

Das Kartenkontrol und Benutzereingaben

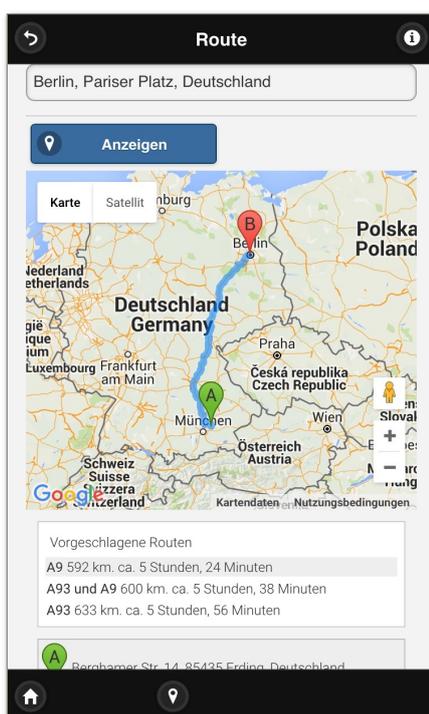
Einleitung



Die unterschiedlichen Möglichkeiten bei der „Umgestaltung“ des Kartencontrols durch eigene JavaScript-Funktionen, die wiederum die Möglichkeiten der Google Maps API nutzen, wurden bereits ausführlich behandelt.

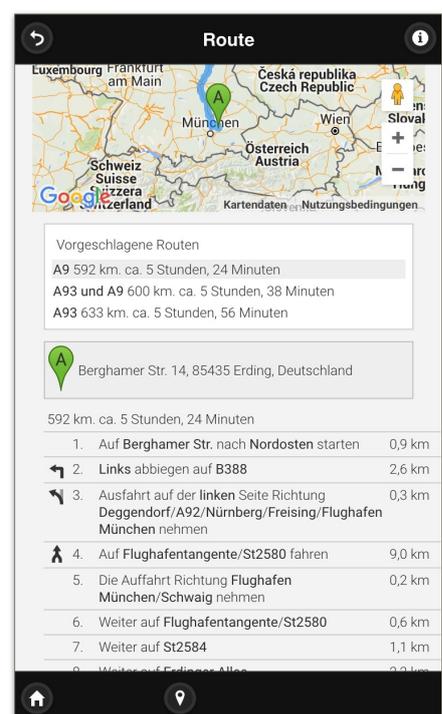
In diesem vorläufig letzten Papier zum Thema „Google Maps“ und TD Mobile JavaScript“ soll es darum gehen, das Ganze in einem möglichen Einsatzszenario noch einmal zusammenfassend darzustellen.

Folgendes Szenario wird angenommen: Es gibt zwei Eingabefelder, in denen der Anwender eine Start- und eine Endadresse eintragen kann. Wenn der Anwender auf die Schaltfläche Anzeigen tippt, soll die Route vom Anfangs- zum Endpunkt in die Karte eingetragen und – zusätzlich darunter – mögliche alternative Routen zur Auswahl angeboten werden. Darunter schließlich soll die ausgewählte Route in Textform mit allen möglichen Wendepunkten angezeigt werden.



Links: Es wird die Route von Erding nach Berlin angezeigt. Darunter wird eine Auswahlbox für alternative Fahrtrouten angezeigt:

Rechts: Zu sehen ist zusätzlich die ausführliche, textliche Routenbeschreibung der ausgewählten Route



Codierung und Erläuterung

Zur Realisierung der Aufgabenstellung müssen folgenden Arbeitsschritte erledigt werden:

- Anlegen von zwei Datenfeldern für die Eingabe der Start- und Endadresse. Diese Datenfelder werden mit der String-Bindung VON und NACH verbunden.
- Es muss ein Container angelegt werden, in dem die Routenbeschreibung, die von Google stammt, angezeigt werden kann.
- Es muss die Route (und mögliche Routenalternativen) aufgrund der Benutzereingaben berechnet und angezeigt werden und
- Optional müssen die jeweilige Routenbeschreibung und die Box, in der alternative Routen angezeigt werden, (etwas) gestaltet werden.

