



DOAG 2013 Development, 19. Juni 2013, Bonn

Google und Nokia Maps in APEX

Data Design & Management GmbH

Bauernwaldstraße 126, 70195 Stuttgart

Telefon: 0711 – 69 70 71 72

Westerwaldweg 16, 53424 Remagen
(Oberwinter)

Telefon: 022 28 – 911 622

Kontaktperson: Dr. Jan Golka

Mobil: 0172 – 710 32 30

E-Mail: j.golka@d-d-m.de

www.d-d-m.de



- Vorstellung
- Grundlagen
- Google Maps APIs
 - ▶ JavaScript API
 - ▶ JavaScript API v2 ↔ JavaScript API v3
 - ▶ RESTful Web Services
- Nokia Maps API
 - ▶ JavaScript API
- Links zur Dokumentation
- Demosystem

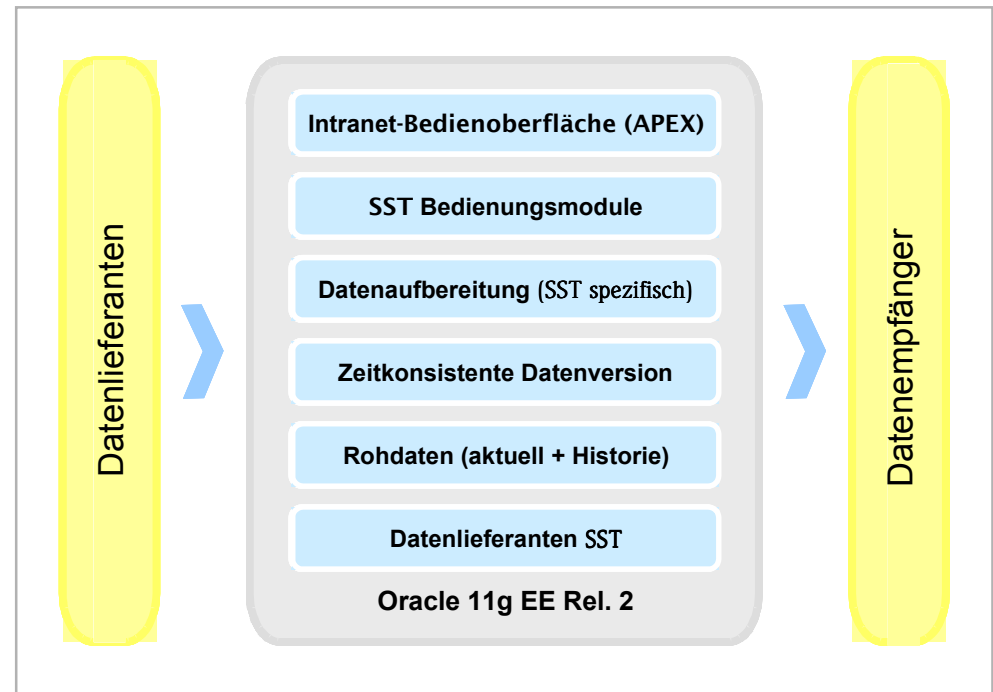


- Wir sind ein kleines, seit 25 Jahren erfolgreiches Softwarehaus (1987 bis 1998 als Personengesellschaft „Dr. Jan Golka Technische Informationssysteme“).
- Unser Schwerpunkt seit 1995: Design und Entwicklung datenbankbasierter Informationssysteme auf Basis von Oracle RDBMS
 - ▶ Analyse, Design und Entwicklung von Informationssystemen
 - ▶ Datenmodellierung und Datenbankdesign
 - ▶ Programmierung der Businesslogik in PL/SQL und Java
 - ▶ Oracle Spatial (Geodaten), Oracle XML DB
- ▶ Entwicklung webbasierter Anwendungen in Oracle APEX, mit Google Maps API, Nokia Maps API, JavaScript und jQuery.
- ▶ Applikationsentwicklung für Android-Plattformen



- Projektbeispiel: Produktions-Stammdaten-Managementsystem (DHL)

- ▶ 815 Datenbanktabellen, 19 Schemas, zwei Oracle Instanzen.
- ▶ Businesslogik: ca. 250.000 PL/SQL und ca. 5.000 Java Codezeilen.
- ▶ APEX UI mit 325 Bildschirmmasken.



