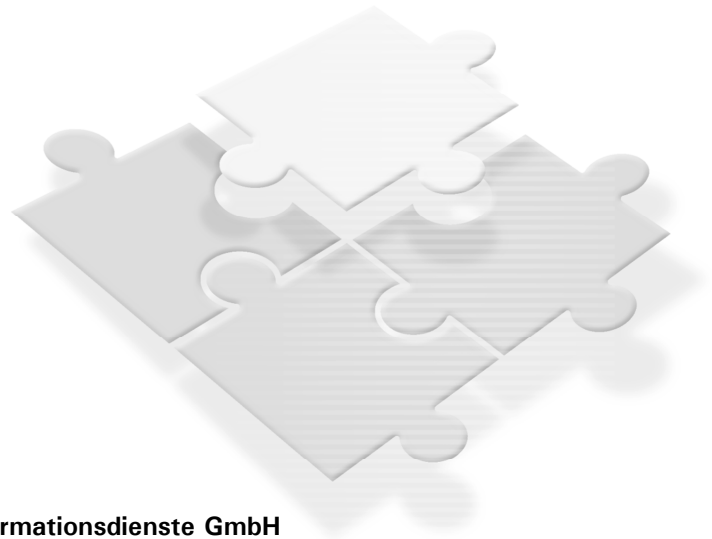


Accordion Control in TD Mobile-Anwendungen



MD Consulting & Informationsdienste GmbH

www.md-consulting.de

Michaelisstraße 13 a
99084 Erfurt
03 61 / 5 65 93-0

Berghamer Straße 14
85435 Erding
0 81 22 / 97 40-0

info@md-consulting.de

Einleitung

In TD Mobile werden in der Regel Daten aus der Datenbank nach der Ausführung einer SELECT-Anweisung an einen clientseitig definierten Array eine user defined variable (UDV) als „Bindung“ übergeben. In diesem Papier wird beschrieben, was unternommen werden muss, wenn man die Ergebnismenge einer Bindung an ein nicht zum Lieferumfang von TD Mobile gehörendes HTML (5) Control übergeben will. In diesem Beispiel handelt es sich um das Accordion-Control, welches zwar zum Umfang von JQuery mobile, aber nicht zum Lieferumfang von TD Mobile gehört.

Standard-Vorgehensweise

Standardmäßig müssen folgende Arbeitsschritte vorgenommen werden, um eine komplexe Ergebnismenge einer SELECT-Anweisung an den Client zu übertragen:

Definition einer Klasse, um die Struktur der SELECT-Anweisung abzubilden

Instanziierung der Klasse für die Ergebnismenge und für jeden einzelnen Datensatz

Codierung der Operation, um die Bindungen mit Werte zu versorgen

Zuweisung der Bindungen an einen Listen-Container

Definition einer Klasse

In diesem Beispiel sollen lediglich zwei Strangs – *Titel* und *Inhalt* – aus einer Tabelle selektiert werden. Daher wird eine Klasse *clsObjekt* mit zwei String-Instanzvariablen *Titel* und *Inhalt* angelegt.

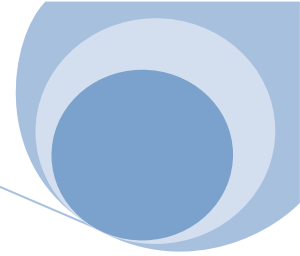
Funktionale Klasse: clsInhalt

```
...
Instanz Variablen
  String: Titel
  String: Text
  String: Kommentar
...
```

Source Code 1: Definition der Klasse clsInhalt

Instanziierung der Klasse als Bindung(en)

Die unter dem Namen *clsObjekt* angelegte Klasse wird auf der Beispielseite *page1* zwei Mal instanziiert: als Instanz eines einzelnen Datensatzes mit dem Namen *EINOBJEKT* und als



Instanz der Ergebnismenge *DIEOBJEKTE[*]*. Mit dem Stern in eckigen Klammern wird festgelegt, dass es sich um einen Array handelt.

Webseite: page1

```

...
Client
  Bindungen
    clsInhalt: EINOBJEKT
    clsInhalt: DIEOBJEKTE[*]
...

```

Source Code 2: Instanziierung der Klasse als Bindung(en)

Codierung einer Operation

In der Operation wird als Rückgabewert ein Array der Klasse *clsInhalt* angelegt. Um festzulegen, dass der Rückgabewert ein Array ist, wird hinter *clsInhalt: [*]* angegeben.

