

Das Kartencontrol und die Google Maps API

Einleitung



In einem anderen Papier habe ich beschrieben, wie das Kartencontrol in TD Mobile ohne Frontend-Programmierung, sondern nur durch Zuweisung von Eigenschaftswerten verwendet werden kann.

In diesem Papier geht es um die Klärung der Frage, wie der Entwickler vorgehen muss, wenn er eine etwas andere, vom Standard abweichende Darstellung im Kartencontrol vornehmen möchte.

Zunächst heisst das, dass er nicht ohne JavaScript-Programmierung auskommen wird. Das bedeutet aber auch, dass er neben den eigentlichen Anforderungen auch zusätzliche Funktionalitäten codieren muss, um ein ähnliches Verhalten des Kartencontrols wie im Standardfall zu implementieren. Und schließlich muss sich der Entwickler mit der (JavaScript) API von Google Maps auseinandersetzen.

Da dieses drei unterschiedliche, schwierige Einstiegshürden sind bzw. sein können, soll in diesem einfachen, mit JavaScript programmierten Beispiel schrittweise vorgegangen werden.

Inhaltlich geht es zunächst um folgende Fragen:

- Die Karte soll erst angezeigt werden, wenn der Benutzer eine Eingabe vorgenommen hat. In diesem Fall muss der Anwender lediglich auf die Schaltfläche tippen, um die Anzeige der Karte mit den individuellen Markern auszulösen. Es soll also vereinfacht der Fall behandelt werden, bei dem die Karte dann gefüllt wird, wenn der Benutzer eine Eingabe vorgenommen oder seine Geoposition übermittelt hat.
- Da die Marker erst nach einer Eingabe angezeigt werden sollen, muss die Standardanzeige, die auf irgendeinen Punkt in Westafrika zeigt, verändert werden. Dieses soll „Initialisierung“ genannt werden und diese Initialisierung wird über eine JavaScript-Funktion `karteInitialisieren` vorgenommen. Wir werden weiter unten sehen, dass der Vorgang etwas komplexer ist.
- Nach Antippen der Schaltfläche soll die Karte mit den zwei Markierungen mit einem eigenen Image so angezeigt werden, dass alle Markierungen im Bildschirmausschnitt direkt sichtbar sind. Es geht also darum, die Karte so zeichnen zu lassen, dass der Zoomfaktor der Karte durch die Begrenzungen der Geopositionen errechnet wird.

Wie das live-Beispiel zeigt, können derartige Aufgaben mit TD Mobile gelöst werden. Die Frage, die sich selbstverständlich stellt und die nun erörtert werden soll, lautet: Wie macht man so etwas?



oben: Initialisierung der Karte: nicht Westafrika, sondern Deutschland, Österreich und die Schweiz werden angezeigt.

unten: die Karte mit Markierungen für die beiden Büros



Codierung und Erläuterung

Arbeiten mit der google.maps API

Die Firma Google stellt ihre Kartendienste (und weitere) über unterschiedliche API's zur Verfügung. In diesem Fall wird die JavaScript API verwendet, da es darum geht, die Geräte- und Herstellerunabhängigkeit der mit TD Mobile geschriebenen Anwendungen zu bewahren.

Die API hat hinsichtlich ihrer Nutzung und Nutzungshäufigkeit Einschränkungen, die aber in Bezug auf die Nachvollziehbarkeit des Beispiels nicht wichtig sind. Bevor der produktive Einsatz der Google Maps API vorgenommen wird, muss man sich daher mit den Lizenzbedingungen von Google auseinandersetzen.

Eine globale Variable definieren

Die Karte (innerhalb des Containers `map1`) muss in der Anwendung mindestens zwei Mal angesprochen werden: einmal um die Karte zu initialisieren, das zweite Mal, um die Markierungen auszuführen.

Wir brauchen also eine JavaScript-Variable, die von beiden Funktionen referenziert werden kann, die also auch nach dem Ablauf der einen oder anderen JavaScript-Funktion noch existiert. Eine in diesem Sinne „globale“ Variable auf einer Seite kann beim HTML-Objekt selbst hinterlegt werden. So wird die „globale“ Variable `karte` definiert.

```
<script>
var karte;
</script>
```

Innerhalb des HTML-Controls wird ein Script-Tag angelegt und in diesem Tag wird die Variable `karte` deklariert. Es wird noch kein Wert zugewiesen.