- Konzept einer „Chaotischen Sendungszustellung“, GPS-unterstützter dynamischer Zustellung der Paketsendungen.

- ▶ Dieser Vortrag ist ein „offspring“ dieser Überlegungen.



- Sowohl Google als auch Nokia verfügen um immense Mengen von Geodaten und bieten umfangreiche Schnittstellen und Tools für ihre Nutzung und Darstellung.
 - ▶ Nokia hat 2008 den Geodatenanbieter NAVTEQ gekauft.
- Google und Nokia bieten ihre Daten auch für kommerzielle Applikationen an (Beispiel: Nokia/ NAVTEQ Daten in Navigationssystemen).
 - ▶ Nokia bietet zum Beispiel für die kommerzielle Nutzung ihre sehr umfangreiche Maps Web Services (Traffic, Routing, Roadsigns, Points of Interest) nicht nur als REST sondern als auch SOAP.
- In diesem Vortrag beschränken wir uns auf die frei verfügbaren Dienste, wobei auch diese so umfangreich sind, dass wir hier nur einen kleinen Teil davon ansprechen.



- Sowohl Google als Nokia schränken die nichtkommerzielle Maps-Nutzung über eine maximale Anzahl von Zugriffen pro Tag ein:
 - ▶ Google's Geocoding API erlaubt max. 2.500 Geocoding-Requests pro Tag.
- Die Nutzungsbedingungen erlauben einen freien Zugriff auf die Geodaten nur unter der Voraussetzung, dass sie auch auf der Karte angezeigt werden. Google:
 - ▶ „the Geocoding API may only be used in conjunction with a Google map; geocoding results without displaying them on a map is prohibited.“
- Für Nokia Maps notwendig ist die Beantragung der Credentials („appId“, „authenticationToken“) auf per-Anwendung Basis.
- Google hat den bisher notwendigen API Key abgeschafft.



- Google und Nokia bieten freie Nutzung der Geodaten sowie die Programmierschnittstellen für ihre Verwendung in Browserprogrammen, allerdings ohne direkte Zugriffsmöglichkeiten auf die Rohdaten.
- Angeboten werden jeweils zwei Arten der Programmierschnittstellen:
 - ▶ JavaScript API: ein Satz der Funktionsaufrufe in der Programmiersprache JavaScript. Sie sind geeignet für dynamischen Zugriff, z. B. aus einem BO Element.
 - ▶ Webservices: eine Sammlung der HTTP-Schnittstellen zum Bereitstellen der geografischen Daten für Kartenanwendungen.
- Denkbar ist die Nutzung der Karten als Frontend für Applikationssysteme, die eigene Geodaten verarbeiten (also z. B. als Ersatz für den Oracle MapViewer).



- Google stellt folgende Dienste für Kartenanwendungen bereit:
 - ▶ Geocoding API: Konvertierung von Adressen in geografische Koordinaten oder umgekehrt.
 - ▶ Directions API: Berechnung der Routen zwischen Standorten.
 - ▶ Elevation API: Liefert Höhendaten für Standorte auf der Erde.
 - ▶ Places API: Liefert Information über geografische Punkte bzw. die sog. „Points of Interest“.
- Diese Dienste kommen in zwei Ausprägungen vor:
 - ▶ Google Maps **JavaScript API** z. B. für interaktive Karten im Web.
 - ▶ **Webdienste**: URL-Anfragen über Geocodierungs-, Routen-, Höhen und Ortsinformationen.



