PDOK Kaart – Technische Documentatie – Versie 1.0.0

Datum: 26 juli 2013

1. Inleiding

Dit document bevat informatie over PDOK Kaart. Hoofdstuk 2 beschrijft de verschillende technische componenten van PDOK Kaart. Hoofdstuk 3 gaat over het gebruik van de Wizard en de Application Programming Interface (API). In hoofdstuk 4 worden beheer technische zaken besproken. Het is belangrijk om op te merken dat de software aangeboden wordt “asis”. Het Kadaster biedt geen ondersteuning aan PDOK Kaart gebruikers anders dan hetgeen aangeboden wordt in deze documentatie, op het PDOK Loket en op het PDOK Forum.

2. Broncode

2.1. Componenten

De PDOK Kaart bestaat uit de volgende componenten:

1. Webservices van PDOK voor de achtergrondkaart, geocoder en andere kaartlagen van
   PDOK
2. API en scripts voor aanroepen van interactieve kaarten met OpenLayers, inclusief functies voor de locatieprikker
3. Instellingenpagina (Wizard) om de API aan te roepen en te configureren
4. Documentatie pagina waarop kort alle parameters beschreven zijn
5. Online help webpagina
6. Voorbeeld webpagina's voor de inpassing van PDOK Kaart: statische HTML en demo's in een website.

2.2. Locatie van de broncode

De broncode kan geladen worden via het PDOK Loket: https://www.pdok.nl/nl/producten/pdok-software/pdok-kaart-software. Naast de broncode is hier ook een webpagina gemaakt welke nadere uitleg geeft over PDOK Kaart en hoe de Application Programming Interface (API) gebruikt kan worden. De uitleg- en voorbeeldpagina is te vinden op http://kaart.pdok.nl/api/index.html. Daarnaast is een korte handleiding te vinden vanuit de Wizard via de Help knop.

2.3. Licentie

De software is beschikbaar in de vorm van een CC-BY-SA 3.0 licentie. Dit betekent dat u de software zelf mag uitbreiden mits u de resultaten ook publiekelijk ter beschikking stelt. De CCBY-SA 3.0 licentie is ook van toepassing op bijvoorbeeld onderstaande achtergrondkaart

zoals deze getoond wordt binnen de PDOK Kaart. De PDOKKaart maakt gebruik van OSS bibliotheken Geozet, OpenLayers en jQuery. OpenLayers heeft een '2-clause BSD License' (zie: https://raw.github.com/openlayers/openlayers/master/license.txt) en jQuery heeft een 'MIT License' (zie: http://github.com/jquery/jquery/blob/master/MIT-LICENSE.txt).

2.4. Bestanden

Voor de instellingenpagina zijn de volgende bestanden en directories van belang:

- api (dit wordt hieronder apart behandeld)
- css
    - rijkshuisstijl (dit is de opmaak voor de rijkshuisstijl welke as-is gebruikt wordt)
    - main.css
- documentatie
    - index.html
- examples
    - deze directory bevat data en bestanden welke de werking van de API demonstreren
- img
    - help
        - deze directory bevat de plaatjes welke gebruikt worden in de helppagina
    - deze directory bevat de plaatjes welke gebruikt worden op de instellingenpagina
- js
    - Geozet
    - img
    - Geozet.js
    - geozet-lib.js
    - jquery.js
    - wizard.js
- help.html
- index.html
- README.md

Voor de API belangrijke bestanden zijn deze:

- img
    - cloud-popup-relative.png
    - layer-switcher-maximize.png
    - layer-switcher-minimize.png
- js
    - OpenLayers.js
    - pdok-api.js
    - pdok-layers.js
    - pdok-markers.js
    - proj4js-compressed.js
- markertypes
    - deze directory bevat de plaatjes van de ondersteunde markers
- sld
    - gemeentegrenzen_label_grijs_gestippeld.sld (dit is een alternatieve opmaak voor de gemeentegrenzen kaart waarin de namen van de gemeenten als labels worden getoond en de grenzen grijs gestippeld weergegeven worden)

- styles
    - default
    - api.css
- api-style.css
- api.html
- index.html

2.5. Technische beslissingen en uitgangspunten

PDOK kaart bestaat uit 2 componenent:

- de API
- de instellingenpagina (Wizard)

De API is bewust ZONDER hulpbibliotheken geschreven, dit om versie problemen in combinatie met CMS'en en dergelijke te voorkomen. De instellingenpagina maakt gebruik van jQuery. Er is echter GEEN gebruik gemaakt van jQuery-ui componenten waardoor de code minimaal gebleven is. Geprobeerd is zo dicht mogelijk bij OpenLayers te blijven. Apimethoden geven - of OpenLayers objecten terug, - of er is een zo dun mogelijke wrapper om OpenLayers-objecten geschreven.

