

Das Kartencontrol und Routen

Einleitung



Nachdem wir die Verwendung des Kartencontrols ohne Programmierung und die einfache Programmierung von Markierungen mit eigenen Images behandelt haben, soll in diesem Papier die Frage behandelt werden, wie man eine Route auf einer Karte mithilfe der Google Maps API einzeichnen lassen kann.

Aufgrund der Ausführungen im Papier „Das Kartencontrol und die Google Maps API“ ist geklärt, wie der Entwickler vorgehen muss, um Funktionalitäten der Google Maps API in das TD Mobile Kartencontrol zu integrieren:

- Die Karte muss in irgendeiner Form initialisiert werden (unsichtbar oder vorinitialisiert auf einen bestimmten Kartenausschnitt)
- Nach der Benutzereingabe muss der Aufbau und die Anzeige der gewünschten Karte angefordert und angezeigt werden.
- Um zu wissen, welche Google-Maps-Klassen für die gewünschte Karte benötigt werden (und wie sie gegebenenfalls initialisiert werden), muss man sich ein bisschen mit der entsprechenden Dokumentation beschäftigen.

In diesem Papier soll exemplarisch gezeigt werden, dass es noch weitere Klassen in der API gibt, die über JavaScript in einer TD Mobile Anwendung integriert werden können.

Um eine Route in eine Karte eintragen zu können, müssen die (Google Maps) Klassen **DirectionsService** und **DirectionsRenderer** instanziiert werden. Die Klasse **DirectionsService** bietet Methoden an, mit denen man sich beliebige Routen berechnen lassen kann, während die Klasse **DirectionsRenderer** den Verlauf einer Route auf einer Karte visualisiert.

Schauen wir uns die Verwendung dieser Klassen in einem JavaScript-Beispiel genauer an:

Codierung und Erläuterung

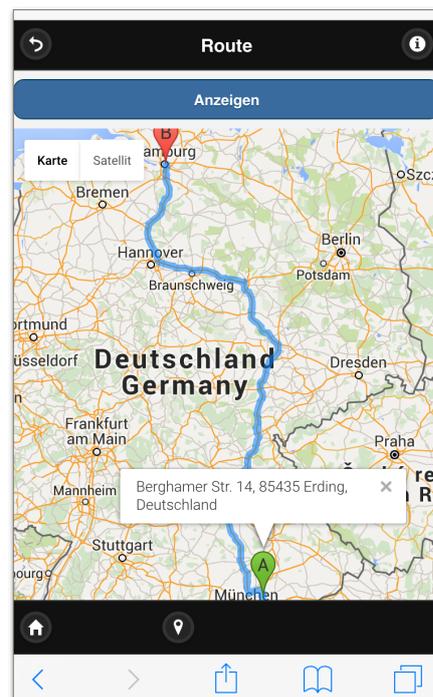
Die JavaScript-Funktion **karteZeigen** – in diesem Fall auf der Seite **pageKarteRoute** – wird ausgeführt, nachdem der Anwender auf die Schaltfläche **Ausführen** getippt hat.

Der Inhalt der JavaScript-Funktion **karteZeigen** ist nachfolgend dargestellt.

```

Funktion: karteZeigen
Parameters:
  Var: positionen
Javascript
var directionsService = new google.maps.DirectionsService();
var directionsDisplay = new google.maps.DirectionsRenderer();
var start = positionen[0].Adresse;
var ende = positionen[1].Adresse;
var options = {
  origin: start,
  destination: ende,
  travelMode: google.maps.TravelMode.DRIVING,
  provideRouteAlternatives: false,
};

```



Das Kartencontrol zeigt die (Auto-) Route von Erding nach Hamburg. Für Ausgangs- und Endpunkt sind Standardmarker gesetzt. Im Infofenster werden die Start- und Zieladresse angezeigt, wenn der Marker angetippt wird.

```

directionsService.route( options, function(response, status) {
  if (status == google.maps.DirectionsStatus.OK) {
    directionsDisplay.setDirections(response);
  };
});
directionsDisplay.setMap(karte);