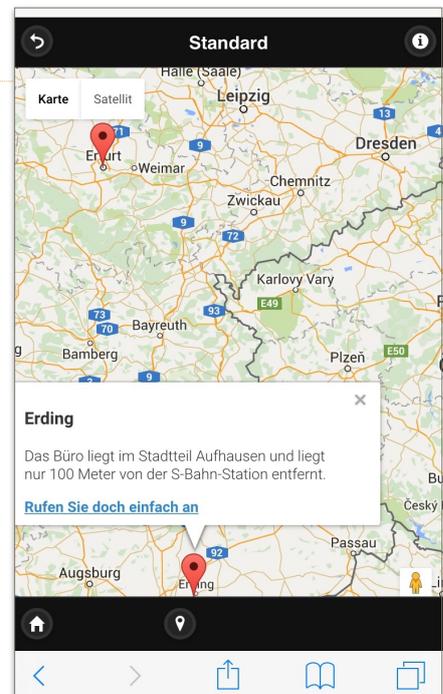
Initialisieren des Kartencontrols

Der hier „Initialisierung“ genannte Vorgang ist tatsächlich etwas komplexer: Wenn die Seite und damit auch der Kartencontainer auf die Seite übertragen ist, wird die Karte standardmäßig von der Seite (auf dem mobilen Gerät) bei Google angefordert. Da aber zu diesem Zeitpunkt kein Breiten- und Längengrad definiert ist, wird die Karte auf den Längengrad 0 und Breitengrad 0 ausgerichtet und zentriert. Das Kartencontrol wird somit **vor**initialisiert, das bedeutet, dass das Kartencontrol im Standardfall einen Ausschnitt vom westlichen Afrika anzeigt. Das ist unschön und irreführend.

Aus diesem Grund soll die Karte erst dann angezeigt werden, wenn sie auf einen vorinitialisierten Wert mit einer entsprechenden Kartenausschnitt ausgerichtet ist.

Um eine andere Karte anzuzeigen, muss eine veränderte Karte angefordert werden. In diesem Fall soll diese Initialisierung auf den Breitengrad 48.274654 und den Längengrad 11.896448 zentriert und mit einer Zoomstufe 5 eingestellt werden. Der Zoom wird deshalb so eingestellt, damit mindestens die Länder Deutschland, Österreich und die Schweiz angezeigt werden. Eine (neue) Karte wird über die API durch Instanziierung des Map-Objekts vorgenommen. Dabei müssen einige Optionen angegeben werden. Diese „Initialisierung“ wird über den nachfolgenden Code, der in der Funktion `karteInitialisieren` hinterlegt ist.

```
var optionen = {center: {lat: 48.274654, lng: 11.896448}, zoom:5};
```



Das Infenster einer Markierung wird angezeigt: es kann direkt gewählt und angerufen werden.

```
karte = new google.maps.Map(document.getElementById ( 'map1' ), optionen);
```

In der Variablen `optionen` werden zwei Attribute (`center`, `zoom`) definiert und als **ein** Argument – deshalb die äußeren Klammern `{}` – als Parameter bei der Instanziierung (`new`) übergeben. Das Ergebnis wird der „globalen“ Variable `karte` zugewiesen.

Diese eigene „Initialisierung“ kollidiert mit der ursprünglichen Initialisierung, die automatisch aufgrund der Anzeige des Kartencontrols vorgenommen wird. Um diese Kollision zu vermeiden, müssen diese beiden „Initialisierungen“ nacheinander vorgenommen werden.

```
Invoke StandorteErmitteln
  Invoke Initialisieren
    If STATUS_OK
      Javascript karteAnzeigen
```

Die Sichtbarkeit des Kartencontrols wird an die Bindung `STATUS_OK` gebunden, die zunächst `false` ist. In der Create-Sektion der Seite wird die Operation `StandorteErmitteln` aufgerufen. Hierdurch wird die Bindung `DIEADRESSE` mit Werten versorgt. Die danach abzuarbeitende Operation `Initialisieren` führt lediglich `SalPause (500)` aus und gibt danach `true` an die Bindung `STATUS_OK` zurück. Das Kartencontrol wird angezeigt, während die Anfrage zur Anzeige der Karte aus der JavaScript-Funktion `karteInitialisieren` ausgeführt wird: die Karte ist sichtbar und zeigt Deutschland, Österreich und die Schweiz. Nach Anklicken der Schaltfläche `Anzeigen` wird die JavaScript-Funktion `karteAnzeigen` ausgeführt.

