CRS extensions for spatial APIs

Public session 3: 2021-12-02

Overview

- Introduction
- Goal
- Findings
- Encodings

Introduction – GIS Specialisten

- Geo-ICT with 13 years of experience
- 20+ colleagues from 7 different countries
- Located in Utrecht
- Geo-ICT services:
    - Creation of tailor made Spatial Data Infrastructures
    - Operation of SDI's
    - GIS analysis
- Product development: Geoservice.Cloud
    - Gis-as-a-Service
    - Hosting, services, analyses and online presentation in Web-GIS
    - Build on Open Source, runs in the cloud
- Working through co-creation

Goal

1. Demonstrate how one can serve spatial data in both RD and ETRS89 CRS in such a way that the implementation is OGC API Features – Part 2: CRS-compliant.
2. Demonstrate how the effort, needed to serve spatial data in both RD and ETRS in accordance with OGC API Features – Part 2: CRS, relates to the effort, needed to serve the same spatial data in those CRSs in accordance with the Dutch API strategy (using content negotiation for CRS).
3. Demonstrate how easy or difficult it is to adapt an existing API that uses content negotiation for CRS, to a version that follows the OGC API Features – Part 2: CRS specification, or to a version that supports both mechanisms.
4. Demonstrate, by using at least one client, how users can request data in a specific CRS from a.) an API that implements the OGC API Features – Part 2: CRS specification and b.) an API that implements the Dutch API Strategy mechanism with content negotiation for CRS.
5. Demonstrate how your favorite (i.e. useful / user-friendly / original / popular) existing API that supports multiple CRSs, can be converted into a version that is OGC API Features – Part 2: CRS-compliant.

Serve data in multiple CRS-es

Demonstrate how one can serve spatial data in both RD and ETRS89 CRS in such a way that the implementation is OGC API Features – Part 2: CRS-compliant.

- Demo viewer available soon
- API supports multiple CRS-es:
  - WGS84
  - ETRS89
  - RD New
- Not fully compliant
  - Only relevant part of specs implemented for now
  - Relative links vs absolute links

Content negotiation

Demonstrate how the effort, needed to serve spatial data in both RD and ETRS in accordance with OGC API Features – Part 2: CRS, relates to the effort, needed to serve the same spatial data in those CRSs in accordance with the Dutch API strategy (using content negotiation for CRS).

- CRS in Headers vs URL Encoding

6.3.2. Parameter crs

Requirement 10	/req/crs/fc-crs-definition

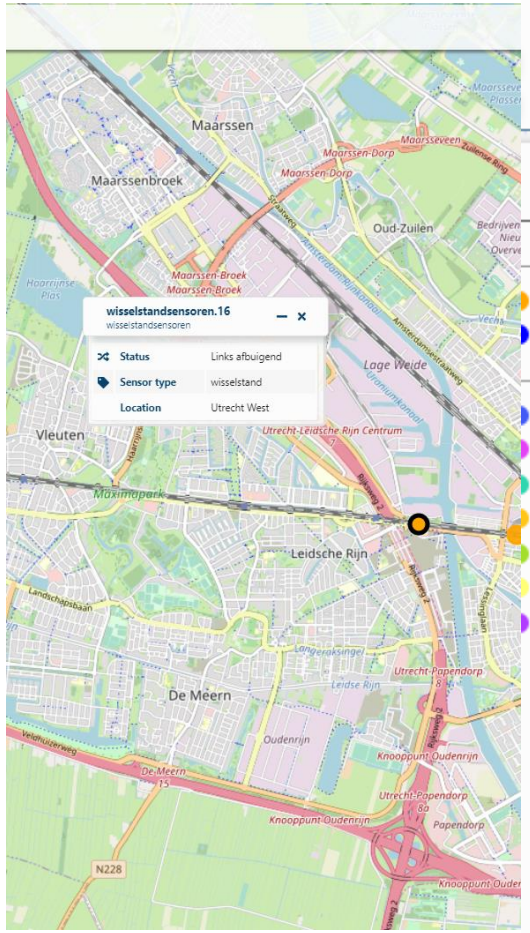
Each GET request on a 'features' or 'feature' resource SHALL support a query parameter named `crs` with the following characteristics:
 	
`name: crs`
`in: query`
`required: false`
`schema:`
`  type: string`
`  format: uri`
`style: form`
`explode: false`	

Requirement 11	/req/crs/fc-crs-valid-value
A	If the value of the `crs` parameter is not one of the CRS identifiers from the list of supported CRS identifiers, then the server SHALL respond with the HTTP status code 400.
B	The list of supported CRS identifiers is found in the collection object using the `crs` property.

