Geodesy Report 16-03 2016

Geodetic REST Web services (REFRAME Web API)

User manual

Jérôme Ray

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Bundesamt für Landestopografie swisstopo
Office fédéral de topographie swisstopo
Ufficio federale di topografia swisstopo
Uffizi federal da topografia swisstopo

www.swisstopo.ch

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Federal Department of Defence,
Civil Protection and Sport DDPS

Federal Office of Topography swisstopo

Geodetic REST Web services (REFRAME Web API)

16-03 A

User Manual

Jérôme Ray

April 2016

Table of contents

© 2016 **swisstopo**
Bundesamt für Landestopografie
Office fédéral de topographie
Ufficio federale di topografia
Uffizi federal da topografia
Federal Office of Topography
SQS-Zertifikat ISO 9001:2009

Redaction: A. Wiget
Seftigenstrasse 264
CH-3084 Wabern

Tel: +41 58 469 01 11
Fax: +41 58 496 04 59
E-mail: infogeo@swisstopo.ch

1 Introduction

1.1 REFRAME

REFRAME is a planimetry and/or altimetry transformation software module for technical surveying or cadastral surveying applications, with the highest precision requirements.

This product supports the following Swiss reference frames/systems:

- Plane coordinates LV03 (CH1903)
- Plane coordinates LV95 (CH1903+)
- Global geographic or geocentric coordinates CHTRS95/ETRS89/WGS84
- National levelling network LN02 (levelled heights)
- National height network LHN95 (orthometric heights, CHGeo2004)
- Ellipsoidal heights (Bessel or GRS80)

For more information about geodetic software or web services, please refer to our website:

http://www.swisstopo.ch/geosoftware

And for technical aspects about Swiss reference frames and the new frame LV95:

http://www.swisstopo.ch/lv95

A REFRAME library is also available. It allows software developers to implement REFRAME computing in their own applications with .NET, COM or LIB components (.NET or C++ applications, Silverlight websites, VBA macros, further DLL or plug-ins, etc.) as well as Java desktop or Web applications.

1.2 Web API (REST)

As described in Wikipedia, a Web API is an application programming interface (API) for either a web server or a web browser. A RESTful web API is accessible via standard HTTP methods by a variety of HTTP clients including browsers and mobile devices.

Representational state transfer (REST) is a web standards based architecture and uses HTTP protocol for data communication. A RESTful web service usually defines a URI, Uniform Resource Identifier for a service, provides resource representation such as JSON and set of HTTP Methods.

2 Technical aspects

This API can compute planimetric and altimetric reference frame changes. All the transformations are not linear and are based on binary datasets organized in grids (altimetry) or triangular network (planimetry).

The following datasets are embedded in this API:

- **CHENyx06 (FINELTRA)**

  The transformation from **LV03** to **LV95** coordinates and reverse is based on the **FINELTRA algorithm** ("affine transformation with finite elements"). It is based on a triangular network transformation dataset named CHENyx06. Current version: 03.2007.

- **HTRANS**

  The transformation from **LN02** (levelled) to **LHN95** (orthometric) heights is based on the **HTRANS** transformation dataset. This transformation is based on 3 regular grids (with a resolution of 1km x 1km) containing data depending on the influence of the gravimetric field, the incidence of kinematics (alpine rising over 100 years) and distortions of LN02 (quality of the precise levelling 1864-91). Current version: 2005.

- **CHGeo2004**

  The transformation from **LHN95** (orthometric) to **Bessel** (ellipsoidal) heights is based on the Swiss geoid model. This model is defined as a regular grid (with a resolution of 1km x 1km). Current version: 2004.

Internally, the GNSS software used at Swipos management center (Trimble Pivot) works with ETRS89 geocentric coordinates. To be able to transform directly such coordinates, this library integrates the Swiss projection formulas:

- **Swiss Grid**

  For the transformation from Swiss LV95 plane coordinates to global geographic or geocentric coordinates (CHTRS95/ETRS89/WGS84) or reverse, the rigorous projection formulas[1] are used (Oblique Mercator and "LV95" parameters). The Swiss projection, usually called "Swiss Grid", is an oblique Mercator projection.

