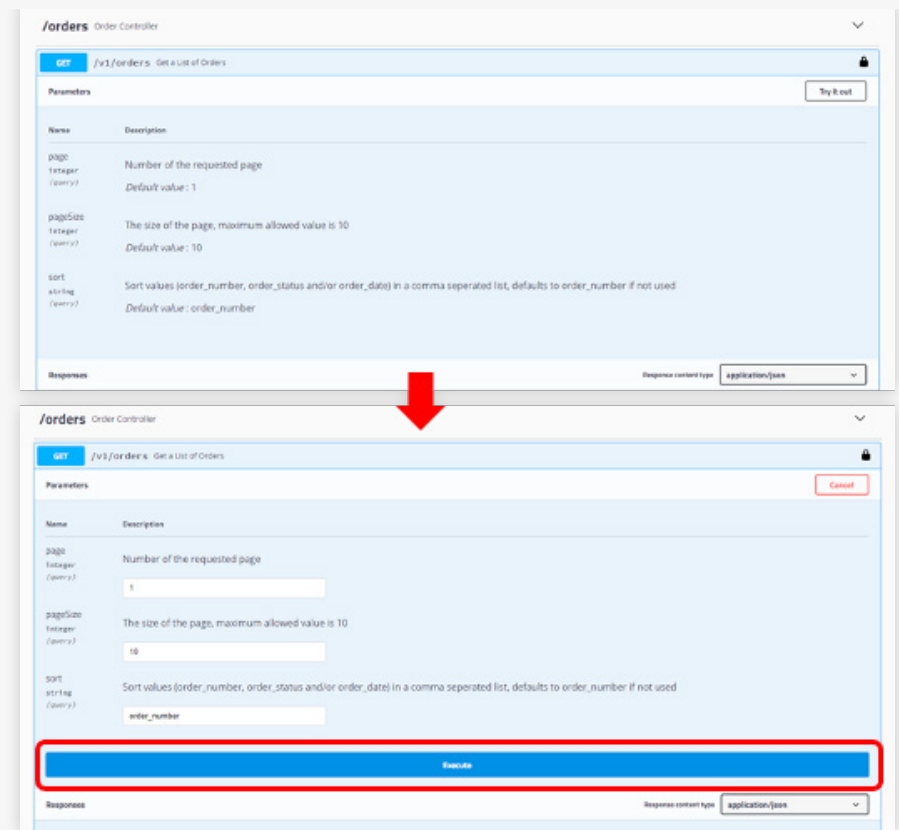

Figure 5: Parameter and instructions for API request: "Get list of orders" as example

Should you want to test the API request, please enter your ICCID followed by clicking the "Execute" button.

2. AUTHENTICATION REQUEST FOR IT-BACKEND INTEGRATION

Your server has to provide an authentication request to prove its identity to the IoT Easy Connect Portal. Only when you are authenticated you can send requests and receive the responses.

OAuth2 authentication is the method used for authenticating the requests send using the Customer Portal REST-API. Authenticate yourself via OAuth2 by entering your username and password, which you use to access the Portal GUI. Note that API access is only possible for users that are an "Administrator (Portal)".

In response to the OAuth2 authentication, the IoT Easy Connect portal sends you a token. This token con-tains a universal Unique Identifier (UUID) information that can be used by your user account.

Follow the next workflow to receive the token:
- Send your credential base64 encoded to the authentication server.
- Get a response including an UUID for authentication.
- Use the UUID to authenticate REST requests.

