



OGC API - Features: Enhanced usability for existing WFS services

Inspire Helsinki 2019 Workshop

Overview

- Introduction and Context
 - OGC API Features
 - OGC API Features in INSPIRE / MIG-T
 - Idproxy
- Idproxy Hands-on
 - Setting up APIs
 - Configuring APIs
 - Browsing the data and using the API in a Browser
 - Using the API in other clients
 - Extensions
- Questions and Discussion



OGC API – Features – Part 1: Core

OGC API – Features

- **OGC API standards** define modular API building blocks to spatially enable Web APIs in a consistent way
- **OGC API – Features** specifies the fundamental API building blocks for interacting with features (query, create and modify features on the Web)
- Started as a revision of the OGC Web Feature Service standard that
 - proposes a modernized service architecture,
 - follows the current Web architecture and the (Spatial) Data on the Web Best Practices,
 - leverages the OpenAPI specification,
 - specifies modular building blocks for fine-grained access to spatial data that can be used in data APIs,
 - acknowledges the importance of HTML,
 - removes the dependency on XML,
 - has a focus on the developer experience,
 - is developed in an open process doing all work in a public GitHub repository, early implementations, in-depth validation, slow release
- Version 1.0.0 of Part 1 (Core) approved by OGC in September, published last week
 - Few changes since the first stable draft in April 2018

OGC API – Features – Part 1: Core Resources of the Web API

Table 1. Overview of resources, applicable HTTP methods and links to the document sections

Resource	Path	HTTP method	Document reference
Landing page	/	GET	7.2 API landing page
Conformance declaration	/conformance	GET	7.4 Declaration of conformance classes
Feature collections	/collections	GET	7.13 Feature collections
Feature collection	/collections/{collectionId}	GET	7.14 Feature collection
Features	/collections/{collectionId}/items	GET	7.15 Features
Feature	/collections/{collectionId}/items/{featureId}	GET	7.16 Feature

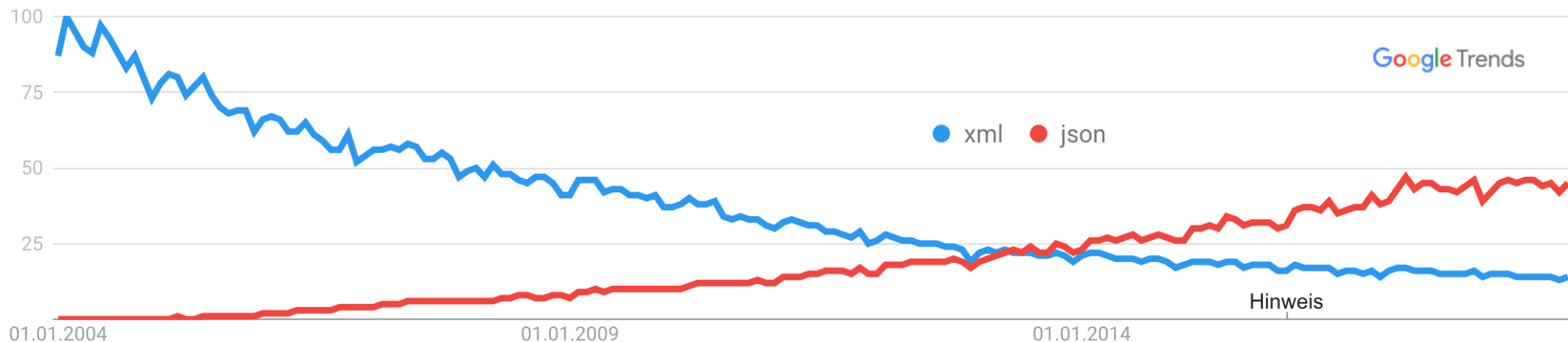
information
about the API

a dataset with a
sub-division into
named collections
of features

the features

<http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#tldr>

Not tied to any particular encoding, GeoJSON currently recommended (where applicable)



OGC API standards support two approaches how applications can use an API

1. Clients that know the resources specified in the OGC API standards navigate them based on the provided information and links

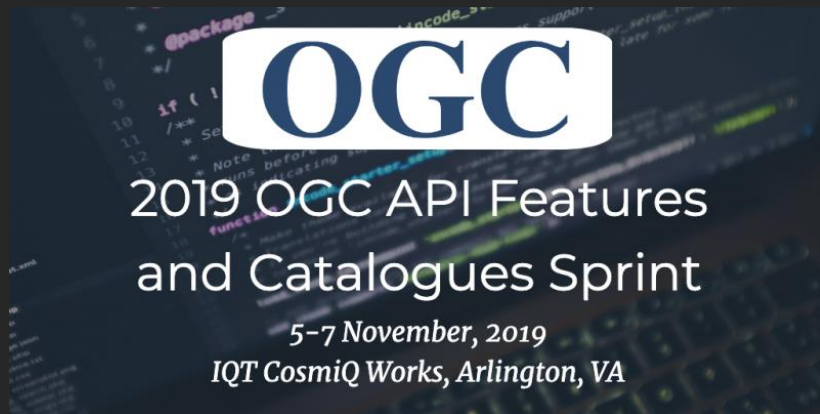
- Start at the landing page or another resource, analyse the information, follow links; repeat
- The OpenAPI definition may be used to determine details e.g. on filter parameters, but this may not be necessary depending on the application
- Clients can connect to multiple APIs as long as they implement the OGC API standard(s)

2. Study the OpenAPI definition and develop client applications against it

- Familiarity with OpenAPI expected, but no previous knowledge of the OGC standards required
- Developers will study the API definition/documentation and play with the API to understand it
- Supports code-generation based on the API definition
- It should not be necessary to study the OGC API standards
- “Time to first successful call”

What's next?

- Part 2: Support for Coordinate Reference Systems beside WGS 84
- More query / filter / sorting capabilities



- More spatial resource types in various stages of maturity / discussion in OGC or in the wider community
 - Tiles (spatial partitioning)
 - Maps
 - Styles (symbols/rules to render maps)
 - Processes (spatial data processing)
 - Coverages (e.g. elevation data, satellite images, etc.)
 - Routes
- Data updates using POST, PUT, DELETE, PATCH
- Building blocks for event-driven APIs
 - Change-only updates

Implementation example using Idproxy

