Vorlesungsreihe
EwA

Javascript –Frameworks am Beispiel von JQuery

Prof. Dr.-Ing. Thomas Wiedemann
email: wiedem@informatik.htw-dresden.de

HOCHSCHULE FÜR TECHNIK UND WIRTSCHAFT DRESDEN (FH)
Fachbereich Informatik/Mathematik

Gliederung

- JavaScript-Frameworks
  - Einführung / Überblick / Aufgaben
  - Aktuelle Optionen / Vergleich
- JQuery – Grundlagen
  - Selektoren
  - Ajax – Einbindung
  - Überblick zu GUI-Toolkits

Quelle(n) (und weitere im Text):

[1] http://jquery.com/ **(JQuery –Homepage)**

[2] http://visualjquery.com/ **(Beispiele , Demos und Syntax)**

[3] http://www.w3schools.com/jquery/

Übersicht zu Javascript-Frameworks I

- JavaScript wird in den Browsern zur Client-seitigen Manipulation der Webseite verwendet –> **bildet die Basis für Web 2.0 und RIA**
- ist keine direkte Ableitung von Sun´s JAVA, sondern eine Eigenentwicklung von Netscape aus dem Jahre 1995
- Entwicklung ist auch heute noch hauptsächlich Browser-Hersteller-getrieben

Aktuelle Hauptprobleme sind Inkonsistenzen in den Browsern :

- bei der generellen Verfügbarkeit von einzelnen Funktionen
- beim Timing und der zeitlichen Verfügbarkeit beim Aufbau der Seite
- bei der entstehenden Komplexität zur Bewältigung größerer Aufgaben unter Berücksichtigung der ersten beiden Problemstellen
- **Ziel der JS-Frameworks** ist die möglichst einheitliche Behandlung aller Browser (sowohl Typ wie auch Version)

Übersicht zu Javascript-Frameworks II

- aktuell existieren ca. 20 aktive JavaScript –Frameworks nach

  http://en.wikipedia.org/wiki/Comparison_of_JavaScript_frameworks

- Die 3 aktivsten und bekanntesten Frameworks sind
  - **JQuery (Version 1.11.x oder 3.x - 2016)**
  - **DoJo (Version 1.10.x 2014)**
  - **MooTools (Version 1. Februar 2015)**
- Ein aktueller Vergleich nach *[Schreier 2012 (Diplomarbeit)]* zeigt keinen absoluten Sieger, JQuery liegt jedoch sowohl bei den Nutzerzahlen sehr deutlich und bei der Performance deutlich vorn (31ms zu 39ms zu 43ms für Referenzaufgabe)
- Alle 3 Frameworks verfügen über relativ umfangreiche Zusatz-Toolkits und GUI-Bausteine, erreichbar von den Homepages.
- Nachfolgend soll deshalb JQuery genauer vorgestellt werden ,die Syntax und Funktionsweise der anderen Frameworks ist ähnlich!

JQuery – Überblick

Historie

- Entstehung 2006 auf einem Barcamp in New York
- verantw. Entwickler John Resig
- Motto : **„write less, do more“**

Softwaretyp – Opensource unter

- MIT – Lizenz : frei verfügbar auch für geschlossene SW, falls Copyright-Vermerk in allen Kopien vermerkt wird

Ausblick :

- sehr gute Akzeptanz, aktuell auch Einsatz durch MS in VisualStudio und anderen IDE´s
- JS-Frameworks wie auch JQuery könnten mit besserer (einheitlicher) Integration von JS in die Browser wieder an Bedeutung verlieren (?)

JQuery – Installation / Einbindung in eigene Projekte

Beschaffung

- Download von JQuery-Homepage **http:jquery.com**

Achtung: größere Änderungen ab Version 1.9

Ab Version >= 1.9. kein SUPPORT für IE 8.0 (2008…2011) und älter !

Versionen – aktuell 1.11 oder 3.x entweder als

- Vollversion (gut kommentiert) – ca. 300 Kbyte
- minimierte Version (ohne Kom./gepresst) - ca. 100 Kbyte
- Achtung: Umfang hat Einfluss auf erste Ladezeit !

Alternativ auch direkte Einbindung von externen Servern :

- z.B. von MS oder Google (ggf. günstiger durch Browserchache) :

http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js

Liste: https://developers.google.com/speed/libraries/devguide#jquery

Einbindung von JQuery in HTML-Seiten

- Einbindung in HTML oder auch Script-Seiten (*.php etc.)
- Link zu JQuery-Library im Header der HTML-Seite, ggf. auch zu umbenannter Datei **jquery.js**

```
<html><head>
<script   src="jquery-2.1.min.js "></script>   ...
```

- oder Link zu externer Jquery-Library im Header der HTML-Seite

```
<html><head>
<script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery.min.js" >
</script> “ > </script>   ...
```

Generelle Vorgehensweise bei JQuery

```
<html><head>
<script   src="jquery.js"> </script>

<script type="text/javascript">
$(document).ready(function(){
  $("button").click(function(){
    $("p").css("background-color","yellow");
  });
});
</script>
</head>

<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button  Onlick="… ">Click me</button>
</body> </html>
```

- Einbindung JQuery
- spezielle ready-Funktion, welche erst nach Laden des DOM die Funktionen bereitstellt und die Events an den HTML-Code anbindet
- $ - ist Kurzform für Jquery-Aufruf
- **KEIN JS im HTML-Code !!!!!!!**

Adressierung im DOM mit JQuery – Selectoren

- Eine besondere Stärke von JQuery sind die Selektoren, welche GLEICHZEITIG eine Menge von HTML-Objekten ansprechen können :

