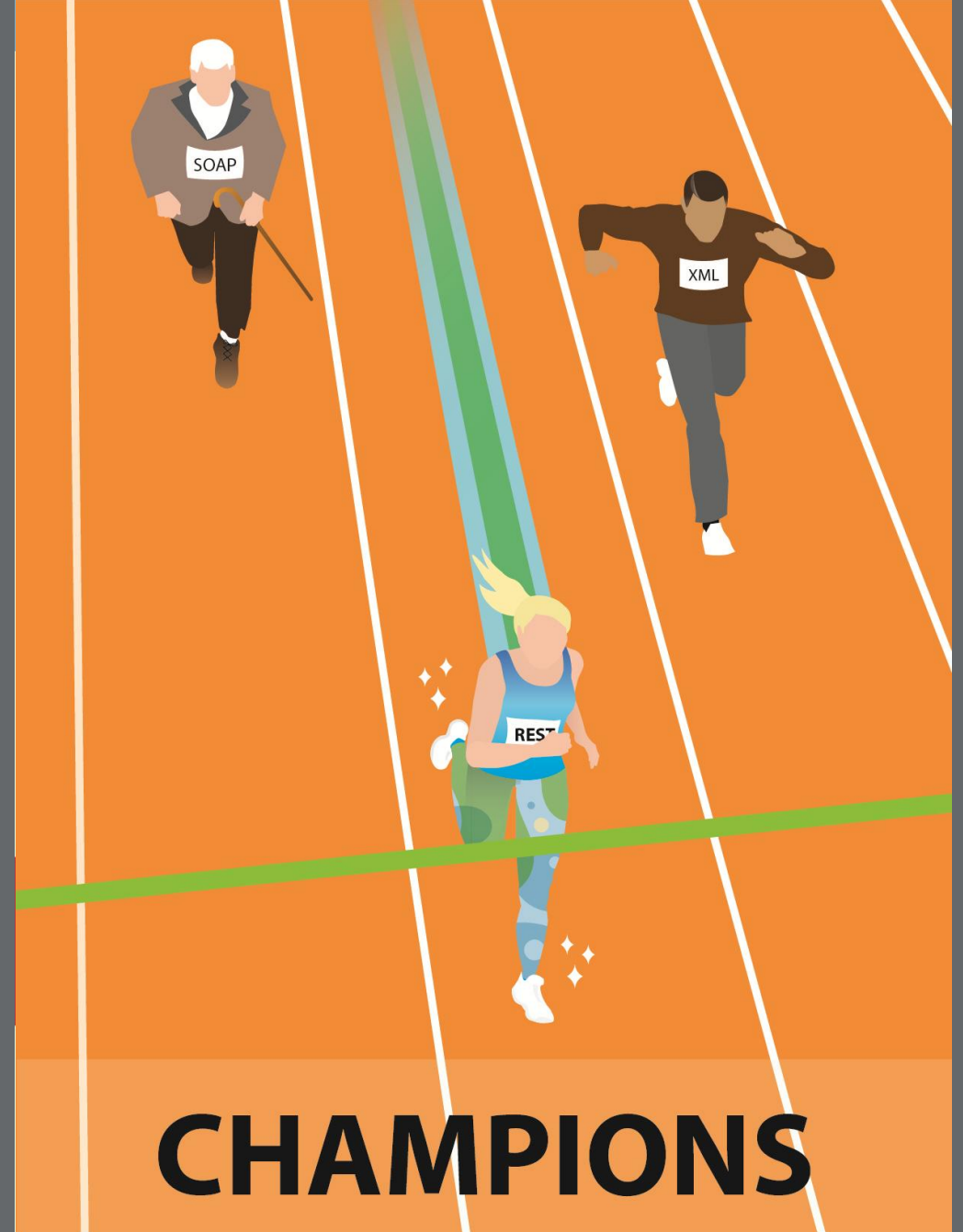


Zelf aan de slag

OGC API Features Server

OGC API Features (web)client



CHAMPIONS

Follow the OGC API Standards

nodejs, express – javascript

Node.js is een [open source](#) en [multiplatform](#) JavaScript-runtime-omgeving waarmee [JavaScript](#) buiten een browser uitgevoerd kan worden. Met Node.js kunnen ontwikkelaars JavaScript gebruiken voor [command-line-tools](#) en [server-side scripting](#). Het resultaat is een "JavaScript overal"-paradigma dat [webapplicatie-ontwikkeling](#) verenigt rond een enkele programmeertaal, in plaats van verschillende talen voor server- en client-side scripts.^[3]

Express 4.19.2

Fast, unopinionated, minimalist
web framework for [Node.js](#)

```
$ npm install express --save
```

🔊 Express 5.0 beta documentation is now available.

The beta [API documentation](#) is a work in progress. For information on what's in the release, see the Express [release history](#).