Anlegen eines <div>-Elements für die Routenbeschreibung

Da in diesem Beispiel nicht nur die Route in der Karte eingetragen werden soll, sondern auch alternative Routen und deren jeweilige textliche Beschreibung angezeigt werden soll, muss ein „Behälter“ in der Oberfläche angelegt werden, der diese Elemente aufnehmen kann.

```
<div id="beschreibung" class="beschreibung">
</div>
```

In diesem Fall wird ein weiteres HTML-Control mit dem Namen html3 auf die Oberfläche (unter das Kartencontrol) gezogen. Als HTML-Code wird lediglich ein `div`-Tag eingetragen, in dem die Attribute `id` und `class` spezifiziert werden. Über die `id` kann das Element in der JavaScript-Funktion angesprochen werden und das `class`-Attribut wird (später) dazu verwendet, die Darstellung der Daten (etwas) zu formatieren.

Die JavaScript-Funktion karteZeigen

Auf dieser Seite hat die JavaScript-Funktion folgende Codezeilen, die sich nicht wesentlich von dem vorherigen Beispiel unterscheiden

```
var optionen = {center: {lat: 48.274654, lng: 11.896448}, zoom:6};
karte = new google.maps.Map(document.getElementById ( 'map1' ), optionen);
var directionsService = new google.maps.DirectionsService();
var directionsDisplay = new google.maps.DirectionsRenderer();
var start = Td.Data.Binding.get("VON").getValue();
var ende = Td.Data.Binding.get("NACH").getValue();

var options = {
  origin: start,
  destination: ende,
  travelMode: google.maps.TravelMode.DRIVING,
  provideRouteAlternatives: true,
};
directionsService.route( options, function(response, status) {
  if (status == google.maps.DirectionsStatus.OK) {
    directionsDisplay.setDirections(response);
  } else {
    alert('Es konnte keine Adresse ermittelt werden [ ' + status + ' ]');
  }
});
directionsDisplay.setMap(karte);
```

```
directionsDisplay.setPanel(document.getElementById("beschreibung"));
```

Auf folgende Punkte der JavaScript-Funktion `karteZeigen` soll hingewiesen werden:

- Als erstes wird eine Dummy-Initialisierung der Karte vorgenommen.
- Die Benutzereingaben werden an die Bindungen VON und NACH gebunden. Daher kann in der JavaScript-Funktion über die TD Mobile API auf diese Bindungen zugegriffen werden.
- Bei den Optionen für die Methode `route` wird in diesem Fall das Attribut `provideRouteAlternatives` auf `true` gesetzt.
- Da der Benutzer Unsinn eingeben kann, wird im else-Zweig der Statusabfrage eine Nachricht ausgegeben, dass keine Route ermittelt werden konnte.
- In der letzten Zeile der Funktion schließlich wird mit `setPanel` die Darstellung von Routenalternativen und Routenbeschreibungen angefordert, die im (div-) Element `beschreibung` angezeigt werden soll.

Über dieses Skript werden die oben skizzierten Anforderungen umgesetzt.

Festlegung des Aussehens der Routenbeschreibung

Als Letztes bleibt noch übrig, die Anzeige der `beschreibung` etwas zu gestalten. Beim div-Tag mit der id `beschreibung` wurde festgelegt, dass das div-Tag nach den Regeln der class `beschreibung` formatiert werden soll. In einer CSS-Datei, die bei den Serverressourcen angelegt werden kann, wird folgender CSS-Code eingetragen.

```
.beschreibung{  
font-size:12px;  
font-family:sans-serif;  
padding:10px;  
}
```

Die hinterlegte Definition bezieht sich auf die Klasse `beschreibung`. Dem div-Element mit der id `beschreibung` wurde ebenfalls die class `beschreibung` zugewiesen. Daher wird bei der Darstellung der Alternativen-Box und der textlichen Routenbeschreibung diese (rudimentäre) Gestaltung verwendet.

Zusammenfassung

In diesem Beispiel konnte gezeigt werden, wie Benutzereingaben, JavaScript-Verarbeitung, Anzeige und Gestaltung zusammenspielen können.

Letztlich sollte mit diesem Beispiel gezeigt werden, dass es in TD Mobile keine Einschränkungen hinsichtlich der individuellen Anpassung einer Oberfläche gibt. Sollten die im Standard angebotenen Container und Objekte nicht ausreichen, so kann der Entwickler seine Umgebung durch andere jQuery Mobile kompatible Controls beliebig erweitern.

MD Consulting & Informationsdienste GmbH

Michaelisstraße 13a
99084 Erfurt

Berghamer Straße 14
85435 Erding

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