Het uitgangspunt is 'convention over configuration'. Uitgaande van een div in een html pagina met de id 'map' moet het aanmaken van een Api-instantie standaard resulteren in een kaart met daarin de BRT Achtergrondkaart. Er wordt vanuit gegaan dat de pdok-api bestanden (js, html, css en images) centraal gehost worden. Het is echter ook mogelijk om het geheel ZELF te hosten door de code te downloaden en op een eigen server te installeren.

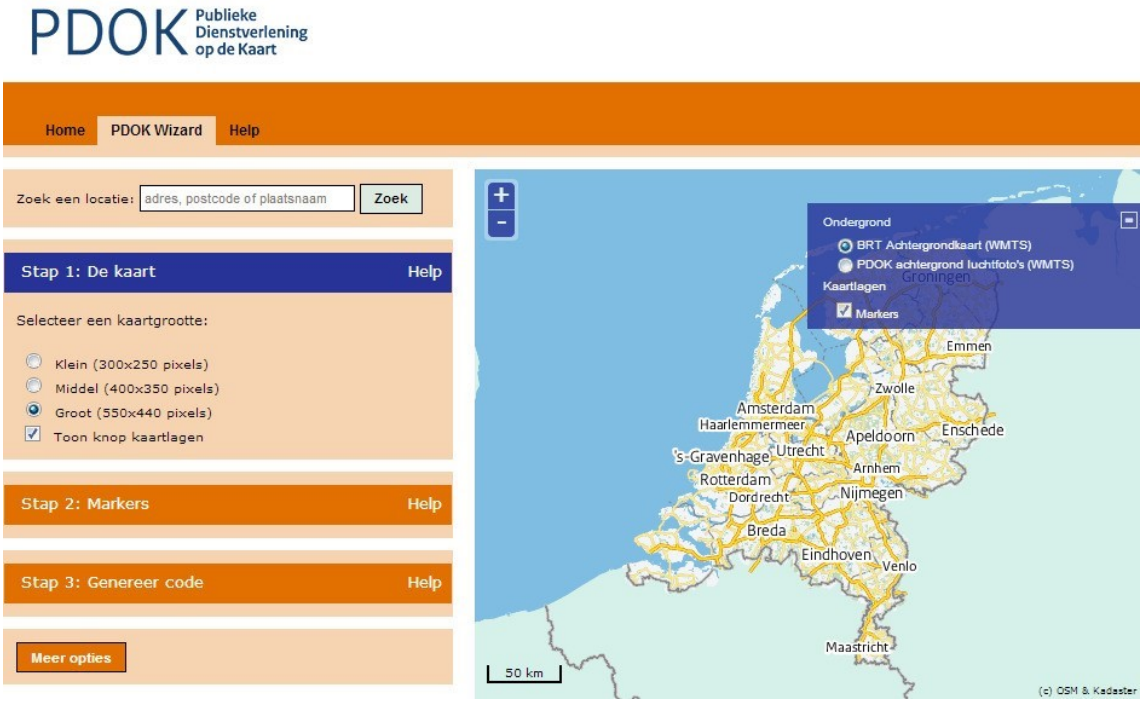
Om zoveel mogelijk onderdelen van de pdok-api aanpasbaar te maken, is het mogelijk om de configuratie bestanden voor de lagen en voor de symbolen extern beschikbaar te stellen. Dit is de reden dat er voor deze configuratie is gekozen voor het json-formaat. Dat kan een browser namelijk cross-domain ophalen.

Indien gebruik gemaakt wordt van externe data (geen images), moet er vrijwel zeker een proxy worden gebruikt. In de api.html pagina is de api instantie bewust globaal aangemaakt onder de naam 'api' om het op deze manier eenvoudig te maken om via javscript tools (firebug) het object te bekijken/bevragen.

3. Gebruik

3.1. Wizard

De Wizard is eenvoudigweg te gebruiken door deze te openen via de url: http://kaart.pdok.nl.