  If LV03 coordinates have to be transformed to global coordinates, they must be transformed to LV95 first, and then to ETRS89. The same intermediate step applies for the reverse computation: ETRS89 to LV95 and then to LV03.

3 Installation and requirements

You don't need any installation, you only require a web access to our website. If some security rules or a proxy do not allow it, you will have to add the domain "**geodesy.geo.admin.ch**" to the authorized domains.

Note that if you plan to integrate the use of our REST services in your own Web service or API, **cross-origin resource sharing** (CORS[2]), JSONP or iFrame integration may be needed, depending on your architecture and security rules, because your application will not be hosted in the same domain than our REST services.

[1] More information about the Swiss projection and "LV95" parameters : *http://www.swisstopo.admin.ch/internet/swisstopo/en/home/topics/survey/sys/refsys.html*

[2] *http://www.w3.org/TR/cors/*

4 Geodetic REST services (swisstopo REFRAME)

Swisstopo offers 12 geodetic REST services in order to compute all the possible transformations between the most useful reference frames in Switzerland: ETRF93 (≈WGS84), LV95 and LV93 in position, Bessel (ellipsoidal), LHN95 and LN02 in height.

These services are:

- **WGS84 to LV95**
- **LV95 to WGS84**
- **WGS84 to LV03**
- **LV03 to WGS84**
- **LV95 to LV03**
- **LV03 to LV95**

And:

- **LN02 to LHN95**
- **LHN95 to LN02**
- **LHN95 to Bessel**
- **Bessel to LHN95**
- **LN02 to Bessel**
- **Bessel to LN02**

All these services work in a similar way: arguments and output data are similar, only the URL is different.

Two output data representations (formats) are available: **JSON**[3] and **GeoJSON**[4]**.** JSON is an open standard data-interchange format, based on a subset of the JavaScript programming language. It is a text format that is completely language independent. JSON supports objects (set of name and value pairs) and arrays (ordered collection of values). GeoJSON is based on JSON but designed for representing simple geographical features.

All queries will get a **HTTP response status code**. This code indicates the client which kind of result has been sent or if an error has been encountered. Used status codes are:

- **200 OK**
- **400 Bad Request**

[3] http://www.json.org/

[4] http://geojson.org/

4.1 WGS84 to LV95

4.1.1 Use

Convert global WGS84 (or ETRS89/ETRF93) global coordinates into Swiss projected coordinates LV95.

4.1.2 Base URI

http://geodesy.geo.admin.ch/reframe/wgs84tolv95

4.1.3 Arguments

Required:

- **easting** Ellipsoidal longitude in decimal degrees [°] on WGS84 (≈ GRS80)
- **northing** Ellipsoidal latitude in decimal degrees [°] on WGS84 (≈ GRS80)

Optional:

- **altitude** Ellipsoidal height in meters [m] on WGS84 (≈ GRS80)
- **format**
  - “**json**” JSON data representation
  - “**geojson**” GeoJSON data representation, used by default if argument not specified
- **cb** Callback function name

4.1.4 HTTP status codes

- **200** The transformation **succeeded**
- **400** The transformation **failed**: insufficient or incorrect arguments

4.1.5 Output data

JSON object (if argument “format”=”json”):

- **easting** East coordinate in meters [m] in LV95
- **northing** Nord coordinate in meters [m] in LV95
- **altitude** Ellipsoidal height in meters [m] on Bessel (if argument “altitude” was specified)

or GeoJSON object (if argument “format”=”json”):

- **type** “Point”
- **coordinates** 2D or 3D vector (if argument “altitude” was specified) containing East and North coordinates in meters [m] in LV95 and optionally the ellipsoidal height in meters [m] on Bessel ellipsoid

4.1.6 Examples

http://geodesy.geo.admin.ch/reframe/wgs84tolv95?easting=7.43863&northing=46.95108&altitude=550.0&format=json

answer:

```
{"easting": "2599999.810036594", "northing": "1199999.6801912596", "altitude": "500.3777916841209"}
```

http://geodesy.geo.admin.ch/reframe/wgs84tolv95?easting=7.43863&northing=46.95108&cb=myfunction

answer:

```
myfunction({ "type": "Point", "coordinates": [2599999.810602969, 1199999.6813454267] });
```

4.2 LV95 to WGS84

4.2.1 Use

Convert Swiss projected coordinates LV95 into WGS84 (or ETRS89/ETRF93) global coordinates.

