

# Les 4: Vectorlagen

Vectoriele gisdata kan worden binnen gehaald als gestructureerde tekst in plaats van een afbeelding.

## Types vectorlagen

Er bestaan enkele standaard formaten die openlayers onmiddellijk van url kan openen. Het gaat onder meer om Geojson (<http://geojson.org/>), Topojson, GML, GPX en KML. Vooral Geojson is een veel gebruikt formaat in WebGIS. Je moet ook steeds het gewenste coördinaatsysteem voor de vectorlaag opgeven, normaal gezien is dit de kaartprojectie.

```
var geojsonBron = new ol.source.GeoJSON({
  projection: map.getView().getProjection(),
  url: 'data/gemeenten.geojson'
});
```

Dit soort data kan je ook, zelf stylen. Afhankelijk van de het datatype moet je andere types van stijlen gebruiken. Het object `new ol.style.Style` is een collectie van stijlen.

**Een punt** wordt meestal gevisualiseerd met een icon of marker, een kleine afbeelding.

```
var marker = new ol.style.Icon({
  anchor: [0.5,1],
  src: 'data/icon.png'
});
var stijl = new ol.style.Style({
  image: marker
});
```

**Een lijn** heeft een breedte en kleur.

```
var stijl = new ol.style.Style({
  stroke: new ol.style.Stroke({
    color: 'green',
    width: 1
  })
});
```

**Een polygoon** kan ook een opvulling hebben.

```
var stijl = new ol.style.Style({
  stroke: new ol.style.Stroke({
    color: 'blue',
    width: 2
  }),
  fill: new ol.style.Fill({
    color: 'rgba(0, 0, 255, 0.1)'
  })
});
```

Met de databron en een stijl kan je de Vectorlaag samenstellen.

```
var vectorLaag = new ol.layer.Vector({
  source: geojsonBron,
  style: stijl
});
```

Je kun een stijl ook toewijzen als callback functie op de feature, dan kan je op basis van de eigenschappen van de feature de inkleuring doen of zelf een label toewijzen. In het onderstaande voorbeeld zie hoe je vanaf resolutie 5 de bron gemeenten, de features worden gelabeld met het attribuut naam.

```

var vectorLaag = new ol.layer.Vector({
  source: gemeenten,
  style: function(feature, resolutie) {
    var stijl;
    if( resolutie >= 5 ) {
      stijl = new ol.style.Style({
        text: new ol.style.Text({
          text: feature.get('naam'),
          fill: new ol.style.Fill({
            color: '#111'
          })
        }),
        image: new ol.style.Circle({
          radius: 7 + (size * 0.7),
          fill: new ol.style.Fill({ color: 'rgba(255, 255,
0, 0.5)' }),
          stroke: new ol.style.Stroke({color: 'red', width:
1})
        })
      });
    }
    else {
      stijl = new ol.style.Style({
        image: new ol.style.Circle({
          radius: 7 + (size * 0.7),
          fill: new ol.style.Fill({ color: 'rgba(255, 255,
0, 0.5)' }),
          stroke: new ol.style.Stroke({color: 'red', width:
1})
        })
      });
    }
    return stijl;
  }
});

```

## Identificeren

Je kunt ook op vectorlagen ook een identificeren via een muis klik. Hier zien hoe je een div met id="info" kunt bevolken met de waarde 'naam' na een muisklik op de kaart:

```

var displayFeatureInfo = function(pixel) {
  var feature = map.forEachFeatureAtPixel(pixel, function(feature, layer) {
    return feature;
  });
  var info = $('#info');
  if (feature) {
    info.html( feature.get('naam'));
  } else {
    info.html('');
  }
};
map.on('click', function(evt) {
  displayFeatureInfo(evt.pixel);
});

```

Een voordelen van vectorlagen is de mogelijkheid om bij een muisbeweging over de kaart al iets met de data te doen.

```

$(map.getViewport()).on('mousemove', function(evt) {
  var pixel = map.getEventPixel(evt.originalEvent);
  displayFeatureInfo(pixel);
});

```

## Vectorlagen bewerken

Vectorlagen kan je ook bewerken. Dit kan je door middel van de *interaction* objecten, die je toevoegt aan het map-object.

Met *ol.interaction.Draw* kan aan een laagbron nieuwe objecten toevoegen door ze op de kaart in te tekenen.

```

var draw = new ol.interaction.Draw({
  source: geojsonBron,
  type: 'Point'
});
map.addInteraction(draw);

```

Om een laag te wijzingen of verwijderen moet je eerst het gewenste object selecteren. Let op standaard gebruiken zowel draw als select de *singlemouse* als trigger, ze kunt ze niet tegelijk actief hebben. Dit kan met de *ol.interaction.select* interactie. In dit object geef je aan met optie layers welke lagen selecteerbaar zijn. Met optie multi geef je aan of meer dan 1 feature te gelijk wilt kunnen selecteren. Je kunt ook de stijl van de geselecteerde features aangegeven.

```

var select = new ol.interaction.Select({
  layers: [vectorLaag],
  multi: true ,
  style: new ol.style.Style({
    image: new ol.style.Icon({
      anchor: [0.5,1],
      src: '/images/selection-icon.png'
    })
  })
});
map.addInteraction(select);

```

De selectie is een collectie van features. Op deze selectie kan je dan allerlei bewerkingen uitvoeren, zoals deze objecten verwijderen uit de laag. Via de methode `getFeatures` van de interactie krijg je toegang tot de selectie.

```

select.getFeatures().forEach( function( feat, index , col ){
  vectorSource.removeFeature( feat )
});
select.getFeatures().clear();

```

Geselecteerde features kan je bewerken met de *ol.interaction.modify* interactie. Voor een punten laag is dit enkel verplaatsen, aan polygonen kan je ook vertices toevoegen of verwijderen.

```

var modify = new ol.interaction.Modify({
  features: select.getFeatures()
});
map.addInteraction(modify);

```

Om wijzingen en nieuwe features in vectorlagen op te slaan, moet je dit wegschrijven naar je server. Dit doe je meestal met webservice die je zelf maat schrijft, al bieden sommige providers zoals cartoDB of ESRI arcgis server wel webservices aan die optioneel ook insert en update query's tegen je data kunnen uitvoeren.

Probeer het zelf (tryit?file=examples/OL3\_LES4\_vector.html&msg=Vervang de laagbron met de file /resource/zendmast.geojson, gebruik de iconen selection-icon.png en marker-icon.png. Voeg ook de attributen voor GEMEENTE en ADRES toe aan statusbalk)