```

- Die Bindung mit in diesem Fall zwei Adressen (Ort, Strasse, Land) wird als Aufrufparameter an die JavaScript-Funktion unter dem Namen `positionen` übergeben.
- Zunächst werden die beiden Klassen `DirectionsService` und `DirectionsRenderer` instanziiert.
- Aus dem Array `positionen` wird das Element 0 und 1 extrahiert und der jeweilige Wert der Variablen `Adresse` wird den Variablen `start` und `ende` zugewiesen. (Das ist zwar nicht notwendig, macht aber den Code leserlicher, man kann die Werte auch selbstverständlich direkt zuweisen.)
- Als nächstes werden die Optionen für den Aufruf der Methode `route` der Klasse `DirectionsService` definiert: Als `origin` wird `start`, als `destination` wird `ende` und als `travelMode` wird die Google Maps-Konstante `google.maps.TravelMode.DRIVING` definiert. Als letztes schließlich wird festgelegt, dass keine Routenalternativen berechnet werden sollen.
- In der nächsten Zeile wird die Methode `route` des Objekts `directionsService` aufgerufen und als erster Parameter werden die Routenkriterien, die in der Variablen `options` definiert wurden, übergeben. Beim zweiten Parameter handelt es sich um eine Callback-Funktion. Diese Funktion wird automatisch ausgeführt, nachdem `directionsService.route` beendet ist.
- Im Parameter `status` wird zurückgegeben, ob die Ausführung von `directionsService.route` erfolgreich war. Aus diesem Grund wird die Variable `status` mit der Google-Maps-Konstanten verglichen, die signalisiert, ob die Anfrage erfolgreich bearbeitet werden konnte.
- Wenn die Routenanfrage von Google erfolgreich bearbeitet werden konnte – was in diesem Fall möglich ist – wird die Methode `setDirections` der Klasse `DirectionsDisplay` mit der (komplexen) Route, die als erste Parameter `response` zurückgegeben wurde, eingetragen (und noch nicht angezeigt).
- Erst die Methode `setMap` der Klasse `DirectionsDisplay` bindet die berechnete Route mit dem Kartencontrol. Deswegen wird als Parameter die „globale“ Variable `karte` bei der Methode `setMap` eingetragen.

Zusammenfassung

Mit diesem Beispiel wird gezeigt, dass man mit dem Kartencontrol und etwas JavaScript-Programmierung ganz andere Sachen machen kann als im Standardfall vorgesehen. In diesem Fall wird das Kartencontrol genutzt, um eine Route anzuzeigen anstatt lediglich eine oder mehrere Geopositionen mit dem Standardimage oder einem eigenen Image anzuzeigen.

Wichtig ist meines Erachtens bei diesem Beispiel, dass keine Geopositionen des Start- und Endpunkts einer Route angegeben wurden, sondern „Adressen“ (in der Form Ort, Strasse, Land). Die Methode `route` akzeptiert Adressangaben, was den Entwickler von der mühseligen Aufgabe enthebt, zunächst die Geoposition von den beiden Punkten zu ermitteln.

Über das konkrete Beispiel hinaus (in dem als Beispiel die Google Maps API verwendet wurde), sollte gezeigt werden, dass man in TD Mobile Anwendungen

- aus einer JavaScript-Funktion auf die Bindungen (unabhängig von der Komplexität oder Anzahl der Werte) einer Seite zugreifen kann und
- mit diesen Daten, die in der Regel serverseitigen Operationen entstammen, vollkommen andere Visualisierungen vornehmen kann, als im Standard vorgesehen ist.

MD Consulting & Informationsdienste GmbH

Michaelisstraße 13a
99084 Erfurt

Berghamer Straße 14
85435 Erding

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