3.2. Parameters API

De API kan op twee verschillende manieren gebruikt worden:

1. De eerste manier is een URL maken en deze gebruiken als link in een website (eventueel als iFrame of Object in de website) of document.
2. De tweede manier is de API opnemen in een webpagina en via JavaScript de kaart aanmaken. Hierbij kan gebruik worden gemaakt van de parameters welke ook voor de URL gelden of van functies welke in de API beschikbaar zijn. Een deel van de instellingen is direct via de instellingenpagina te configureren, zodat URL's en script voorbeelden te maken zijn.

In de documentatie worden de parameters, markers en layers genoemd. Deze zijn op te vragen via de URL http://kaart.pdok.nl/api/index.html.

Naast deze mogelijkheden heeft de API ook meerdere functies welke vanuit JavaScript aangeroepen kunnen worden:

• addFeaturesFromString(data, type, zoomToFeatures)
Functie waarmee data in KML of TXT formaat (als string) aan de kaart worden toegevoegd

• addFeaturesFromUrl(url, type, zoomToFeatures)
Functie waarmee data op basis van een URL aan de kaart wordt toegevoegd

• addKML(url)
Functie waarmee een KML bestand op basis van de URL aan de kaart wordt toegevoegd

• addLayers(arrLayerNames, map)

Functie waarmee een kaartlaag aan een kaart wordt toegevoegd

• addOLLayer(openLayersLayer)
Functie waarmee een TMS, WMS of WMTS layer aan de kaart kan worden toegevoegd

• addTMS(tmsurl, tmslayer, tmstype)
Functie waarmee een TMS op basis van de service informatie kan worden toegevoegd aan de kaart

• addTxt(url)
Functie waarmee een tekstbestand kan worden toegevoegd aan de kaart

• addWMS(wmsurl, wmslayers)
Functie waarmee een WMS op basis van de service informatie kan worden toegevoegd aan de kaart

• addWMTS(wmtsurl, wmtslayer, wmtsmatrixset, wmtsstyle, wmtsmatrixids) Functie waarmee een WMTS op basis van de service informatie kan worden toegevoegd aan de kaart

• createFeature(wkt, typestyle, name, description)
Functie waarmee een feature gemaakt kan worden en aan de featurekaartlaag worden toegevoegd

• createOlMap()
Maakt een OpenLayers kaart op basis van de beschikbare configuratie

• createStyles()
Functie waarmee de stijlen (punt-, lijn- en polygoonstijl) worden gemaakt

• createTMSLayer(layerConfigObj)
Functie waarmee een OpenLayers TMS laag wordt gemaakt

• createWMSLayer(layerConfigObj)
Functie waarmee een OpenLayers WMS laag wordt gemaakt

• createWMTSLayer(layerConfigObj)
Functie waarmee een OpenLayers WMTS laag wordt gemaakt

• deleteLayers(nonVectorLayers, vectorLayers)
Functie waarmee kaartlagen verwijdert worden uit de kaart, de zichtbare baselayer en de featureslayer worden nooit verwijderd

• disableDrawingTool()
Functie waarmee de tekenfunctionaliteit uitgezet wordt

• disableEditingTool()
Functie waarmee de editfunctionaliteit uitgezet wordt

• disableLocationTool()
Functie waarmee de locatie functionaliteit uitgezet wordt

• disablePopups()
Functie waarmee de popups uitgezet worden

• enableDrawingTool(styletype, featureAddedCallback)
Functie waarmee het tekenen aangezet wordt, er moet een stijl en een functieopgegeven worden welke aangeven wat en hoe getekend moet worden en welke functie aangeroepen wordt als het tekenen gereed is

• enableEditingTool(featureModifiedFunction)
Functie waarmee het wijzigen van features mogelijk gemaakt wordt

• enableLocationTool()
Functie waarmee de locatie tool aangezet wordt

• enablePopups()
Functie waarmee de popups worden aangezet

• getConfig()
Functie waarmee de actuele configuratie wordt opgevraagd

• getMapObject()
Functie waarmee de map-object van OpenLayers wordt opgevraagd

• handleGetFeaturesResponse(response)
Functie welke controleert of ontvangen data (bv KML) correct is ontvangen

• moveVectorLayersToTop()
Functie waarmee de vectorenlaag als bovenste laag in de kaart gezet wordt