4.2.2 Base URI

http://geodesy.geo.admin.ch/reframe/lv95towgs84

4.2.3 Arguments

Required:

- **easting** East coordinate in meters [m] in LV95
- **northing** Nord coordinate in meters [m] in LV95

Optional:

- **altitude** Ellipsoidal height in meters [m] on Bessel
- **format**
  - “**json**” JSON data representation
  - “**geojson**” GeoJSON data representation, used by default if argument not specified
- **cb** Callback function name

4.2.4 HTTP status codes

- **200** The transformation **succeeded**
- **400** The transformation **failed**: insufficient or incorrect arguments

4.2.5 Output data

JSON object (if argument “format”=”json”):

- **easting** Ellipsoidal longitude in decimal degrees [°] on WGS84 (≈ GRS80)
- **northing** Ellipsoidal latitude in decimal degrees [°] on WGS84 (≈ GRS80)
- **altitude** Ellipsoidal height in meters [m] on WGS84 (≈ GRS80) (if argument “altitude” was specified)

or GeoJSON object (if argument “format”=”json”):

- **type** “Point”
- **coordinates** 2D or 3D vector (if argument “altitude” was specified) containing ellipsoidal longitude and latitude in decimal degrees and optionally the ellipsoidal height in meters [m] on WGS84 (≈ GRS80)

4.2.6 Examples

http://geodesy.geo.admin.ch/reframe/lv95towgs84?easting=2600000&northing=1200000&altitude=550.0&format=json

answer:
{"easting": "7.438632502714565", "northing": "46.95108288705891", "altitude": "599.6221912624314"}

http://geodesy.geo.admin.ch/reframe/lv95towgs84?easting=2600000&northing=1200000&cb=myfunction

answer:
myfunction({ "type": "Point", "coordinates": [7.438632495274896, 46.95108287667706] });

4.3 WGS84 to LV03

4.3.1 Use

Convert global WGS84 (or ETRS89/ETRF93) global coordinates into Swiss projected coordinates LV03.

4.3.2 Base URI

http://geodesy.geo.admin.ch/reframe/wgs84tolv03

4.3.3 Arguments

Required:

- **easting** Ellipsoidal longitude in decimal degrees [°] on WGS84 (≈ GRS80)
- **northing** Ellipsoidal latitude in decimal degrees [°] on WGS84 (≈ GRS80)

Optional:

- **altitude** Ellipsoidal height in meters [m] on WGS84 (≈ GRS80)
- **format**
  - “**json**” JSON data representation
  - “**geojson**” GeoJSON data representation, used by default if argument not specified
- **cb** Callback function name

4.3.4 HTTP status codes

- **200** The transformation **succeeded**
- **400** The transformation **failed**: insufficient or incorrect arguments

4.3.5 Output data

JSON object (if argument “format”=”json”):

- **easting** East coordinate in meters [m] in LV03
- **northing** Nord coordinate in meters [m] in LV03
- **altitude** Ellipsoidal height in meters [m] on Bessel (if argument “altitude” was specified)

or GeoJSON object (if argument “format”=”json”):

- **type** “Point”
- **coordinates** 2D or 3D vector (if argument “altitude” was specified) containing East and North coordinates in meters [m] in LV03 and optionally the ellipsoidal height in meters [m] on Bessel ellipsoid

4.3.6 Examples

http://geodesy.geo.admin.ch/reframe/wgs84tolv03?easting=7.43863&northing=46.95108&altitude=550.0&format=json

answer:
{"easting": "599999.7270399226", "northing": "199999.61419550685", "altitude": "500.3777916841209"}

http://geodesy.geo.admin.ch/reframe/wgs84tolv03?easting=7.43863&northing=46.95108&cb=myfunction

answer:
myfunction({ "type": "Point", "coordinates": [599999.7276062882, 199999.6153496582] });

4.4 LV03 to WGS84

4.4.1 Use

Convert Swiss projected coordinates LV03 into WGS84 (or ETRS89/ETRF93) global coordinates.