Node.js



Ontwikkelaar(s)	Ryan Lienhart Dahl, Node.js Developers ↗, Joyent
Uitgebracht	27 mei 2009 (15 jaar)
Recentste versie	22.0.0 (24 april 2024) ^[1] ✎
Onderhoudsversie	18.14.2 (LTS) ^[2] (2023-02-21)
Status	Actief
Besturingssysteem	Linux, Microsoft Windows, BSD, macOS
Geschreven in	C , C++ , JavaScript
Categorie	Softwareplatform
Licentie(s)	MIT
Versiebeheer	Officiële broncode ↗
Website	(ar) (de) (en) Projectpagina ↗

Portaal 

Informatica
Vrije software

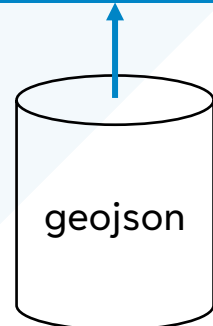
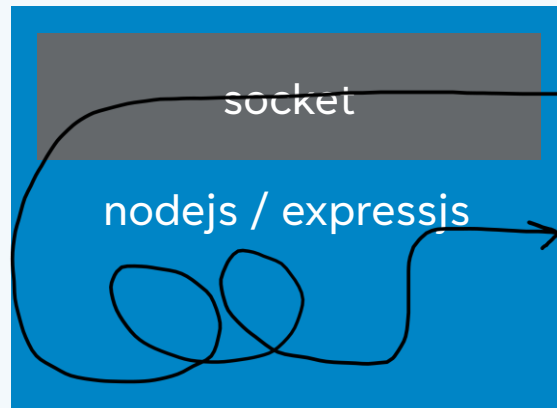
OGC API Features – ‘Routes’

- Landing page <https://api.pdok.nl/lv/bgt/ogc/v1>
- OAS <https://api.pdok.nl/lv/bgt/ogc/v1/api>
- Collections <https://api.pdok.nl/lv/bgt/ogc/v1/collections>
- Collection-id <https://api.pdok.nl/lv/bgt/ogc/v1/collections/bak>
- Items <https://api.pdok.nl/lv/bgt/ogc/v1/collections/bak/items>
- Item-id: <https://api.pdok.nl/lv/bgt/ogc/v1/collections/bak/items/5d394ef5-6a5d-5011-a729-29def1c51dd9>

Parameters:

- ?f=json or ?f=html
- ?crs vb: ?crs=http://www.opengis.net/def/crs/EPSG/0/28992
- ?bbox vb: ?bbox=160.6,-55.95,-170,-25.89

‘Architectuur’



```
{ "type": "FeatureCollection", "features": [ { "type": "Feature", "geometry": { "type": "Polygon", "coordinates": [ [ [ 4.836904719830233, 51.71261186219708 ], [ 4.842367931828861, 51.67965404535827 ], [ 4.781071321809747, 51.66390211812952 ], [ 4.787505939028108, 51.63704801116956 ], [ 4.731388847030403, 51.64299386821649 ], [ 4.699059657817242, 51.638073190658574 ], [ 4.684902977273319, 51.666366490887185 ], [ 4.64513658535018, 51.71906732347356 ], [ 4.676304223904792, 51.724923649009945 ], [ 4.698553145800539, 51.72199204842582 ], [ 4.733650891305746, 51.73650736658969 ], [ 4.764199375929489, 51.759857403256646 ], [ 4.853126170313312, 51
```

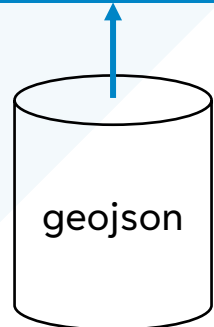
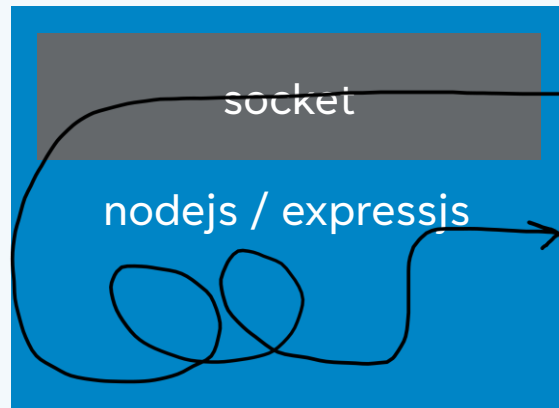
Bruno

WebBrowser

QGIS, Esri, ...

curl, ...

‘Architectuur’



Bruno

WebBrowser

QGIS, Esri, ...

curl, ...

Zelf experimenteren

Wat gaan we doen?

- Een minimale basis OGC API Features server bouwen
- NodeJS
- Visual Studio Code
- Bruno (of andere API Client)
- Zie: <https://github.com/Geonovum/ogc-api-workshops/tree/main/04%20OGC-API-Features/handson>
- Een (web)client gebruiken om met code een OGC API Features endpoint te bevragen
 - Python
 - Leaflet

Extra: via Pygeoapi

1. Installeer Pygeoapi
2. Pas [config.yml](#) aan
3. Maak Open API Specificatie aan (OAS) in cmd:

```
pygeoapi openapi generate /path/to/my-pygeoapi-config.yml >  
/path/to/my-pygeoapi-openapi.yml
```
4. Start de server: `pygeoapi serve`
5. Zie ook <https://docs.pygeoapi.io/en/latest/>

Extra: via Pygeoapi en Docker

1. [Installeer Docker](#) en start docker desktop als admin
2. Maak een image van pygeoapi (zoek laatste versie via Search in blauwe balk). Als het goed is heet die “**geopython/pygeoapi**”
3. Pas [config.yml](#) aan op je host (=eigen locatie)
4. Type in de cmd als admin:
`docker run -p 5000:80 -v $(pwd)/my.config.yml:/pygeoapi/local.config.yml -it geopython/pygeoapi`
‘\$(pwd)/my.config.yml’ is daarbij het pad naar je config.yml uit stap 3
5. Zie ook <https://docs.pygeoapi.io/en/latest/>