• removeFormCoordinates()
Functie waarmee de locatie verwijderd wordt uit de pagina als de locatieprikker gebruikt wordt en opnieuw een locatie gemaakt wordt

• removeLocationToolProps()
Functie waarmee de standaardinstellingen voor de locatieprikker gezet worden

• reprojectWGS84toRD(lat, lon)
Functie waarmee een WGS84 locatie omgezet wordt naar een RD locatie

• serialize(obj, stringQuotes)

Functie welke een object serialiseert

• setLayerSwitcherVisible(isVisible)
Functie waarmee de LayerSwitcher (on)zichtbaar gemaakt kan worden

• setLocation(loc)
Functie waarmee het middelpunt van de kaart gezet kan worden

• setLocationToolProps(styletype, zmin, zmax, xorwkt, y)
Functie waarmee de locatieprikker eigenschappen gezet worden

• setZoomLevel(zl)
Functie waarmee het zoom nivo gezet wordt

• startLocationTool()
Functie waarmee de locatieprikker gestart wordt.

3.3. Locatieprikker (Geometrie tekenen met de API)

Om met de API een geometrie te kunnen tekenen in een externe website moet eerst voldaan worden aan enkele voorwaarden. In de instellingenpagina onder 'Meer opties' – 'Kaartprikker functionaliteit toevoegen' zijn deze te configureren. De locatieprikker schrijft in formulierelementen waarden van de geometrie weg. De identifiers van deze elementen dienen opgegeven te worden. De locatieprikker kan voor een punt een losse x- en y-coördinaat wegschrijven in twee aparte formulierelementen en voor een punt, lijn of vlak de gehele geometrie in één formulier element, in het OGC WellKnownText formaat.

Bijvoorbeeld: We willen een HTML formulier maken waarin de locatieprikker moet komen. In twee tekstinvoervelden (al dan niet verborgen) moeten een x en een y waarde opgeslagen

worden. De identifiers (bijvoorbeeld: x_coordinaat en y_coordinaat) van die velden moeten dan opgegeven worden bij de aanroep van de locatieprikker. In de instellingenpagina zijn dat de velden “X-coordinaat” en “Y-coordinaat”. De instellingenpagina genereert hiermee de benodigde JavaScript code om te kopiëren in het betreffende webformulier.

3.5. Werkende versie en andere resources

Op de onderstaande locaties zijn de wizard en andere resources te vinden die betrekking hebben op PDOK Kaart.

PDOK Kaart Wizard:
- http://kaart.pdok.nl/

PDOK Kaart API voorbeelden en documentatie:
- http://kaart.pdok.nl/api

PDOK Kaart Losse voorbeelden:
- http://kaart.pdok.nl/examples/

Deze laatste lijst bevat verscheidene demo's van aanroepen van de API van PDOK Kaart via een URL en via JavaScript includes.

PDOK Kaart Help:
- http://kaart.pdok.nl/help.html

Veel gestelde vragen en antwoorden:
- https://www.pdok.nl/nl/hoe/veelgestelde-vragen#pdok-kaart
- http://kaart.pdok.nl/faq.html

PDOK Kaart Forum:
- https://www.pdok.nl/nl/forums/pdok-kaart-onderwerpen

4. Productie

4.1. Beheer

De laatste broncode van PDOK Kaart is te downloaden van https://www.pdok.nl. De installatie op een webserver is eenvoudig. Nadat van het PDOKK Loket de zipfile gedownload is kan dit uitgepakt worden in een directory op de webserver. Er van uitgaande dat de instellingenpagina en de API geïnstalleerd worden hoeft er niet veel ingesteld te worden.

In het bestand api/js/pdok-api.js moeten de volgende variabelen goed gezet worden:

Pdok.ApiUrl = 'http://kaart.pdok.nl/api';

```
OpenLayers.ProxyHost = "http://"+window.location.host+"/proxy.php?url=";
OpenLayers.ImgPath = './img/';
```

De eerste variabele geeft aan wat de URL van de API is. Deze wordt gebruikt om de markers, de markerdefinities en de layerdefinities van een correcte URL te voorzien.
De tweede variabele geeft aan wat de URL van het proxy-script is welke gebruikt wordt voor het ophalen van onder andere KML- en tekst-bestanden.
De derde variabele geeft aan wat de URL is naar onder andere de plaatjes welke door OpenLayers gebruikt wordt. Omdat de wizard los gebruikt kan worden van de API kan ook in het bestand js/wizard.js deze variabele gezet worden.

