



monotch

SMART MOBILITY PLATFORMS

TOPOLEX API v1.0

Table of contents

1	Version history	3
2	Introduction	4
3	Security	5
3.1	Encryption	5
3.2	Authorization	5
4	API endpoints	6
4.1	Document types	7
4.1.1	GET /documenttypes	7
4.2	Road regulators	9
4.2.1	GET /roadregulators	9
4.3	Intersections	11
4.3.1	GET /intersections	11
4.4	Revisions	13
4.4.1	POST /revisions	13
4.4.2	GET /currentrevisions	16
4.4.3	HEAD /currentrevisions	18

1 Version history

Version	Date	Author	State	Changes
0.1	20-03-2018	Y.Duran	Draft	Initial version
0.2	22-03-2018	M.Nieuwdorp	Draft	Added documentation for API endpoints: <ul style="list-style-type: none">• GET /documenttypes• GET /roadregulators• GET /intersections• POST /revisions• GET /currentrevisions• HEAD /currentrevisions
1.0	28-03-2018	L. Rijnveld	Release	Bumped version number for final release

2 Introduction

In order to facilitate automatic intersection document management a REST API is available with specific end points for uploading and downloading documents.

3 Security

3.1 Encryption

The API will be exposed using HTTPS only.

3.2 Authorization

The API uses "Basic authentication". Use the same username and password as on topolex.nl.

4 API endpoints

Method	URI	Description
GET	/documenttypes	Retrieves all document types
GET	/roadregulators	Retrieves all road regulators
GET	/intersections	Retrieves all intersections
POST	/revisions	Uploads a new or updated version of a document (multipart/form-data) of a specific type for a specific intersection
GET	/currentrevisions	Downloads one or more documents of a specific intersection
HEAD	/currentrevisions	Returns only the header of GET /currentrevisions which contains the document's version

4.1 Document types

4.1.1 GET /documenttypes

Retrieves all document types

4.1.1.1 Request

Request

```
GET <API base URL>/documenttypes HTTP/1.1
Host: <hostname>
Authorization: Basic <base64-encoding of username:password>
```

4.1.1.2 Response

Response body (JSON)

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "types": [
    <document type>, <document type>
  ]
}
```

4.1.1.3 Example

request

```
GET api/v1/types HTTP/1.1
Host: topolex.nl
Authorization: Basic dG9wb2xleDp0b3BvbGV4
```

response

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "types": [
    {
      "name": "ITF",
      "multi": false
    }
  ]
}
```


4.2 Road regulators

4.2.1 GET /roadregulators

Retrieves all road regulators

4.2.1.1 Request

Request

```
GET <API base URL>/roadregulators HTTP/1.1
Host: <hostname>
Authorization: Basic <base64-encoding of username:password>
```

4.2.1.2 Response

Response body (JSON)

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "roadRegulators": [
    <road regulator>, <road regulator>
  ]
}
```

4.2.1.3 Example

request

```
GET api/v1/roadregulators HTTP/1.1
Host: topolex.nl
Authorization: Basic dG9wb2xleDp0b3BvbGV4
```

response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "roadRegulators": [
    {
      "name": "Highway regulator"
    },
    {
      "name": "North-west regional road regulator"
    }
  ]
}
```

4.3 Intersections

4.3.1 GET /intersections

Retrieves all intersections

4.3.1.1 Request

Request

```
GET <API base URL>/intersections HTTP/1.1
Host: <hostname>
Authorization: Basic <base64-encoding of username:password>
```

4.3.1.2 Response

Response body (JSON)

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "intersections": [
    <intersection>, <intersection>
  ]
}
```

4.3.1.3 Example

request

```
GET api/v1/intersections HTTP/1.1
Host: topolex.nl
Authorization: Basic dG9wb2xleDp0b3BvbGV4
```

response

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "intersections": [
    {
      "roadRegulator": "Highway regulator",
      "intersectionId": "exhwint2"
    },
    {
      "roadRegulator": "Highway regulator",
      "intersectionId": "exhwint1"
    }
  ]
}
```

4.4 Revisions

4.4.1 POST /revisions

Creates a new document revision (which may be a new document or a newer version of an existing one) for a new or previously existing intersection. This is done using a "Multipart form POST", with the intersection id, road regulator and document type in the query, and the file in the 'file' field of the form.