- Die API besteht aus folgenden Elementen:
 - ▶ Map: `google.maps.Map`
 - ▶ Controls: `google.maps.MapTypeControlOptions`
 - ▶ Overlays (Beispiele):
 - Marker: `google.maps.Marker`
 - InfoWindow: `google.maps.InfoWindow` („Infoblase“).
 - ▶ Services (Beispiele):
 - Geocoder: `google.maps.Geocoder`
 - Directions: `google.maps.DirectionsService`
 - Elevation: `google.maps.ElevationService`
 - ▶ Punkt in geografischen Koordinaten (Breite, Länge):
 - `google.maps.LatLng`.



```
src="http://maps.google.com/maps/api/js?sensor=false,,
function initMap() {
  var options = {
    zoom: 10,
    center: new google.maps.LatLng(&P23_LOCATION.),
    mapTypeId: google.maps.MapTypeId.ROADMAP,
    mapTypeControl: true,
    mapTypeControlOptions: {style: google.maps.MapTypeControlStyle.DROPDOWN_MENU},
    scaleControl: true,
    navigationControl: true,
    navigationControlOptions: {style: google.maps.NavigationControlStyle.DEFAULT}
  };
  var map = new google.maps.Map(document.getElementById("karte"), options);

  var marker = new google.maps.Marker();
  marker.setPosition(map.getCenter());
  marker.setMap(map);

  var title = "Str.: &P23_STREET. <br>Ort: &P23_CITY.<br>Koord.: &P23_LOCATION.";
  var infoWindow = new google.maps.InfoWindow();
  infoWindow.setContent(title);
  infoWindow.open(map, marker);
}
```



- Kein API Key mehr:
 - ▶ `<script src="http://maps.google.com/maps/api/js?sensor=true" type="text/javascript"></script>`
- Die Ausgabe als CSV ist in der API Version 3 nicht mehr möglich. JSON wird empfohlen.
- V2:

```
function initialize() {
    if (GBrowserIsCompatible()) {
        var map = new GMap2(document.getElementById("karte"));
        map.setCenter(new GLatLng(37.4419, -122.1419), 13);
        map.setUIToDefault();
    }
}
```
- V3:

```
function initialize() {
    var myOptions = {zoom: 13,
        center: new google.maps.LatLng(37.4419, -122.1419),
        mapTypeId: google.maps.MapTypeId.ROADMAP
    };
    var map = new google.maps.Map(
        document.getElementById("karte"), myOptions);
}
```



- Eine Sammlung von HTTP-Schnittstellen für Google-Dienste zum Bereitstellen geografische Daten für Kartenanwendung.
Dienstbeispiele:
 - ▶ Geocoding API: Konvertierung von Adressen in geografische Koordinaten oder umgekehrt.
 - ▶ Directions API: Berechnung der Routen zwischen Standorten.
 - ▶ Elevation API: Liefert Höhendaten für Standoere auf der Erde.
 - ▶ Places API: Liefert Information über geografische Punkte bzw. die sog. „Points of Interests“.
 - ▶ URL für die Dokumentation:
<https://developers.google.com/maps/documentation/webservices/>



- Url für die Webanfrage:

- ▶ `http://maps.googleapis.com/maps/api/<dienst>/<output>?parameters`
- ▶ dienst: geocoding, elevation, directions, ...
- ▶ output:
 - json (empfohlen), Ausgabe in JavaScript Object Notation (JSON).
 - xml, Ausgabe in XML-Format.
- ▶ parameters: Kontextabhängig. Abschluss mit „sensor=(true/false)“.