API-40: Pass the coordinate reference system (CRS) of the request and the response in the headers

The coordinate reference system (CRS) for both the request and the response are passed as part of the request headers and reponse headers. In case this header is missing, send the HTTP status code `412 Precondition Failed`.

Demo



Name

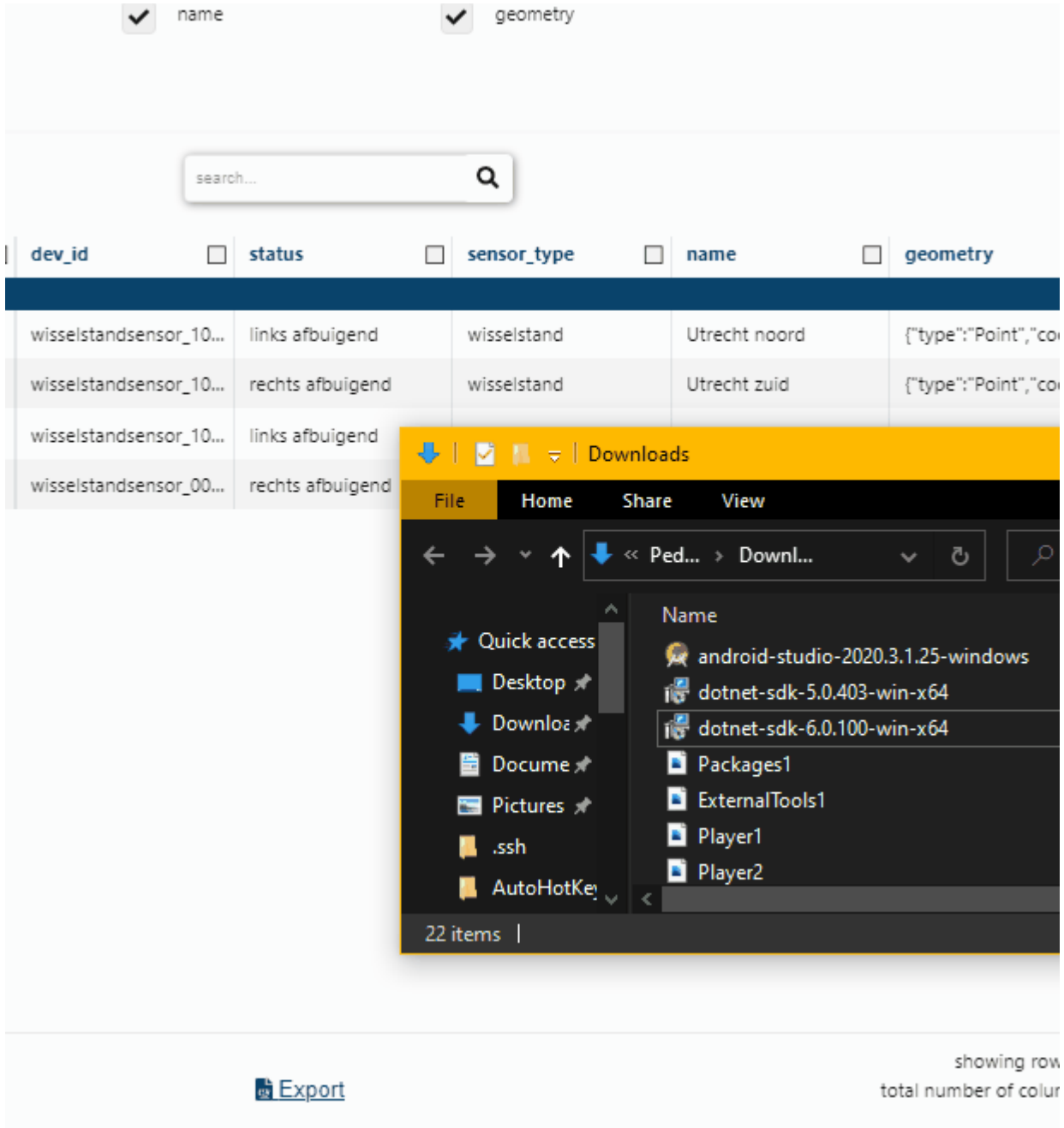
- layers.json
- collections
- ows?service=WMS&version=1.3.0...
- ?EIO=4&transport=polling&t=Npc...
- ?EIO=4&transport=polling&t=Npc...
- ?EIO=4&transport=polling&t=Npc...
- ?EIO=4&transport=polling&t=Npc...
- ?EIO=4&transport=polling&t=Npc...
- ?EIO=4&transport=polling&t=Npc...
- ows?service=WFS&version=1.0.0&r...
- ows?service=WFS&version=1.0.0&r...
- ows?service=WFS&version=1.0.0&r...
- v4_0?service=WFS&version=2.0.0&...
- wfs?service=WFS&version=1.1.0&r...
- ows?service=WFS&version=1.0.0&r...
- ows?service=WFS&version=1.0.0&r...
- items
- item
- v4_0
- wms?service=WMS&version=1.3.0...
- ows?service=WMS&version=1.3.0...
- ows?service=WMS&version=1.3.0...
- ows?service=WMS&version=1.3.0...
- ows?service=WMS&version=1.3.0...