4.4.1.1 Request

Request	
<pre>POST <API base URL>/revisions?intersectionId=<intersection id>&roadRegulator=<road regulator>&documentType=<document type> HTTP/1.1 Host: <hostname> Authorization: Basic <base64-encoding of username:password> Content-Type: multipart/form-data; boundary=<form data boundary marker> --<form data boundary marker> Content-Disposition: file; name="file"; filename="<document file name>" Content-Type: <content type> <file contents> --<form data boundary marker></pre>	
Name	Description
road regulator	Name of existing road regulator.
document type	Name of existing document type.
intersection id	Intersection identifier Must be exactly 8 characters and made up of letters and numbers only.
document file name	File to be sent.
file contents	

4.4.1.2 Response

Response body (JSON)

HTTP/1.1 201 CREATED

4.4.1.3 Example

request

```
POST api/v1/revisions?intersectionId=exhwint1&roadRegulator=Highway regulator&documentType=ITF HTTP/1.1
Host: topolex.nl
Authorization: Basic dG9wb2xleDp0b3BvbGV4
Content-Type: multipart/form-data; boundary=7MA4YWxkTrZu0gW
--7MA4YWxkTrZu0gW
Content-Disposition: file; filename="itf.xml"
Content-Type: text/plain
<topology>
  <formatVersion>2.0</formatVersion>
  ...
</topology>
--7MA4YWxkTrZu0gW
```

response

```
HTTP/1.1 201 CREATED
```

4.4.2 GET /currentrevisions

Retrieves the set of latest revisions of all documents associated with a given intersection, as a .zip file when the query leads to multiple results and as a single file otherwise. Optionally, the search request can be narrowed by also supplying a document type, or widened by disabling the valid-only filter. By default, the valid-only filter is enabled.

4.4.2.1 Request

Request

```
GET <API base URL>/currentrevisions?intersectionId=<intersection id>[&documenttype=<document type>][&validOnly=Yes|No] HTTP/1.1
Host: <hostname>
Authorization: Basic <base64-encoding of username:password>
```

4.4.2.2 Response

Response body (JSON)

```
HTTP/1.1 200 OK
X-RevisionIdentifier: <revision identifier>
Content-Type: application/octet-stream;charset=UTF-8
Content-Disposition: attachment;filename=<suggested file name>
<binary>
```


4.4.2.3 Example 1

request

```
GET api/v1/currentrevisions?intersectionId=exhwint1 HTTP/1.1
Host: topolex.nl
Authorization: Basic dG9wb2xleDp0b3BvbGV4
```

response

```
HTTP/1.1 200 OK
X-RevisionIdentifier: 1.4
Content-Type: application/octet-stream;charset=UTF-8
Content-Disposition: attachment;filename=exhwint8_2018_03_22_07_36_10.zip
...
```

4.4.2.4 Example 2

request

```
GET api/v1/currentrevisions?intersectionId=exhwint1&documentType=ITF HTTP/1.1
Host: topolex.nl
Authorization: Basic dG9wb2xleDp0b3BvbGV4
```

response

```
HTTP/1.1 200 OK
X-RevisionIdentifier: 1.2
Content-Type: text/plain;charset=UTF-8
Content-Disposition: attachment;filename=Highway_regulator_exhwint1_ITF.xml
<topology>
  <formatVersion>2.0</formatVersion>
  ...
</topology>
```

4.4.3 HEAD /currentrevisions

Retrieves only the HTTP headers associated with the GET /currentrevisions request, omitting any file content

4.4.3.1 Request

Request

```
HEAD <API base URL>/currentrevisions?intersectionId=<intersection id>[&documenttype=<document type>][&validOnly=Yes|No] HTTP/1.1
Host: <hostname>
Authorization: Basic <base64-encoding of username:password>
```

4.4.3.2 Response

Response body (JSON)

```
HTTP/1.1 200 OK
X-RevisionIdentifier: <revision identifier>
Content-Type: application/octet-stream;charset=UTF-8
Content-Disposition: attachment;filename=<suggested file name>
```

4.4.3.3 Example

request

```
HEAD api/v1/currentrevisions?intersectionId=exhwint8 HTTP/1.1
Host: topolex.nl
Authorization: Basic dG9wb2xleDp0b3BvbGV4
```

response

```
HTTP/1.1 200 OK
X-RevisionIdentifier: 1.4
Content-Type: application/octet-stream;charset=UTF-8
Content-Disposition: attachment;filename=exhwint8_2018_03_22_07_36_10.zip
```