

Les 7: Vector lagen op maat

Openlayers ondersteunt allerlei standaard tekstformaten voor de opbouw van Vectorlagen, zoals GeoJSON, WKT, KML en GML. Veel webservices volgen om allerlei redenen deze structuren niet of maar gedeeltelijk. In dat geval zal je zelf het bestand moeten parseren in javascript een omzetten naar Vectorlaag-bron.

In deze oefening zullen we de output van GIPOD public api (<http://gipod.api.agiv.be/#!index.md>) gebruiken.

Zo kunnen via de GIPOD-api de eerste 1000 werkopdrachten van vandaag tot binnen 30-dagen ophalen (default indien geen start-en end-date wordt opgegeven) voor de provincie Limburg. We geven ook de url-parameter `crs=31370` op, om de gegevens in Lambert 72 te krijgen, zodat we niet moeten herprojecteren.

```
http://gipod.api.agiv.be/ws/v1/workassignment?  
crs=31370&limit=1000&province=limburg  
(http://gipod.api.agiv.be/ws/v1/workassignment?  
crs=31370&limit=1000&province=limburg)
```

Het resultaat is een JSON-array van objecten met de volgende inhoud:

```
[  
  {  
    gipodId: 123137,  
    owner: "Aquafin nv",  
    description: "3840 - Verbindingsriolering Jesserenstraat - deel Broekstraat -  
rioleringwerken",  
    startDateTime: "2012-03-12T00:00:00",  
    endDateTime: "2015-03-02T00:00:00",  
    importantHindrance: false,  
    coordinate: {  
      coordinates: [ 222762.38,166568.57],  
      type: "Point",  
      crs: {  
        type: "name",  
        properties: {name:"urn:ogc:def:crs:EPSG::31370"}  
      }  
    },  
    detail: "http://gipod.api.agiv.be/ws/v1/workassignment/123137?crs=Lambert72",  
    cities: ["Borgloon","Tongeren"]  
  },  
  ...  
]
```

Het integer veld *GipodId* is de unieke ID, het veld *coordinate* bevat het geometrie-object. De andere velden zijn strings behalve *cities*, dat een array van strings is en *importantHindrance* dat een boolean is. Je kunt deze data ophalen via een *ajax* request in JQuery:

```
$.ajax({
  url: "http://gipod.api.agiv.be/ws/v1/workassignment",
  dataType : "json"
  data: {
    province: "Limburg",
    crs: 31370,
    limit: 1000 } }).done(
  function(data){
    //doe iets met de data
    alert(data[0].description);
  });
```

Om deze data aan de kaart toe voegen hebben een laag en lege databron nodig waarin we deze gegevens zullen toevoegen.

```
var vectorSource = new ol.source.Vector({projection: 'EPSG:31370'});
var vectorLayer = new ol.layer.Vector(
  {style: new ol.style.Style({
    image: new ol.style.Circle({
      radius: 5,
      fill: new ol.style.Fill({
        color: 'rgba(255, 0, 0, 0.5)'})
    })
  })},
  source: vectorSource
});
```

Om een feature Object te maken heb je gewoon JSON-object nodig met een Openlayers geometrie object aan gekoppeld. Omdat de geometrie in de gipod API wordt opgeslagen als geojson object kunnen we de format-lezer uit openlayers gebruiken om dit om te zetten een openlayer geometrie object. Als we alle gegevens hebben uitgelezen naar een array van features kunnen we deze aan de laagbron toevoegen.

```
function gipodParser(data){
  var geojsonReader = new ol.format.GeoJSON();
  var features = [];
  $.each(data, function(index, row){
    var geom = geojsonReader.readGeometry(row.coordinate);
    delete row.coordinate; //niet meer nodig
    row.geometry = geom; //voeg ol geometrie toe
    var feature = new ol.Feature( row );
    features.push( feature );
  });
  vectorSource.addFeatures(features);
}
```