http://localhost:3001/collections/wisselstandsensoren/items

Headers | Preview | Response | Initiator | Timing

▼{bbox: [4.44973900093371, 51.931170004589, 5.10590200161786, 52.1124090047
▼bbox: [4.44973900093371, 51.931170004589, 5.10590200161786, 52.112409004
0: 4.44973900093371
1: 51.931170004589
2: 5.10590200161786
3: 52.1124090047034
▼crs: {type: "name", properties: {name: "http://www.opengis.net/def/crs/O
▼properties: {name: "http://www.opengis.net/def/crs/OGC/1.3/CRS84"}
name: "http://www.opengis.net/def/crs/OGC/1.3/CRS84"
type: "name"
▶features: [{id: "wisselstandsensoren.2",…}, {id: "wisselstandsensoren.4"
▼links: [{href: "/collections/wisselstandsensoren/items", rel: "self", ty
▼0: {href: "/collections/wisselstandsensoren/items", rel: "self", type:
href: "/collections/wisselstandsensoren/items"
rel: "self"
title: "this document"
type: "application/geo+json"
numberMatched: 17
numberReturned: 17
▼storageCrs: {type: "name", properties: {name: "http://www.opengis.net/de
▼properties: {name: "http://www.opengis.net/def/crs/EPSG/0/28992"}
name: "http://www.opengis.net/def/crs/EPSG/0/28992"
type: "name"
-03T14:20:05.809Z"
totalFeatures: 17

Demo


name
geometry
search...
dev_id
status
sensor_type
name
geometry
wisselstandsensor_10...
links afbuigend
wisselstand
Utrecht noord
wisselstandsensor_10...
rechts afbuigend
wisselstand
Utrecht zuid
wisselstandsensor_10...
links afbuigend
wisselstandsensor_00...
rechts afbuigend
Downloads
File
Home
Share
View
Ped...
Downl...
Quick access
Desktop
Downloa
Docume
Pictures
.ssh
AutoHotKe
Name
android-studio-2020.3.1.25-windows
dotnet-sdk-5.0.403-win-x64
dotnet-sdk-6.0.100-win-x64
Packages1
ExternalTools1
Player1
Player2
22 items
Export
showing row
total number of colur

Encodings

- GeoJSON
- JSON-FG

Best practices

- Geonovum Github: https://github.com/Geonovum/testbed-spatial-APIs

And further

- Finishing the demo
- Share hurdles and experiences of developing the API

Questions