In der Operation wird als lokale Variable ein Array der Klasse *clsInhalt* *clsInhalt: oObjekt[*]* eingegeben.

Operation: ObjekteLesen

```

...
Returns
  clsInhalt: [*]
  Binding: DIEOBJEKTE
Lokale Variablen
  clsInhalt: oObjekt[*]
  Sql Handle: hSql
  String: sStatement
  Number: nReturn
  Number: n
Aktionen
  If fnConnect(hSql)
    Set sStatement = "select firma_1, thema_1, Kommentar_1 from angebote a, kunden k
    where a.knr = k.knr and angebdat = sysdate
    INTO :oObjekt[n].Titel, :oObjekt[n].Text, :oObjekt[n].Kommentar"
    If SqlPrepareAndExecute(hSql, sStatement)
      While SqlFetchNext(hSql, nReturn)
        Set n = n + 1
      If SqlCommit(hSql)
        If SqlDisconnect(hSql)
      Return oObjekt

```

Source Code 3: die Operation ObjekteLesen

In der SELECT-Anweisung sind die INTO-Variablen als dimensionierte Variable angelegt. In der While-Schleife braucht somit lediglich die Laufvariable n erhöht zu werden, um die einzelnen Datensätze dem Array zuzuweisen.

Am Ende des Codes wird mit Return das Array-Objekt insgesamt zurückgegeben.

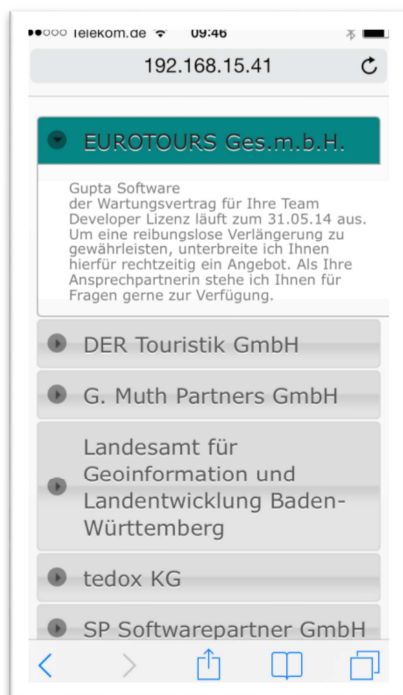
Das Accordion-Control als Visualizer

Das Accordion Control gehört zum Umfang von JQuery Mobile. Allerdings ist diese Control (noch) nicht im Lieferumfang von TD Mobile integriert. Anders formuliert: das Accordion steht zwar grundsätzlich auch in TD Mobile zur Verfügung, allerdings müssen einige Arbeitsschritte ausgeführt werden, um

- Das Control einzubinden und
- Das Control mit Daten (aus der Datenbank) zu versorgen und
- Die Gestaltung der Daten im Accordion vorzunehmen

Das Accordion-Control eignet sich besonders, um eine Fülle von Informationen auch auf kleineren Displays wie beispielsweise einem Smartphone anzuzeigen, weil immer nur der ausgewählte Eintrag und seine Daten angezeigt werden. Wird ein anderer Eintrag ausgewählt, wird automatisch der vorher ausgewählte Eintrag zusammengezogen. Durch diese Funktionalität unterscheidet sich das Accordion beispielsweise vom Expander, der zwar auch einen Bereich auf- und zuklappen kann, bei dem es allerdings keinen Zusammenhang zu anderen „Bereichen“ gibt.

Die Oberfläche der Beispielanwendung sieht in etwa folgendermaßen aus:



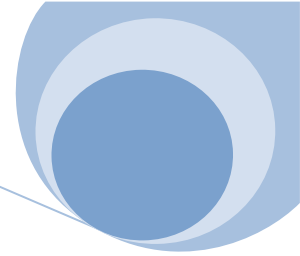


Abbildung 1: Die Oberfläche der Beispielanwendung

Integration des Accordions

Der nachstehende Code wird im Fenster HTML eines (TD Mobile) HTML-Controls hinterlegt.

```
<link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
<script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
```

```
<div id="accordion" class="container">
</div>
```

Source Code 4: Integration des Accordion

Es wird somit ein Stylesheet und eine Javascript-Bibliothek aus der Version 1.10.4 von JQuery referenziert. In einem Div-Container wird angegeben, dass das Objekt die ID *accordion* hat und ein Container-Objekt ist.