- Beispiel:

- ▶ `http://maps.googleapis.com/maps/api/geocode/json?address=Hauptstr.+1+Remagen&sensor=false`
- ▶ Der Parameter `sensor=false` besagt, dass die Anwendung keinen Sensor zur Standortbestimmung verwendet.



```
res utl_http.html_pieces;
```

```
...
```

```
address := :P23_STREET||','||:P23_CITY;
```

```
address := replace(address, ' ','+');
```

```
if :P23_XML_JSON = 'J' then
```

```
    url := 'http://maps.googleapis.com/maps/api/geocode/json?address=
           '||address||'&'||'sensor=false';
```

```
else
```

```
    url := 'http://maps.googleapis.com/maps/api/geocode/xml?address=
           '||address||'&'||'sensor=false';
```

```
end if;
```

```
res := utl_http.request_pieces(url,2, APEX_APPLICATION.G_PROXY_SERVER);
```

```
if res.count > 0 then
```

```
    for i in 1..res.count loop
```

```
        response := response || res(i);
```

```
    end loop;
```

```
end if;
```



- Credentials („ appId“ und „ authenticationToken“) müssen auf <http://developer.here.com> beantragt werden.
- Die Schnittstelle besteht aus drei Bibliotheken:
 - ▶ Maps: The Maps library is designed to help you create interactive and static maps in several modes, among them the standard physical street/road maps, terrain maps and maps based on satellite imaging (with and without labels). The maps can contain multiple layers of content such as routes, markers, heat maps or geo shapes.
 - ▶ Places: The Places library allows to enrich Web sites by providing a sophisticated search facility for users and a customizable means to display the results that include such information as place name, address, contact details, location map, rating, reviews, as well as related premium content.
 - a *UI library* – a set of user interface widgets with methods that support place data search, retrieval and display.
 - ▶ Services (Positioning, Routing, Traffic).



```
src="http://api.maps.nokia.com/2.2.4/js1.js?with=all"
```

```
nokia.Settings.set("appld", ... ); nokia.Settings.set("authenticationToken", ... );
```

```
function initMap() {
```

```
  infoBubbles = new nokia.maps.map.component.InfoBubbles();
```

```
  var options = { center: [&P23_LOCATION.]
```

```
    , zoomLevel: 12
```

```
    , components: [infoBubbles
```

```
      , new nokia.maps.map.component.Behavior()
```

```
      , new nokia.maps.map.component.ZoomBar()
```

```
      , new nokia.maps.map.component.ScaleBar()
```

```
      , new nokia.maps.map.component.TypeSelector()
```

```
      , new nokia.maps.map.component.Overview()
```

```
      , new nokia.maps.map.component.PublicTransport()
```

```
      , new nokia.maps.map.component.Traffic() ] };
```

```
var map = new nokia.maps.map.Display(document.getElementById(„karte“), options);
```

```
var marker =
```

```
  new nokia.maps.map.StandardMarker( [&P23_LOCATION.], { text: "A", draggable: false } );
```

```
map.objects.add(marker);
```

```
var title = "<div>"+"Str.: &P23_STREET."+"<br/>Ort.: &P23_CITY."+"</div>";
```

```
var bubble = infoBubbles.addBubble(title, [&P23_LOCATION.]);
```

```
}
```


- Google Maps JS API:
 - ▶ <https://developers.google.com/maps/documentation/geocoding/?hl=de>
 - ▶ <https://developers.google.com/maps/documentation/javascript/geocoding>
- Google Maps REST Web Services:
 - ▶ <https://developers.google.com/maps/documentation/webservices/>
- Google Dokumentation ist vollständig und sehr übersichtlich organisiert!
- Nokia Maps JS API:
 - ▶ <http://developer.here.com/>



- Es folgt jetzt eine live Demo mit Codebeispielen.
- Wir beantworten gerne Ihre Fragen und unterstützen Sie gerne in Ihren Entwicklungen!
- Unsere Kontaktdaten:



Data Design & Management GmbH

Dr. Jan Golka

E-Mail: j.golka@d-d-m.de

Telefon: 0711 – 69 70 71 72

Mobil: 0172 – 710 32 30

Internet: www.d-d-m.de