Integration von Geoinformationen mit here-Karten in APEX

Dr. Gudrun Pabst
Trivadis GmbH
München

Schlüsselworte
APEX, Geodaten, here, Kartendarstellung

Einleitung

Durch die Globalisierung können Probleme und Katastrophen auf der ganzen Welt die Lieferketten der produzierenden Unternehmen beeinflussen. Die Visualisierung der Lieferketten und der Risiken in Karten hilft, Zusammenhänge einfacher zu erkennen und schneller auf die Probleme zu reagieren.

Um diese Visualisierung umzusetzen, wurde eine APEX-Applikation programmiert, die teilweise in der Datenbank und teilweise im Browser Geodaten aus dem Geokodierungsdienst von „here“ (ehemals Nokia) erhält. Diese Geo-Informationen werden dann unter Verwendung von APEX-Features sowie den JavaScript-APIs von „here“ in Karten angezeigt.

Im folgenden werden die Schritte gezeigt, die nötig sind, um die „here“-Karten in APEX einzubinden, Abfragen an den Geocodierungsdienst von „here“ zu senden und die erhaltenen Daten mit Hilfe von APEX-Features mit den Geodaten aus der Datenbank zusammenzuführen und in den Karten anzuzeigen.

Wahl eines Kartenanbieters

Es gibt viele Anbieter von Kartendiensten. Die Auswahl des geeigneten Anbieters hängt dabei ab von verschiedenen Kriterien wie den Lizenzkosten, der Genauigkeit des Kartenmaterials sowie der angebotenen API. Im vorliegenden Projekt wurde trotz der Lizenzkosten die Entscheidung für den kartenanbieter here getroffen, da gegenüber Open Street Maps die Qualität der Geokodierung von Adressen sich in der Evaluierung als weniger schwankend gezeigt hat. here stellt neben dem Kartenmaterial auch eine JavaScript-API zum Einbinden von Karten in die eigenen Webseiten sowie eine REST-API zur Abfrage von Geokodierungsinformationen zur Verfügung.

Einbinden einer here-Karte in eine APEX-Seite

Um eine here-Karte in eine APEX-Seite einzubinden, wird die JavaScript-API von here verwendet. Dabei werden folgende Komponenten benötigt:

- Zugangsinformationen
  - here – App-Id
  - here – App-Code
- JavaScript-Libraries
  - `https://js.api.here.com/v3/3.0/mapsjs-core.js`
  - `https://js.api.here.com/v3/3.0/mapsjs-service.js`

- `https://js.api.here.com/v3/3.0/mapsjs-mapevents.js` → Zoom, Verschieben
- `https://js.api.here.com/v3/3.0/mapsjs-ui.js`

- CSS
  - `https://js.api.here.com/v3/3.0/mapsjs-ui.css`
- Region in der Seite
- Dynamic Action mit Initialisierungscode

Die Benutzung von here ist lizenzpflichtig, die Zugangsdaten können für Tests und Evaluierungen aber auch als 90-day Free Trial erworben werden. Die Zugangsdaten bestehen aus der App-Id und dem App-Code, beide Angaben müssen sowohl bei der Verwendung der JavaScript-API als auch der REST-API mitgegeben werden.

Das Einbinden der JavaScript-Liberaries und des CSS in die APEX-Anwendung kann auf zwei Weisen erfolgen:
Wenn nur wenige Seiten Karten enthalten, werden die JavaScript-Libraries und das CSS lokal in den Page-Eigenschaften unter „JavaScript → File URLs“ bzw. „CSS → File URLs“ eingetragen. Sind jedoch in der Mehrzahl der Seiten Karten eingebettet, können die JavaScript-Libraries sowie die CSS-Datei global hinterlegt werden. Dabei wird unter den „Application Properties“ auf der Karteikarte „User Interface“ das User Interface „Desktop“ geöffnet, und dort werden im Bereich „JavaScript → File URLs“ die JavaScript-Libraries und im Bereich „Cascading Style Sheets → File URLs“ die CSS-Datei eingetragen.

Um eine Karte anzeigen zu können, wird ein `<div>`-Bereich mit einer ID benötigt. Daher wird auf jeder Seite, die eine Karte enthält, eine leere HTML-Region für die Karte angelegt. Diese Region hat als Source ein `div` mit einem id-Attribut, mit dem die Region für die Kartendarstellung angesprochen wird, sowie die Größenangaben als `width` und `height` für den gewünschten Kartenbereich.

Damit die leere HTML-Region mit der Kartendarstellung gefüllt wird, muss ein Initialisierungscode ausgeführt werden. Dieser Code wird in einer Dynamic Action hinterlegt, die beim Page Load ausgeführt wird. Zum Initialisieren einer here-Karte wird zunächst ein Plattform-Objekt benötigt, über das die Zugriffsdaten an here übergeben werden. Weitere Informationen über die Zugriffe auf die here-Daten (Verwendung von https, Verwendung der here-Test-Umgebung) werden ebenfalls an das Plattform-Objekt übergeben. Dieses Plattform-Objekt stellt dann auch die Layer für die Karte zur Verfügung. Auf Basis der Layer wird die Karte angezeigt:

```
// Anlegen des Plattform-Objekts mit den benötigten Einstellungen
vPlattform = new H.service.Platform (
  { 'app_id'   : vHereAppId   ,
    'app_code' : vHereAppCode
  } );
vPlattform.setUseHTTPS ( true );
vPlattform.setUseCIT ( true ); // Test-Umgebung

// Default Map Layer erstellen
vDefaultLayers = vPlattform.createDefaultLayers ();
```

```
// Karte erstellen mit normal map als Base Layer
vMap = new H.Map (
  document.getElementById ( 'HereMapContainer' ) ,
  vDefaultLayers.normal.map ,
  { zoom   : vDefaultZoom ,
    center : { lng : vDefaultLong , lat : vDefaultLat }});
```