Das Skript DatenAnzeigen

Bei der Generierung der Seite wird zunächst die Operation *ObjekteLesen* ausgeführt, sodass nach Beendigung der Operation clientseitig die Bindung *DIEOBJEKTE[*]* mit der Ergebnismenge der SELECT-Anweisung „gefüllt“ ist.

Im Skript *DatenAnzeigen* wird die Bindung *DIEOBJEKTE* verarbeitet und die Daten dem Accordion-Control zugewiesen.

```
var dieTitel = Td.Data.Binding.get("DIEOBJEKTE");
var items = dieTitel.getValue();

$.each( items, function( key, value ) {
    $('#accordion').append('<h3><a href="#">' + value.Titel + '</a></h3><p>' + value.Text + '<br>' +
value.Kommentar + '</p>');
});
$('#accordion').accordion();
```

Source Code 5: das Skript DatenAnzeigen

Die Daten aus der Bindung – alle Datensätze der Ergebnismenge – sollen innerhalb des Accordion-Controls angezeigt werden. Dazu muss die Bindung *DIEOBJEKTE[*]* innerhalb des Skripts zur Verfügung stehen, um dann in einer Schleife die Ergebnismenge in die angezeigte Seite zu integrieren.

In der ersten Zeile wird eine Funktion aus der TD Mobile API verwendet, um die Bindung im Skript unter dem Namen *dieTitel* verfügbar zu machen. In der zweiten Zeile wird der Inhalt

der Bindung *dieTitel* der Javascript Variablen *items* zugewiesen: der Inhalt der Bindung DIEOBJEKTE[*] steht nun in der Variablen *items* zur Verfügung.

Die zu JQuery gehörende Funktion `.each` bietet die Möglichkeit, eine Schleife (über alle Bereiche eines Arrays) auszuführen. Die JQuery-Funktion `.each` hat zwei Parameter: einen Datenarray und eine Funktion mit wiederum zwei Parametern *key* und *value*, wobei *key* eine Laufvariable ist und *value* ein Tupel des Arrays (items).

Was die Funktion `.each` zu tun hat, wird in geschweiften Klammern `{}` angegeben. Inhaltlich sollen in der Funktion `.each` für jedes Element des Arrays der Reiter des Accordions mit dem `h3`-Tag, die Überschrift innerhalb Absatzes `<p></p>` gesetzt werden und darunter `
` der Inhalt der Variable Kommentar angezeigt werden. Die Werte aus der Bindung DIEOBJEKTE[*] sollen somit in HTML-Code „eingebettet“ werden, um diesen HTML-Code in den bereits angelegten Div-Container mit dem Namen `accordion` einzufügen.

Mit der Anweisung `$('#accordion')` wird in JQuery die Id eines HTML-Objekt angesprochen (JavaScript `document.getElementById("...")`) und die Funktion `Append (...)` bettet ein entsprechendes HTML-Snipplet in den angesprochenen Div-Container ein.

Das HTML-Snipplet, dass in den Div-Container `accordion` eingebettet werden soll, hat in diesem Beispiel folgenden Aufbau:

```
'<h3><a href="#">' + value.Titel + '</a></h3><p>' + value.Text + '<br>' + value.Kommentar + '</p>'
```

Source Code 6: das HTML-Snipplet

Innerhalb der Funktion werden die „Bestandteile“ der UDV *clsObjekt* durch einen Punkt und den Namen ermittelt: `value.[wert aus der Bindung]`.

Nachdem die Schleife `.each` beendet ist, sind alle Werte aus der Bindung DIEOBJEKTE[*] in den HTML-Code der Seite eingebettet.

Mit der letzten Anweisung der JavaScript-Funktion `DatenAnzeigen` werden die Tags im Div-Container „gestaltet“.

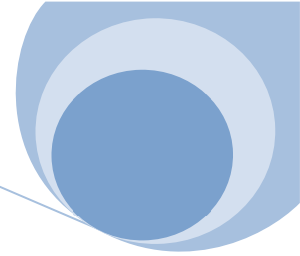


Abbildung 2: die Anwendung (ohne Gestaltung)

Die Anwendung funktioniert insofern, dass die Daten aus der Bindung `DIEOBJEKTE[*]` in ein Accordion-Control eingebettet worden. Allerdings sieht die Oberfläche noch nicht so aus, wie ich mir das vorgestellt habe:

- Der Hintergrund des ausgewählten Eintrags soll in der Firmenfarbe von MD Consulting hinterlegt werden und
- Die Anzeigehöhe eines Eintrags soll sich nach der Länge des Textes für jeden Eintrag dynamisch anpassen. Ein kurzer Eintrag nimmt dann weniger Fläche in Anspruch als ein längerer Eintrag.