4.4.2 Base URI

http://geodesy.geo.admin.ch/reframe/lv03towgs84

4.4.3 Arguments

Required:

- **easting** East coordinate in meters [m] in LV03
- **northing** Nord coordinate in meters [m] in LV03

Optional:

- **altitude** Ellipsoidal height in meters [m] on Bessel
- **format**
  - “**json**” JSON data representation
  - “**geojson**” GeoJSON data representation, used by default if argument not specified
- **cb** Callback function name

4.4.4 HTTP status codes

- **200** The transformation **succeeded**
- **400** The transformation **failed**: insufficient or incorrect arguments

4.4.5 Output data

JSON object (if argument “format”=”json”):

- **easting** Ellipsoidal longitude in decimal degrees [°] on WGS84 (≈ GRS80)
- **northing** Ellipsoidal latitude in decimal degrees [°] on WGS84 (≈ GRS80)
- **altitude** Ellipsoidal height in meters [m] on WGS84 (≈ GRS80) (if argument “altitude” was specified)

or GeoJSON object (if argument “format”=”json”):

- **type** “Point”
- **coordinates** 2D or 3D vector (if argument “altitude” was specified) containing ellipsoidal longitude and latitude in decimal degrees and optionally the ellipsoidal height in meters [m] on WGS84 (≈ GRS80)

4.4.6 Examples

http://geodesy.geo.admin.ch/reframe/lv03towgs84?easting=600000&northing=200000&altitude=550.0&format=json

answer:
{"easting": "7.438632502714565", "northing": "46.95108288705891", "altitude": "599.6221912624314"}

http://geodesy.geo.admin.ch/reframe/lv03towgs84?easting=600000&northing=200000&cb=myfunction

answer:
myfunction({ "type": "Point", "coordinates": [7.438632495274896, 46.95108287667706] });

4.5 LV95 to LV03

4.5.1 Use

Convert Swiss projected coordinates LV95 into LV03.

4.5.2 Base URI

http://geodesy.geo.admin.ch/reframe/lv95tolv03

4.5.3 Arguments

Required:

• **easting**	East coordinate in meters [m] in LV95
• **northing**	Nord coordinate in meters [m] in LV95

Optional:

• **altitude**	Ellipsoidal height in meters [m] on Bessel
• **format**	
○ "**json**"	JSON data representation
○ "**geojson**"	GeoJSON data representation, used by default if argument not specified
• **cb**	Callback function name

4.5.4 HTTP status codes

• **200**	The transformation **succeeded**
• **400**	The transformation **failed**: insufficient or incorrect arguments

4.5.5 Output data

JSON object (if argument "format"="json"):

• **easting**	East coordinate in meters [m] in LV03
• **northing**	Nord coordinate in meters [m] in LV03
• **altitude**	Ellipsoidal height in meters [m] on Bessel (if argument "altitude" was specified, this transformation does not affect the height, which will be equal as in input)

or GeoJSON object (if argument "format"="json"):

• **type**	"Point"
• **coordinates**	2D or 3D vector (if argument "altitude" was specified) containing East and North coordinates in meters [m] in LV03 and optionally the ellipsoidal height in meters [m] on Bessel ellipsoid

4.5.6 Examples

http://geodesy.geo.admin.ch/reframe/lv95tolv03?easting=2600000&northing=1200000&altitude=550.0&format=json

answer:
{"easting": "600000.0", "northing": "200000.0", "altitude": "550.0"}

http://geodesy.geo.admin.ch/reframe/lv95tolv03?easting=2600000&northing=1200000&cb=myfunction

answer:
myfunction({ "type": "Point", "coordinates": [600000.0, 200000.0] });

4.6 LV03 to LV95

4.6.1 Use

Convert Swiss projected coordinates LV03 into LV95.

