External Dataservices

Bing Maps

The `styles` option can be any combination of `Road`, `AerialWithLabels` and `Aerial`. Multiple services can be defined with different keys and assigned to different groups. Tipically the URL is: `http://dev.virtualearth.net`

Options example:

```
  {
    "bingMapsApiKey": "<insert your Bing Maps application key here>",
    "styles": ["Road", "AerialWithLabels" , "Aerial"]
  }
```

Web Map Service (WMS)

For more information about the options parameter see WebMapServiceImageryProvider (https://cesiumjs.org/Cesium/Build/Documentation/WebMapServiceImageryProvider.html), not all Cesium parameters are supported.

- `isBaseMap` set to true shows the layer in the map selection menu.
- `parameters` allows setting extra parameters in the query part of the url.

Options example:

```
{
"isBaseMap": true,
"parameters": {
    "Username": "<username>",
    "Password": "<password>"
},
"layers": "<layer1>,<layer2>",
"credit": "© fake 2017"
}
```

Web Map Tile Service (WMTS)

For more information about the options parameter see WebMapTileServiceImageryProvider (https://cesiumjs.org/Cesium/Build/Documentation/WebMapTileServiceImageryProvider.html), not all Cesium parameters are supported.

Currently the WMTS has to be `epsg:3857` with a global bounding box. Named levels are not supported.

- `isBaseMap` set to true shows the layer in the map selection menu.
- `parameters` allows setting extra parameters in the query part of the url.

The url is a pattern that should use these placeholders that are substituted upon URL generation by Cesium. The placeholders can be used both in the url's parameters and in the url itself.

Substitution Parameter	Raplaced by
`{Layer}`	`layer`
`{TileMatrixSet}`	`tileMatrixSetID`
`{Style}`	`style`
`{TileMatrix}`	this is the level number
`{TileCol}`	`-`
`{TileRow}`	`-`

Options example:

```
{
"isBaseMap": true,
"credit": "© fake 2017",
"parameters": {
    "username": "<username>",
    "password": "<password>",
    "layer": "{Layer}",
    "style": "{Style}",
    "tilematrixset": "{TileMatrixSet}",
    "Service": "WMTS",
    "Request": "GetTile",
    "Version": "1.0.0",
    "Format": "image/jpeg",
    "TileMatrix": "{TileMatrixSet}:{TileMatrix}",
    "TileCol": "{TileCol}",
    "TileRow": "{TileRow}"
},
"layer": "<layer name>",
"style": "default",
"format": "image/jpeg",
"tileMatrixSetID": "<TileMatrixSet>",
"tileWidth": 256,
"tileHeight": 256,
"maximumLevel": 20,
"minimumLevel": 0
}
```

OpenLayers External Dataservices

The OpenLayers support both Arcgis and WFS services. The configuration needed is the following:

- Name: The name of the service.
- Url: The URL of the service:
  - For Arcgis this should be the url that lists all the services.
  - For WFS this should be the url for the WFS service.

- Type: This should be OpenLayers

ArcGis

Parameter	Description
`serviceProvider`	The name of the service Provider, in this case it should be `Arcgis`
`serviceName`	The name of the arcgis service
`serviceType`	The type of service, e.g. 'FeatureServer'
`tokenUrl`	The URL for requesting a token. This is only necessary if authenticate is set to true
`authenticate`	Whether the service needs authentication. Default: false
`layerConfig`	Object of layerConfig properties.

layerConfig Properties:

Property	Description
`layerId`	The id of the layer
`projection`	The projection requested to the server. Default: `urn:x-ogc:def:crs:EPSG:4326`
`returnGeometry`	Whether the server should return geoemtry. Nothing will be visible if set to false. Default: true
`spatialRel`	How the server should intersect for filtering (e.g. with bounding box). Default: esriSpatialRelIntersects
`rollbackEdits`	Whether the server should rollback edits if one fails. Default: true
`whereQuery`	A filter SQL style query for the data
`hasZ`	Whether the layer is 3D or not. If not set this will be retrieved in the setup fase
`outputFormat`	The default outputformat when requesting features. Recommended: geojson`

Example:

```
{
  "serviceProvider": "ArcGis",
  "serviceName": "bedum_street_lights_development",
  "serviceType": "FeatureServer",
  "tokenUrl": "http://webplayer.horus.local/arcgis/tokens/",
  "authenticate": false,
  "layerConfig": {
    "layerId": 2,
    "projection": "urn:x-ogc:def:crs:EPSG:4326",
    "outputFormat": "json"
  }
}
```

WFS

Parameter	Description
`serviceProvider`	The name of the service Provider, in this case it should be "'WFS'"
`authenticate`	Whether the service needs authentication. Default: false"
`layerConfig`	Object of layerConfig properties.

layerConfig Properties:

Property	Description
`service`	The service name. Default: 'WFS'
`version`	The WFS version. For now, only 1.0.0 is supported. Default 1.1.0
`featureNS`	The feature namespace.
`featurePrefix`	The feature prefix.
`featureName`	The feature layer.
`geometryName`	The name of the geometry attribute that will be drawn. This will be automatically retrieved from the server's capabilities. If there is more than one, it will retrieve the first one.
`hasZ`	Whether the layer is 3D or not. Default: true
`sortBy`	One or more attributes to sort the features when requested.
`propertyName`	A list of properties to filter the features by.
`maxFeatures`	The maximum number of features per query.
`projection`	The projection that is requested to the server. Default: urn:x-ogc:def:crs:EPSG:4326

Property	Description
`outputFormat`	The default outputformat when requesting features. Recommended: geojson. This will be retrieved from the server, if not present. Gives priority to geojson.

Example:

```
{
  "serviceProvider": "WFS",
  "authenticate": false,
  "layerConfig": {
    "service": "WFS",
    "version": "1.1.0",
    "featureNS": "http://www.opengeospatial.net/cite",
    "featurePrefix": "cite",
    "featureName": "test_layer",
    "projection": "urn:x-ogc:def:crs:EPSG:4326",
    "maxFeatures": 100000,
    "outputFormat": "json"
  }
}
```

Geocoders

Pdok Locatieserver Geocoder

Parameter	Value	Notes
URL	`http://geodata.nationaalgeoregister.nl/locatieserver/free`	-
Type	`PdokLocatieserver`	-
Options	`{ "delta": 0.01 }`	-

Cowi Geocoder

Parameter	Value	Notes
URL	-	-
Type	`CowiGeocoder`	-
Options	{ "delta": 0.01 }	Delta is the size in degrees of the bounding box centered on the resulting position.