Die Gestaltung des Accordion-Controls wird über eine eigene CSS-Datei vorgenommen, die in das Projekt eingebunden wird. Die beiden Einträge sind in der CSS-Datei enthalten:

```
.....
.accordion .ui-accordion-content {
  background-color: rgba(0, 0, 0, 0);
  color: #777777;
  font-size: 9pt;
  line-height: 10pt;
  width: 90%;
  height: auto !important;
}
.ui-accordion .ui-accordion-header-active {
  background: none repeat scroll 0 0 #008080;
  color: #010101 !important;
}
```

```
height: auto !important;  
}
```

Source Code 7: CSS-Datei des Projekts

Es soll der Inhalt eines Accordion-Reiters und der aktivierte `<h3>`-Bereich – die Überschrift eines Accordion-Reiters – gestaltet werden.

- In der Accordion-Komponente `ui-accordion-content` wird durch `height:auto` mit dem Zusatz `!important` die Standard-Eigenschaft des Accordions überschrieben und damit die „Höhe“ des Eintrags dynamisch an seinen Inhalt angepasst.
- Der Hintergrund der Accordion-Komponente `ui-accordion-header-active` wird die Farbe `#008080` zugewiesen – hierdurch wird sichergestellt, dass der „Reiter“ eines Accordions-Eintrags eine herausragende Farbe erhält.

Zusammenfassung

Anhand des Accordion-Beispiels sollte gezeigt werden, wie man vorgehen muss, um andere, nicht zum Lieferumfang von TD Mobile gehörende Visualizer (Objekte oder Container) in eine TD Mobile-Anwendung zu integrieren.

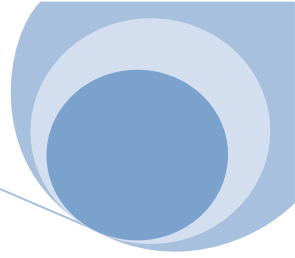
Das entsprechende Control muss als HTML-Objekt angelegt und mit den notwendigen Referenzen auf entsprechende Stylesheets und JavaScript-Bibliotheken definiert werden.

Um dieses HTML-Objekt innerhalb TD Mobile ansprechen zu können, muss ein entsprechender DIV-Container angelegt und mit einer Id versorgt werden.

Über diese Id wird das Control innerhalb der TD Mobile-Anwendung „angesprochen“.

Zur Laufzeit der Anwendung müssen die gewünschten Daten in der Regel über eine Operation an eine clientseitige Bindung übergeben werden. In einer eigenen JavaScript-Funktion wird dann über die TD Mobile API die Bindung in die JavaScript-Funktion integriert.

Bei Array-Bindungen wird dann die JQuery-Funktion `.each` verwendet, um in einer Schleife die einzelnen Bindungswerte in ein entsprechendes HTML-Snipplet zu integrieren. Wenn auf diese Weise der HTML-Code der Seite „erweitert“ wurde, muss die Visualisierung der Daten in der Darstellungsform Accordion vorgenommen werden. Dieses geschieht mit der Anweisung `$("#accordion").accordion();`



Die Einbindung der Daten aus der Bindung funktioniert – wenn nun auch noch eine geänderte Anzeige – abweichend vom Standard – gewünscht wird, so kann das über die Einbettung entsprechenden CSS-Klassen in einem eigenen Stylesheet erfolgen.

Analog zu der beschriebenen Vorgehensweise ist es möglich, in TD Mobile prinzipiell alle HTML-Controls einzubinden wie Charting-Controls, das Google Maps-Control, usw.

In dieser Dokumentation haben wir lediglich versucht, Bindungsdaten anders zu visualisieren als das im TD Mobile Standard möglich ist.

Es wurde noch nicht dokumentiert, wie beispielsweise Ereignisse in diesen Controls abgefangen und mögliche Eingaben in diesen Controls wieder an Bindungen in TD Mobile zurückgegeben werden können. Das hätte den Scope dieses Papiers gesprengt.