4.6.2 Base URI

http://geodesy.geo.admin.ch/reframe/lv03tolv95

4.6.3 Arguments

Required:

- **easting** East coordinate in meters [m] in LV03
- **northing** Nord coordinate in meters [m] in LV03

Optional:

- **altitude** Ellipsoidal height in meters [m] on Bessel
- **format**
  - “**json**” JSON data representation
  - “**geojson**” GeoJSON data representation, used by default if argument not specified
- **cb** Callback function name

4.6.4 HTTP status codes

- **200** The transformation **succeeded**
- **400** The transformation **failed**: insufficient or incorrect arguments

4.6.5 Output data

JSON object (if argument “format”=”json”):

- **easting** East coordinate in meters [m] in LV95
- **northing** Nord coordinate in meters [m] in LV95
- **altitude** Ellipsoidal height in meters [m] on Bessel (if argument “altitude” was specified, this transformation does not affect the height, which will be equal as in input)

or GeoJSON object (if argument “format”=”json”):

- **type** “Point”
- **coordinates** 2D or 3D vector (if argument “altitude” was specified) containing East and North coordinates in meters [m] in LV95 and optionally the ellipsoidal height in meters [m] on Bessel ellipsoid

4.6.6 Examples

http://geodesy.geo.admin.ch/reframe/lv95tolv03?easting=2600000&northing=1200000&altitude=550.0&format=json

answer:

{"easting": "600000.0", "northing": "200000.0", "altitude": "550.0"}

http://geodesy.geo.admin.ch/reframe/lv95tolv03?easting=2600000&northing=1200000&cb=myfunction

answer:

myfunction({ "type": "Point", "coordinates": [600000.0, 200000.0] });

4.7 LN02 to LHN95

4.7.1 Use

Convert Swiss usual heights LN02 (national leveling) into orthometric heights LHN95 (national height network).

4.7.2 Base URI

http://geodesy.geo.admin.ch/reframe/ln02tolhn95

4.7.3 Arguments

Required:

easting	East coordinate in meters [m] in LV03 or LV95
northing	North coordinate in meters [m] in LV03 or LV95
altitude	Usual height in meters [m] in LN02

Optional:

- **format**
  - “**json**” JSON data representation
  - “**geojson**” GeoJSON data representation, used by default if argument not specified
- **cb** Callback function name

4.7.4 HTTP status codes

200	The transformation **succeeded**
400	The transformation **failed**: insufficient or incorrect arguments

4.7.5 Output data

JSON object (if argument “format”=”json”):

easting	East coordinate in meters [m] in LV03 or LV 95 (this transformation does not affect the position, which will be equal as in input)
northing	Nord coordinate in meters [m] in LV03 or LV95 (this transformation does not affect the position, which will be equal as in input)
altitude	Orthometric height in meters [m] in LHN95

or GeoJSON object (if argument “format”=”json”):

type	“Point”
coordinates	3D vector containing East and North coordinates in meters [m] in LV03 or LV95 (same as in input) and orthometric height in meters [m] in LHN95

4.7.6 Examples

http://geodesy.geo.admin.ch/reframe/ln02tolhn95?easting=600000&northing=200000&altitude=550.0&format=json

answer:
{"easting": "600000.0", "northing": "200000.0", "altitude": "549.9579940080323"}

http://geodesy.geo.admin.ch/reframe/ln02tolhn95?easting=2600000&northing=1200000&altitude=550&cb=myfunction

answer:
myfunction({ "type": "Point", "coordinates": [2600000.0, 1200000.0, 549.9579940080323] });

4.8 LHN95 to LN02

4.8.1 Use

Convert Swiss orthometric heights LHN95 (national height network) into usual heights LN02 (national leveling).

4.8.2 Base URI

http://geodesy.geo.admin.ch/reframe/lhn95toln02

4.8.3 Arguments

Required:

- **easting** East coordinate in meters [m] in LV03 or LV95
- **northing** North coordinate in meters [m] in LV03 or LV95
- **altitude** Orthometric height in meters [m] in LHN95

Optional:

- **format**
    - “**json**” JSON data representation
    - “**geojson**” GeoJSON data representation, used by default if argument not specified
- **cb** Callback function name

4.8.4 HTTP status codes

- **200** The transformation **succeeded**
- **400** The transformation **failed**: insufficient or incorrect arguments

4.8.5 Output data

JSON object (if argument “format”=”json”):

- **easting** East coordinate in meters [m] in LV03 or LV95 (this transformation does not affect the position, which will be equal as in input)
- **northing** Nord coordinate in meters [m] in LV03 or LV95 (this transformation does not affect the position, which will be equal as in input)
- **altitude** Usual height in meters [m] in LN02

or GeoJSON object (if argument “format”=”json”):

