

---

# Onomondo Docs Documentation

*Release latest*

Feb 15, 2018



---

# Contents

---

<b>1</b>	<b>API</b>	<b>1</b>
1.1	General Info . . . . .	1
1.2	SIM . . . . .	2
1.3	Connector . . . . .	5
1.4	Usage . . . . .	8
<b>2</b>	<b>Webhooks</b>	<b>9</b>
2.1	General Info . . . . .	9
2.2	Usage . . . . .	9
2.3	Network Validation . . . . .	10
2.4	Network Registration . . . . .	10



## 1.1 General Info

The API is a HTTP RESTful API based around resource oriented URLs. It uses standard HTTP verbs and returns JSON on all requests.

### 1.1.1 Base URL

All URLs referenced in the documentation have the following base:

```
https://api.onomondo.com
```

The API is served over HTTPS. To ensure data privacy, unencrypted HTTP is not supported.

### 1.1.2 Authentication

HTTP requests to the API are protected with API key authentication.

In short, you will use your API key in the `Authorization` header.

Sample Call in NodeJS Request:

```
var request = require("request");

var options = {
  method: "GET",
  url:    "/",
  headers: { Authorization: "YOUR-API-KEY" }
};

request(options, function (error, response, body) {
  if (error) throw new Error(error);
});
```

```
console.log(body);
});
```

## 1.2 SIM

### 1.2.1 Retrieve SIM

- Method

```
GET
```

- URL

```
/sim
```

- Data Params

Parameter	Required	Default Description
id	-	9 digit ID of the SIM

- Success Response

```
Code: 200
Content: { "id": "192837465",
           "alias": "4512345678",
           "active": true }
```

- Error Codes

Code	Error
400	id_invalid
403	id_forbidden
404	sim_not_found

- Sample Call in NodeJS Request

```
var request = require("request");

var options = {
  method: "GET",
  url: "/sim",
  headers: { Authorization: "YOUR-API-KEY" },
  json: { id: "192837465" }
};

request(options, function (error, response, body) {
  if (error) throw new Error(error);
  console.log(body);
});
```

## 1.2.2 Create SIM

- Method

```
POST
```

- URL

```
/sim
```

- Data Params

Parameter	Required	Default	Description
id	-		9 digit ID of the SIM
alias			TODO
active		true	TODO

- Success Response

```
Code: 200
Content: { "message": "OK" }
```

- Error Codes

Code	Error
400	id_invalid
400	alias_invalid
400	active_invalid
403	id_forbidden
409	alias_conflict

- Sample Call in NodeJS Request

```
var request = require("request");

var options = {
  method: "POST",
  url:    "/sim",
  headers: { Authorization: "YOUR-API-KEY" },
  json:   { id:    "192837465",
            alias: "My-Lego-Drone01",
            active: true }
};

request(options, function (error, response, body) {
  if (error) throw new Error(error);
  console.log(body);
});
```

## 1.2.3 Update SIM

- Method

PUT

- URL

/sim

- Data Params

Parameter	Required	Default	Description
id	-		9 digit ID of the SIM
alias			TODO
active		true	TODO

- Success Response

```
Code: 200
Content: { "message": "OK" }
```

- Error Codes

Code	Error
400	id_invalid
400	alias_invalid
400	active_invalid
403	id_forbidden
404	sim_not_found
409	alias_conflict

- Sample Call in NodeJS Request

```
var request = require("request");

var options = {
  method: "PUT",
  url:    "/sim",
  headers: { Authorization: "YOUR-API-KEY" },
  json:   { id:    "192837465",
            alias: "My-Lego-Drone01",
            active: false }
};

request(options, function (error, response, body) {
  if (error) throw new Error(error);
  console.log(body);
});
```

## 1.2.4 Delete SIM

- Method

DELETE



- URL

```
/sim
```

- Data Params

Parameter	Required	Default	Description
id	-		9 digit ID of the SIM

- Success Response

```
Code: 200
Content: { "message": "OK" }
```

- Error Codes

Code	Error
400	id_invalid
403	id_forbidden
404	sim_not_found

- Sample Call in NodeJS Request

```
var request = require("request");

var options = {
  method: "DELETE",
  url:    "/sim",
  headers: { Authorization: "YOUR-API-KEY" },
  json:   { id: "192837465" }
};

request(options, function (error, response, body) {
  if (error) throw new Error(error);
  console.log(body);
});
```

## 1.3 Connector

### 1.3.1 Retrieve Connector

- Method

```
GET
```

- URL

```
/connector
```

- Data Params

Parameter	Description
id	ID of the Connector

- Success Response

TODO

- Error Codes

TODO

- Sample Call in NodeJS Request

TODO

### 1.3.2 Create Connector

- Method

POST

- URL

/connector

- Data Params

Parameter	Description
id	ID of the Connector

- Data Params

TODO

- Success Response

TODO

- Error Codes

TODO

- Sample Call in NodeJS Request

TODO

### 1.3.3 Update Connector

- Method

PUT

- URL

`/connector`

- Data Params

Parameter	Description
id	ID of the Connector

- Data Params

TODO

- Success Response

TODO

- Error Codes

TODO

- Sample Call in NodeJS Request

TODO

### 1.3.4 Delete Connector

- Method

DELETE

- URL

`/connector`

- Data Params

Parameter	Description
id	ID of the Connector

- Success Response

TODO

- Error Codes

TODO

- Sample Call in NodeJS Request

TODO

## 1.4 Usage

### 1.4.1 Retrieve Usage

- Method

GET

- URL

/usage

- Data Params

Parameter	Required	Default Description
sim_id		9 digit ID of the SIM
time_begin		TODO
time_end		TODO

- Success Response

TODO

- Error Codes

Code	Error
400	sim_id_invalid
400	time_begin_invalid
400	time_begin_invalid
403	sim_id_forbidden
404	sim_id_not_found

- Sample Call in NodeJS Request

TODO

### 2.1 General Info

The endpoints has 15 seconds to accept the event. If a success HTTP status code has not been returned within these 15 seconds, the event will be discarded.

#### 2.1.1 Authentication

The webhooks authenticate against the endpoints using API key authentication:

```
POST /usage
Host: www.example.com
Content-Type: application/json
Authorization: OUR-API-KEY
```

The webhooks is served over HTTPS. To ensure data privacy, unencrypted HTTP is not supported.

### 2.2 Usage

Sample of Usage:

```
{ "time": "2018-01-01 01:23:45",
  "bill_id": "1234567890-1234",
  "sim_id": "192837465",
  "country_code": "dk",
  "operation": "finalize",
  "upload": "0",
  "download": "0" }
```

## 2.3 Network Validation

Sample of Network Validation:

```
{ "time":          "2018-01-01 01:23:45",  
  "sim_id":       "192837465",  
  "imei":         "1234567890987654",  
  "country_code": "dk" }
```

## 2.4 Network Registration

Sample of Network Registration:

```
{ "time":          "2018-01-01 01:23:45",  
  "sim_id":       "192837465",  
  "imei":         "1234567890987654",  
  "country_code": "dk",  
  "country_code_previous": "de" }
```