De markers en layers definities kunnen ook op een andere locatie of met een andere naam worden opgeslagen. Deze variabelen moeten dan in de api/js/pdok-api.js aangepast worden:

```
Pdok.markersdef = Pdok.ApiUrl+'/js/pdok-markers.js';
Pdok.layersdef = Pdok.ApiUrl+'/js/pdok-layers.js';
```

In het bestand pdok-marker.js zijn de punt, lijn en vlak symbolen opgenomen als een JavaScript Object. Punten worden aangegeven met de letter m (van marker) gevolgd door de letter t (van type) en dan een uniek nummer, lijnen door de letter l en vlakken door de letter p (van polygoon) waarna de letter t komt, ook deze worden gevolgd door een uniek nummer. Bijvoorbeeld mt1, lt1 en vt1.

Let op, in het bestand pdok-api.js zijn ook drie symbolen opgenomen. Dit zijn mt0, lt0 en pt0. Deze zijn voor de zekerheid in de API opgenomen voor het geval dat ze in het bestand pdokmarkers.js niet gevonden wordt. Deze drie zijn dus altijd aanwezig. In een eigen markersbestand kunnen deze definities overschreven worden.

De definities zijn er als volgt uit:

- Markertype:
  - id – de unieke naam
  - name – een beschrijving
  - externalGraphic – de url naar het plaatje
  - graphicHeight – de hoogte van het plaatje
  - graphicWidth – de breedte van het plaatje
  - graphicYOffset – de offset van het plaatje
- Lijntype:
  - id – de unieke naam
  - strokeColor – de lijnkleur
  - strokeWidth – de lijndikte
  - strokeOpacity – de lijntransparantie
  - name – een beschrijving
- Polygoontype:
  - id – de unieke naam
  - fillColor – de vulkleur
  - fillOpacity – de vultransparantie
  - strokeColor – de lijnkleur
  - strokeWidth – de lijndikte

- name – een beschrijving

In het bestand pdok-layers.js zijn alle beschikbare kaartlagen opgenomen. Let op dat een mapservice meerdere keren kan voorkomen omdat voor de afzonderlijke layers van de service een aparte kaartlaag aangeboden kan worden. Mapservices welke ondersteund worden zijn TMS, WMS en WMTS.

Van de services moeten de volgende gegevens ingevuld worden (dit zijn eigenlijk de OpenLayers properties voor de services):

- TMS
  - layertype - TMS
  - name – naam van de service zoals dit getoond wordt in de lijst
  - url – url van de service
  - layername – unieke naam van de service
  - type – extensie van de images van de TMS
  - visibility – standaard zichtbaar of niet
  - isBaseLayer – achtergrond laag in de kaart of niet
  - attribution – eventuele tekst welke op de kaart getoond wordt (dit is optioneel)
- WMS
  - layertype - WMS
  - name – naam van de service zoals dit getoond wordt in de lijst
  - url – url van de service
  - layers – lagen van de wms
  - transparent – is het resultaat transparant of niet

  - format – formaat van de gegenereerde images
  - visibility – standaard zichtbaar of niet
  - isBaseLayer – achtergrond laag in de kaart of niet
  - singleTile – wordt maar 1 plaatje opgehaald of worden meerdere plaatjes opgehaald (getilede plaatjes, dit geeft het idee van snelheid maar is over het algemeen langzamer)
  - attribution – eventuele tekst welke op de kaart getoond wordt (dit is optioneel)

- WMTS
  - layertype - WMTS
  - name – naam van de service zoals dit getoond wordt in de lijst
  - url – url van de service
  - layer – laag van de wmts
  - style – naam van de te gebruiken opmaak-style
  - matrixSet – naam van de te gebruiken matrix (projectie)
  - transparent – is het resultaat transparant of niet
  - format – formaat van de gegenereerde images
  - visibility – standaard zichtbaar of niet
  - isBaseLayer – achtergrond laag in de kaart of niet
  - attribution – eventuele tekst welke op de kaart getoond wordt (dit is optioneel)

Het is mogelijk alleen de API te installeren. De API staat in de directory 'api' in het gedownloade zip-bestand. Eventueel kan zelfs het bestand index.html, waarin de parameters uitgelegd worden en enkele voorbeelden staan, verwijderd worden en de api.html hernoemd naar index.html, zodat een aanroep zonder html-pagina voldoet.