- **type** “Point”
- **coordinates** 3D vector containing East and North coordinates in meters [m] in LV03 or LV95 (same as in input) and orthometric height in meters [m] in LHN95

4.8.6 Examples

http://geodesy.geo.admin.ch/reframe/lhn95toln02?easting=600000&northing=200000&altitude=550.0&format=json

answer:
{"easting": "600000.0", "northing": "200000.0", "altitude": "550.0420059919677"}

http://geodesy.geo.admin.ch/reframe/lhn95toln02?easting=2600000&northing=1200000&altitude=550&cb=myfunction

answer:
myfunction({ "type": "Point", "coordinates": [2600000.0, 1200000.0, 550.0420059919677] });

4.9 LHN95 to Bessel (ellipsoidal)

4.9.1 Use

Convert Swiss orthometric heights LHN95 (national height network) into ellipsoidal heights (on Bessel 1841).

4.9.2 Base URI

http://geodesy.geo.admin.ch/reframe/lhn95tobessel

4.9.3 Arguments

Required:

easting	East coordinate in meters [m] in LV03 or LV95
northing	North coordinate in meters [m] in LV03 or LV95
altitude	Orthometric height in meters [m] in LHN95

Optional:

- **format**
  - “**json**” JSON data representation
  - “**geojson**” GeoJSON data representation, used by default if argument not specified
- **cb** Callback function name

4.9.4 HTTP status codes

200	The transformation **succeeded**
400	The transformation **failed**: insufficient or incorrect arguments

4.9.5 Output data

JSON object (if argument “format”=”json”):

easting	East coordinate in meters [m] in LV03 or LV95 (this transformation does not affect the position, which will be equal as in input)
northing	Nord coordinate in meters [m] in LV03 or LV95 (this transformation does not affect the position, which will be equal as in input)
altitude	Ellipsoidal height in meters [m] on Bessel

or GeoJSON object (if argument “format”=”json”):

type	“Point”
coordinates	3D vector containing East and North coordinates in meters [m] in LV03 or LV95 (same as in input) and ellipsoidal height in meters [m] on Bessel

4.9.6 Examples

http://geodesy.geo.admin.ch/reframe/lhn95tobessel?easting=600000&northing=200000&altitude=550.0&format=json

answer:
{"easting": "600000.0", "northing": "200000.0", "altitude": "549.3664000034332"}

http://geodesy.geo.admin.ch/reframe/lhn95tobessel?easting=2600000&northing=1200000&altitude=550&cb=myfunction

answer:
myfunction({ "type": "Point", "coordinates": [2600000.0, 1200000.0, 549.3664000034332] });

4.10 Bessel (ellipsoidal) to LHN95

4.10.1 Use

Convert Swiss ellipsoidal heights (on Bessel 1841) into orthometric heights LHN95 (national height network).

4.10.2 Base URI

http://geodesy.geo.admin.ch/reframe/besseltolhn95

4.10.3 Arguments

Required:

- **easting** East coordinate in meters [m] in LV03 or LV95
- **northing** North coordinate in meters [m] in LV03 or LV95
- **altitude** Ellipsoidal height in meters [m] on Bessel

Optional:

- **format**
  - “**json**” JSON data representation
  - “**geojson**” GeoJSON data representation, used by default if argument not specified
- **cb** Callback function name

4.10.4 HTTP status codes

- **200** The transformation **succeeded**
- **400** The transformation **failed**: insufficient or incorrect arguments

4.10.5 Output data

JSON object (if argument “format”=”json”):

- **easting** East coordinate in meters [m] in LV03 or LV95 (this transformation does not affect the position, which will be equal as in input)
- **northing** Nord coordinate in meters [m] in LV03 or LV95 (this transformation does not affect the position, which will be equal as in input)
- **altitude** Orthometric height in meters [m] in LHN95

or GeoJSON object (if argument “format”=”json”):

- **type** “Point”
- **coordinates** 3D vector containing East and North coordinates in meters [m] in LV03 or LV95 (same as in input) and orthometric height in meters [m] in LHN95

4.10.6 Examples

http://geodesy.geo.admin.ch/reframe/besseltolhn95?easting=600000&northing=200000&altitude=549.4&format=json

answer:
{"easting": "600000.0", "northing": "200000.0", "altitude": "550.0335999965667"}

http://geodesy.geo.admin.ch/reframe/besseltolhn95?easting=2600000&northing=1200000&altitude=549.4&cb=myfunction

answer:
myfunction({ "type": "Point", "coordinates": [2600000.0, 1200000.0, 550.0335999965667] });

4.11 LN02 to Bessel (ellipsoidal)

4.11.1 Use

Convert Swiss usual heights LN02 (leveling network) into ellipsoidal heights (on Bessel 1841).