The UUID is valid for the next 240 minutes. The authentication process needs to be repeated after this time. Send the authentication request via **POST** to:

```
https://api-easyconnect.iot.t-mobile.nl/management-api/oauth/token
```

2.1 Authentication Request
The following is an authentication request containing your user name and password.

```json
**Header**
{
'Content-Type': 'application/x-www-form-urlencoded',
'authorization': 'Basic <base64 encoded username:password>'
}
**Body**
{
'grant_type': 'client_credentials'
}
```

2.2 Authentication Response
The following is an authentication response containing the UUID.

```json
{
"access_token":"6ba7b810-9dad-11d1-80b4-00c04fd43XXX", // this is the
UUID token you want to use for authentication
"token_type":"bearer",
"expires_in":3599,
"scope":"all",
"appToken":"<application_token>", // not relevant in this context
"userId":<id>,
"orgId":4321 // example
}
```

3. API DOCUMENTATION

You can find a full overview of the API documentation via **api-easyconnect.iot.t-mobile.nl**. As an alternative to the online overview of the API requests, you can also find a complete overview of all API requests in an API document.

The API document is identical in content, but it is better suited for import to your IT-Tool. Click the following the link to access the API-document directly: **api-easyconnect.iot.t-mobile. nl/management-api/api-docs**

Depending on your browser the website may not be fully compatible (Firefox 63.0 upwards is supported)

Figure 6: Accessing the API documentation

The next two sections focus on sending SMS from and to your IT-backend using the customer API.

3.1 Sending SMS from application to SIM (SMS MT)

You can send a SMS from your IT-backend using the API by creating a SMS in JSON format including a pos-sible source address (ICCID of your SIM Card) and the message text itself.

Send the SMS via POST to the BSC-Customer API with the ICCID of the destination device as a parameter.

```
https://api-easyconnect.iot.t-mobile.nl/management-api/v1/sims/<destination ICCID>/sms
```

Use the authentication UUID to authenticate the application.

REQUEST:

```json
**Header**
```json
{
'Content-Type': 'application/json',
'authorization': 'Bearer 6ba6a810-9mom-11d1-90b4-00c04fd43XXX', // example; the UUID token gotten
from authorization via OAuth2
'accept': 'application/json'
}
```
Body
```json
{
"payload": "message text",
"source_address": "898822806770xxx" // your ICCID
}
```
```

If successful, the server responds with "201 CREATED".

DELIVERY REPORT

Our SMS gateway server sends you a delivery report message after forwarding the SMS to the device. Pro-vide a REST interface in your application enabling PATCH requests.

```json
**Request**
```json
{
'status': {
'status': 'DELIVERED',
'id': '4'
},
'endpoint': {
'id': '8765432',
'name': '8988280666000000xxx'
},
'submit_date': '2018-08-17 16:46:48',
'organisation': {
'id': '4321'
},
'final_date': '2018-08-17 16:46:52',
'id': '6206'
}
```

3.2 Sending SMS from SIM to Application (SMS MO)

SMS communication is possible between the IoT Easy Connect portal and mobile device. If the SMS For-warding service is set-up, SMS MO can be forwarded by the SMS gateway to your IT-backend via HTTP.

CLIENT (SENDER)

Write a script to send SMS with a device (e.g. Quectel BG96) employing the SIM Card.

The SMSC address is 882 285 00001 6860
(see also the network settings menu in the customer portal or read it via an AT-command)

The destination number is not relevant since the SMS is forwarded to an application. You can for example choose 5432145 as the b-party address, but other numbers are also allowed.

SERVER

Provide a REST interface in your application enabling POST requests.
The SMS is forwarded as a JSON via POST.

```
**Headers**
```
Content-Type: application/json
```
**Body**
```json
{
'multi_part_info': {
'partno': 1,
'total': 1,
'identifier': '6202'
},
'payload': 'message text',
'submit_date': '2018-08-17 16:31:51', // GMT+0
'dest_address': '12345', // example
'organisation': {
'id': '4567'
},
'pid': '0',
'id': '6202',
'endpoint': {
'id': '8765432',
'name': '8988280666000000xxx' // from the sending SIM
},
'source_address': '882285100000xxx', // from the sending SIM
'dcs': '0'
}
```
```

With the dcs (data coding scheme) parameters you can change the encoding of the SMS. The value '0' rep-resents the default GSM 7 bit encoding, the value '4' enables an 8-bit data encoding.

The forwarding server expects a 2xx response after a successful forwarding. Otherwise the SMS will be buffered and re-transmitted.