Elementselektoren (jeweils mit Angabe des HTML-Tags)

- $("p") wählt ALLE <p> Elemente aus
- $("p.intro") wählt <p> - Elemente mit class="intro“ aus
- $("p#demo") wählt <p> - Elemente mit with id="demo".
- **Die Syntax # oder . entspricht der analogen CSS-Syntax :**

  .intro { … } #demo { }

Attribut-Selektoren (jeweils mit Angabe des Attributes, analog zu XPATH)

- $("[href”]) wählt ALLE Elemente mit href - Attribut aus
- $("[href='#']") wählt ALLE Elemente mit href - Attribut =“#”
- $("[href!='#']") analog mit ungleich “#”
- $("[href$='.jpg']") wählt href´s mit .jpg am Ende aus

JQuery – Manipulatoren

- Nach erfolgter Adressierung, dabei auch mehrere Elemente gleichzeitig, können deren Eigenschaften über entsprechende Selektoren geändert werden.

CSS-Selektoren (jeweils mit Angabe der CSS-Eigenschaften)

- $("p").**css**("background-color","yellow");
- **Die Syntax in der Klammer ist immer CSS-Attribut , neuer Wert !**

- **Elemente können versteckt oder angezeigt werden :**
  - **$("p").hide(); // Verstecken**
  - **$("p").hide(1000); // Verstecken langsam**
  - **$("p").show(); // Anzeigen**
  - **$("p").show(1000); // Anzeigen langsam**
  - **$("p").toggle(); // wechselweise …**

JQuery – Manipulatoren II

Weitere Optionen sind :

Einblendungen

$(selector).fadeIn(speed,callback) - langsam Einblenden

$(selector).fadeOut(speed,callback) - langsam Ausblenden

$(selector).fadeTo(speed,opacity,callback) - Überblenden

Animationen

$(selector).animate({params},[duration],[easing],[callback])

Beispiel: $("div").animate({height:300},"slow");

- Es existieren weitere Manipulatoren in JQuery
  -> bitte Doku lesen oder zusätzliche JQuery - Plugins installieren !

JQuery – Events

Neben dem Click – Event sind die anderen JS-Events auch ansprechbar:

$(document).ready(function) - bindet Funktionen an Ready-Event
$(selector).click(function) - bindet an Click-Event des Elements
$(selector).dblclick(function) - analog Doppel-Click-Event
$(selector).focus(function) - bei Fokuserhalt
$(selector).mouseover(function) - bei Mouse-Over

- Alle anderen Events entsprechen ebenfalls meist der JS-Syntax -
  -> bitte Doku abrufen !

JQuery – Anbindung von Funktionen zu Events (event handlers)

Auch zur Laufzeit sind Event-Funktionsanbindungen möglich mit der Methode **on** (ältere (deprecated) version: delegate) :

```
$("#b4").on("click", function(){
        alert('New button event');
        $("p").css("background-color","pink");
    });
   alert ("event added");
});
```

Volle syntax **:**

$(selector).on(event,childSelector,data,function,map)

- event – type (Required) – like “click” see previous page
- childSelector (Optional) if only to child elements
- data (Optional) for additional data to pass along to the function
- Function (Required) - specifies the function to run when the event occurs
- map - specifies an event map ({event:function, event:function, ...})

JQuery – HTML – DOM – Manipulation

Neben der Änderung von Elementen kann auch der gesamte DOM-Baum manipuliert werden :

-> Einfügen von neuem HTML –Code mit :

$(selector).html(“<B>Fett</B>) - ersetzt alten durch neuen Text

$(selector).append(“<B>Fett</B>) - hängt neuen Text innerhalb an

$(selector).after (“<B>Fett</B>) - hängt neuen Text danach an

Weitere DOM-Operationen :

Löschen mit : remove(), empty()

Formatierung ändern mit : wrap() , unwrap()

CSS-Manipulation und -Test mit : addclass(), hasclass()

JQuery – AJAX

JQuery vereinfacht auch Einbinden von AJAX –Funktionen sehr stark und vermeidet inesbesondere die Browserweichen :

```
<html><head><script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$("button").click(function(){
  $.ajax({url:"test1.txt",   success:function(result)  {
                                        $("div").html(result);
  }});      });});
</script></head><body>

<div><h2>Let AJAX change this text</h2></div>
<button>Change Content</button>
</body> </html>
```

Jquery – Plugins und Demos

Repository of JQuery-plugins

- http://plugins.jquery.com
- for complex forms : JQueryUI at http://jqueryui.com/ see http://jqueryui.com/demos/
- for responsive design on mobile devices
  - Bootstrap (from http://getbootstrap.com/)

JQuery Demos

- Search for „Jquery examples“ –> actual sites :
  - http://www.jqueryrain.com/
  - http://www.noupe.com/development/50-amazing-jquery-examples-part1.html

Zusammenfasssung

- JQuery hat eine sehr erfolgreiche Entwicklung absolviert
- Für WEB 2.0 – Anwendungen ist ein Einsatz sehr empfehlenswert, da sich
  - **Der Aufwand und die Sicherheit deutlich erhöhen**
  - **Neue Browserversionen sind wesentlich schneller ansprechbar und nutzbar !**
- Interessant wird die Konkurrenz von HTML 5, wobei eine gegenseitige Befruchtung zu erwarten ist :
  - positive JQuery-Funktionen wandern nach HTML 5.0,
  - JQuery bekommt neue Optionen im HTML 5 – DOM !
- In analoger Weise gilt dies auch für andere JS-Frameworks!