Anzeigen der eigenen Markierungen

Im ersten Block wird Bindung `DIEADRESSEN` in ein Array `positionen` umgewandelt und die Anzahl der in diesem Array enthaltenen Elemente in der Variablen `anzahl` ermittelt. Danach werden die lokalen Variablen der Funktion `karteAnzeigen` deklariert.

```
var positionen = Td.Data.Binding.get("DIEADRESSEN").getValue();
var anzahl = positionen.length;
var lat;
var lng;
var mylatlng;
var marker;
var grenzen;

grenzen = new google.maps.LatLngBounds();

var image = {
  url: 'images/MD_Logo.png',
  origin: new google.maps.Point(0, 0),
  anchor: new google.maps.Point(0, 32),
  scaledSize: new google.maps.Size(60, 46)
};

for (i=0; i< anzahl; i++){
  lat = positionen[i].lat.Value;
  lng = positionen[i].lng.Value;
  mylatlng = {lat:lat, lng:lng};
  marker = new google.maps.Marker({
    position: mylatlng,
    map: karte,
    title: positionen[i].Adresse,
    icon: image
  });
  grenzen.extend(marker.getPosition());
```

```
};
karte.fitBounds(grenzen);
```

Die Instanz von Google Maps `LatLngBounds` wird der Variablen `grenzen` zugewiesen. Die Klasse `LatLngBounds` wird verwendet, um aufgrund von Kartenmarkierungen automatisch die äußeren Begrenzungen ermitteln zu lassen. Wie man dabei vorgehen muss, wird im Folgenden erläutert.

Im nächsten Block wird die im Unterverzeichnis `images` liegende Bilddatei `MD_Logo.png` formatiert und das Ergebnis der Variablen `image` zugewiesen. Bei der Instanziierung der Klasse `google.maps.Size()`, ist es im Konstruktor möglich, das Attribut `scaledSize` zu setzen. In diesem Fall wird das (viel größere) Image wird mit dem Attribut `scaledSize` auf 60px Breite und 46px Höhe proportional verkleinert. In der im nächsten Block folgenden `for`-Schleife werden Marker-Klassen mit den entsprechenden Optionen instanziiert.

Folgende Konstruktur-Variablen werden in diesem Beispiel bei der Instanziierung der Marker-Klasse gesetzt:

- `position`: Hier wird die Geoposition in der Form `{lat: wert, lng: wert}` erwartet. Dieses Argument wird in der Variablen `myLatLng` „zusammengesetzt“, indem aus dem Array mit Value der numerische Wert für `lat` und `lng` extrahiert und das Ganze zusammengesetzt wird.
- `map`: Hier wird die „globale“ Variable `karte` angegeben
- `title`: Als Titel der Markierung wird die Instanzvariable `Adresse` aus der Arraybindung `DIEADRESSEN` angegeben
- `icon`: Als Bild wird an der Markierungsposition das vorher verkleinerte Objekt `image` angegeben.

Innerhalb der `for`-Schleife muss nun jede Markierung an das `LatLngBounds` übergeben werden. Mit der Methode `extend` der Klasse `grenzen` wird eine Geoposition an die Klasse `LatLngBounds` der Markierung übergeben, damit die Anzeigegrenzen für alle Markierungen „berechnet“ werden kann. Das geschieht mit der Methode `getPosition` der Markerklasse.

Nachdem alle Marker in der Karte platziert und die Breiten- und Längengrad-Begrenzungen in der Instanz der Klasse `LatLngBounds` aufgenommen worden sind, muss die Karte auf den Kartenausschnitt begrenzt werden. Das wird mit der Methode `fitBounds` des Kartenobjekts `karte` mit dem Parameter `grenzen` vorgenommen.

Zusammenfassung

Zur Anzeige eigener Markierungen werden die (Google Maps) Klassen `Maps`, `Marker`, `LatLngBounds`, `Point` und `Size` verwendet. Bei der Instanziierung mit `new` werden unterschiedliche Konstruktoren spezifiziert. Fassen wir zusammen:

Individuelle Markierungen auf einer Karte zu setzen, ist nur eine von vielen Möglichkeiten, die die Google Maps API anbietet. Schauen Sie sich das Papier an, in dem die Anzeige einer (oder mehrerer) Routen auf einer Karte beschrieben wird. In der nebenstehenden Tabelle werden die verwendeten Klassen und Konstruktoren noch einmal zusammengefasst:

Google Maps Klasse	Konstruktor-Variablen	Erläuterung
<code>Map</code>	<code>center</code> <code>zoom</code>	die Geoposition mit Längen- und Breitengrad eine Zoomstufe von 1 bis 15
<code>Marker</code>	<code>position</code> <code>map</code> <code>title</code> <code>icon</code>	die Geoposition mit Längen- und Breitengrad das Kartenobjekt – die globale Variable <code>karte</code> der Anzeigetext das Image, das angezeigt werden soll
<code>LatLngBounds</code>	(keiner)	
<code>Point</code>	<code>(y)</code> <code>(x)</code>	Angabe in Pixel (px)
<code>Size</code>	(Breite) (Höhe)	Angabe in Pixel (px)

MD Consulting & Informationsdienste GmbH

Michaelisstraße 13a
99084 Erfurt

Berghamer Straße 14
85435 Erding

phone (+49) 8122 97400 email info@md-consulting.de