4.11.2 Base URI

http://geodesy.geo.admin.ch/reframe/ln02tobessel

4.11.3 Arguments

Required:

easting	East coordinate in meters [m] in LV03 or LV95
northing	North coordinate in meters [m] in LV03 or LV95
altitude	Usual height in meters [m] in LN02

Optional:

- **format**
  - "**json**" JSON data representation
  - "**geojson**" GeoJSON data representation, used by default if argument not specified
- **cb** Callback function name

4.11.4 HTTP status codes

200	The transformation **succeeded**
400	The transformation **failed**: insufficient or incorrect arguments

4.11.5 Output data

JSON object (if argument "format"="json"):

easting	East coordinate in meters [m] in LV03 or LV95 (this transformation does not affect the position, which will be equal as in input)
northing	Nord coordinate in meters [m] in LV03 or LV95 (this transformation does not affect the position, which will be equal as in input)
altitude	Ellipsoidal height in meters [m] on Bessel

or GeoJSON object (if argument "format"="json"):

type	"Point"
coordinates	3D vector containing East and North coordinates in meters [m] in LV03 or LV95 (same as in input) and ellipsoidal height in meters [m] on Bessel

4.11.6 Examples

http://geodesy.geo.admin.ch/reframe/ln02tobessel?easting=600000&northing=200000&altitude=550.0&format=json

answer:
{"easting": "600000.0", "northing": "200000.0", "altitude": "549.3243940114655"}

http://geodesy.geo.admin.ch/reframe/lhn95tobessel?easting=2600000&northing=1200000&altitude=550&cb=myfunction

answer:
myfunction({ "type": "Point", "coordinates": [2600000.0, 1200000.0, 549.3664000034332] });

4.12 Bessel (ellipsoidal) to LN02

4.12.1 Use

Convert Swiss ellipsoidal heights (on Bessel 1841) into usual heights LN02 (leveling network).

4.12.2 Base URI

http://geodesy.geo.admin.ch/reframe/besseltoln02

4.12.3 Arguments

Required:

- **easting** East coordinate in meters [m] in LV03 or LV95
- **northing** North coordinate in meters [m] in LV03 or LV95
- **altitude** Ellipsoidal height in meters [m] on Bessel

Optional:

- **format**
  - “**json**” JSON data representation
  - “**geojson**” GeoJSON data representation, used by default if argument not specified
- **cb** Callback function name

4.12.4 HTTP status codes

- **200** The transformation **succeeded**
- **400** The transformation **failed**: insufficient or incorrect arguments

4.12.5 Output data

JSON object (if argument “format”=”json”):

- **easting** East coordinate in meters [m] in LV03 or LV95 (this transformation does not affect the position, which will be equal as in input)
- **northing** Nord coordinate in meters [m] in LV03 or LV95 (this transformation does not affect the position, which will be equal as in input)
- **altitude** Usual height in meters [m] in LN02

or GeoJSON object (if argument “format”=”json”):

- **type** “Point”
- **coordinates** 3D vector containing East and North coordinates in meters [m] in LV03 or LV95 (same as in input) and usual height in meters [m] in LN02

4.12.6 Examples

http://geodesy.geo.admin.ch/reframe/besseltoln02?easting=600000&northing=200000&altitude=549.4&format=json

answer:
{"easting": "600000.0", "northing": "200000.0", "altitude": "550.0756028031193"}

http://geodesy.geo.admin.ch/reframe/besseltoln02?easting=2600000&northing=1200000&altitude=549.4&cb=myfunction

answer:
myfunction({ "type": "Point", "coordinates": [2600000.0, 1200000.0, 550.0756028031193] });

Wabern, April 2016

Federal Office of Topography swisstopo
Domain of geodesy
Geodetic developments and contracts

Jérôme Ray