Erweiterung der Karten um Marker

Die Karte soll Positionen sichtbar machen. Zur Kennzeichnung von Positionen in der Karte werden sogenannte Marker benutzt. Die JavaScript-API von here stellt einen Standard-Marker zur Verfügung. Es können aber auch eigene Marker verwendet werden. Hierfür gibt es zwei Varianten:

- Eigene (statische) Bilder können über eine URL in die Karte eingebunden werden:

- Marker können aber auch als (dynamische) SVG-Grafiken erstellt werden:

Ein rechteckiges SVG-Icon mit Text wird mit folgendem Code erzeugt:

```
/* SVG für Marker-Icon erzeugen */
var vSVGMarker =
  '<svg width="24" height="24" ' +
  '     xmlns="http://www.w3.org/2000/svg">' +
  '<rect stroke="white" fill="#1b468d" x="1" y="1" ' +
  '      width="22" height="22" />' +
  '<text x="12" y="18" font-size="12pt" ' +
  '      text-anchor="middle" fill="white">' +
  vText +
  '</text></svg>';

/* Marker-Icon aus SVG erzeugen */
var vIcon = new H.map.Icon ( vSVGMarker );
```

Um die Ergebnisse einer Suchanfrage an here in einer Karte mit Markern zu kennzeichnen, wird in APEX ein Report erstellt, der die Suchanfrage ausführt und die Ergebnisse auflistet. Die REST-API von here stellt über ihren Geocoding-Dienst die Geoinformationen zu einer Suchanfrage als JSON-Objekt zur Verfügung. Mit Hilfe der Funktion json_table kann dieses Objekt in APEX als Report abgefragt werden:

```
select geo.laenge , geo.breite
  from json_table ( httpuritype (
         'http://geocoder.cit.api.here.com/6.2/geocode.json?' ||
           'app_id=XxXxXxX&app_code=YyYyYyYyYyYyY&gen=9' ||
           '&city='    || :P400_SUCHE_STADT ||
           '&country=DEU' )
         ).getclob() format json ,
         '$.Response.View[*].Result[*].Location'
           columns (
             laenge varchar2(25) path '$.DisplayPosition.Longitude' ,
             breite varchar2(25) path '$.DisplayPosition.Latitude'  ))  geo
```

Die Geopositionen können nun mit JavaScript ausgelesen und als Marker in der Karte angezeigt werden.

Anzeige von Umkreisen

Mit der JavaScript-API von here lassen sich einfach Umkreise von Positionen durch Angabe des Mittelpunkts und des Radius in einer Karte anzeigen:

```
/* Point-Objekt aus Geokoordinaten erzeugen */
var vPoint = new H.geo.Point ( vBreite , vLaenge );

/* Kreis um die Koordinaten */
var vKreis = new H.map.Circle (
               vPoint , vUmkreis ,
               { style: { lineWidth : '1' } } );

/* Kreis zur Karte hinzufügen */
vMap.addObject ( vKreis );
```

Mittels `sdo_geom.sdo_distance` werden die Orte in der DB ausgelesen, die im Umkreis um den gewünschten Punkt liegen, als Report in APEX aufgelistet und anschließend als Marker angezeigt:

```
select gor.gor_ort ,
       gor.gor_geogr_laenge ,
       gor.gor_geogr_breite
  from geo_orte   gor
 where sdo_geom.sdo_distance (
         gor.gor_geoposition ,
         sdo_geometry (
           2001 ,
           8307 ,
           sdo_point_type (
             nvl ( to_number ( :P500_GEOGR_LAENGE ) , 0 ) ,
             nvl ( to_number ( :P500_GEOGR_BREITE ) , 0 ) ,
             null ) ,
           null ,
           null )  ,
         0.005     ,
         'unit=KM' ) <= nvl ( to_number ( :P500_UMKREIS ) , 0 )
```

Verbinden von Positionen

Die JavaScript-API von here stellt auch Funktionalitäten zum Verbinden von Positionen in der Karte mit Linien zur Verfügung. Hierzu muss zunächst aus Koordinatenpaaren ein „Geo-Strip“ erstellt werden. Diese „Geo-Strips“ werden anschließend zur Linienzügen zusammengefasst und in der Karte angezeigt:

```
/* Geo-Strip für die Koordinatenpaare erstellen */
var vGeoStrip = new H.geo.Strip();
vGeoStrip.pushPoint ( { lat: vGeoBreite2 ,
                        lng: vGeoLaenge2 } );
vGeoStrip.pushPoint ( { lat: vGeoBreite1 ,
                        lng: vGeoLaenge1 } );
```

```
/* Linie aus Geo-Strip erzeugen */
var vPolyline =
  new H.map.Polyline (
        vGeoStrip ,
        { style: { lineWidth: 4 ,
                   strokeColor: "rgba(100,0,153,1)" }});
```

Auch die Daten für die Verbindungen können über einen Report in APEX eingelesen und dann mit JavaScript in die Kartendarstellung übertragen werden.

Kontaktadresse:
Dr. Gudrun Pabst
Trivadis GmbH
Lehrer-Wirth-Straße 4
D-81829 München

Telefon:	+49 89 99 27 59 30
Fax:	+49 89 99 27 59 59
E-Mail	gudrun.pabst@trivadis.com
Internet:	www.trivadis.com