NOTES & DOMINO ENTWICKLERCAMP

assono

Mit dojo & XPages

zu einer erstklassigen User-Experience

EntwicklerCamp
22. Februar 2011

Innovative Software-Lösungen.

www.assono.de

Bernd Hort

Diplom-Informatiker, Universität Hamburg

seit 1995 entwickle ich Lotus Notes Anwendungen

IBM Certified Application Developer, System Administrator & Instructor

Sprecher auf diversen Konferenzen & Lotusphere 2008

bhort@assono.de
http://www.assono.de/blog
040/73 44 28-315

assono
IT-Consulting & Solutions

Agenda

- Motivation
- Einführung in dojo
- dojo und XPages
- Dialoge in XPages
- dojo Widgets in XPages
- DataGrids
- dojo Charting
- Fragen & Antworten

Rich User Experience

Motivation – Warum dojo?

Open Source

Browserunterstützung SUPPORTING

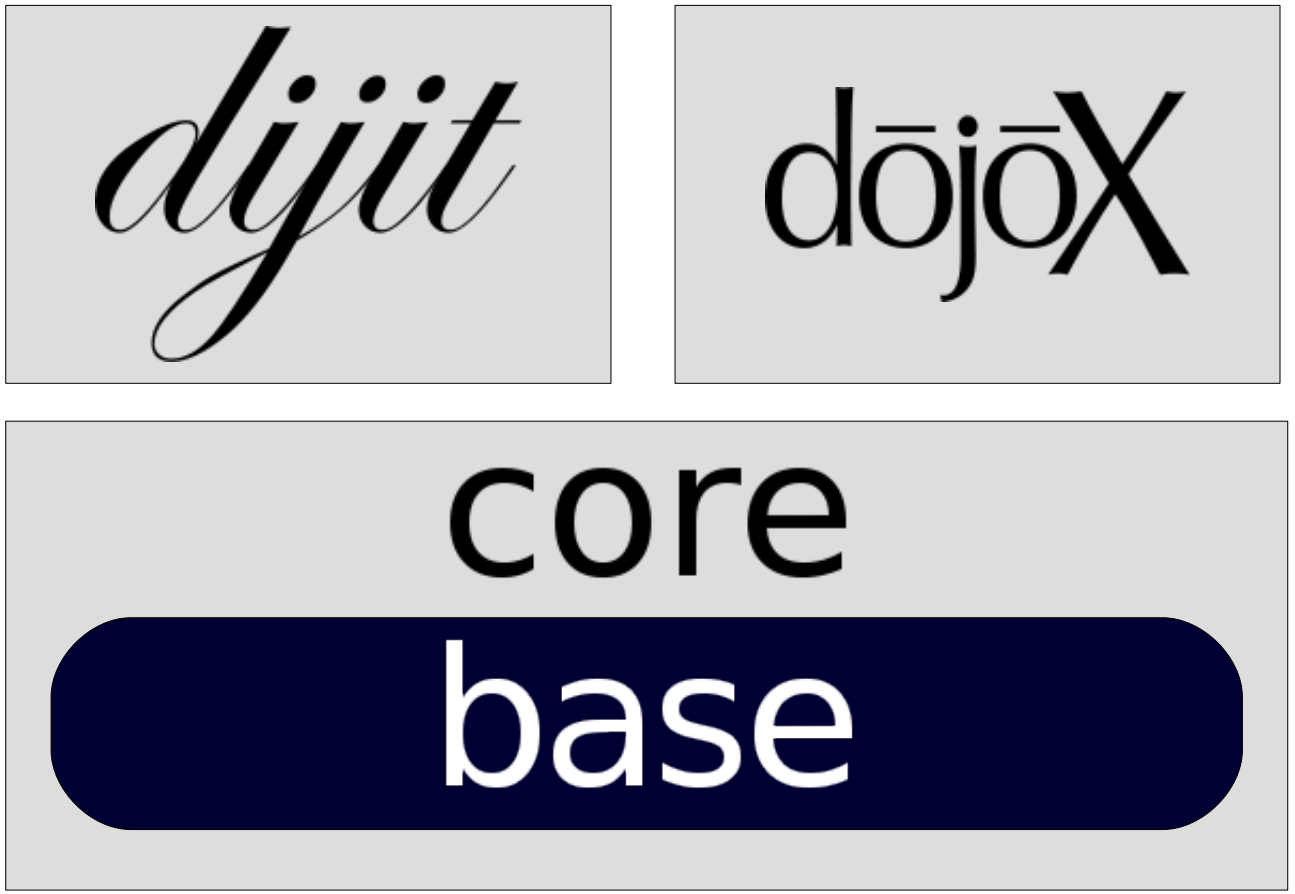
Vielfältige UI Elemente

Internationalisierung

Integraler Bestandteil von Xpages

Performance

Dojo besteht aus drei Teilen



core - base

- Browser-Erkennung
- JSON encoding / decoding
- Laden von Packages
- Eventhandling
- Animation effects
- AJAX
- CSS Utilities
- OOP Unterstützung
- Firebug Integration

base

core – Ergänzung

- Drag & Drop
- I18N Support
- Datumsformatierung
- Zahlenformatierung
- String Utils
- Cookie Handling
- Extended Animations
- Back Button Handling

core

dijit

- Oberflächen Elemente (interface widgets)
- „Theme“ Unterstützung
- Internationalisierung
- Keyboard Unterstützung

dojoX

- „quasi“ experimentell -
  „The future, today“
- Vieles stabil und produktiv einsetzbar
- Charts
- SVG Unterstützung
- DojoX Offline – Integration mit Gears
- DojoX Widgets

«Page»
dojo-Hello World

Hello World

```
<html>
  <head>
    <title>My First Dojo App</title>
    <link rel="StyleSheet" type="text/css"
      href="/domjs/dojo-1.3.2/dojo/resources/dojo.css">
    <link rel="StyleSheet" type="text/css"
      href="/domjs/dojo-1.3.2/dijit/themes/tundra/tundra.css">

    <script type="text/javascript">
      var djConfig = {
        baseScriptUri : "/domjs/dojo-1.3.2/",
        parseOnLoad : true
      };
    </script>

    <script type="text/javascript" src="js/dojo/dojo/dojo.js"></script>
    <script language="JavaScript" type="text/javascript">
      dojo.require("dojo.parser");
      dojo.require("dijit.form.DateTextBox");
    </script>

  </head>
```

Basis CSS

Theme CSS

Basis Konfiguration
muss vor dem Laden von dojo.js erfolgen!

dojo.js (Base)

Nachladen
von Packages

Hello World

```
  <body class="tundra">

    <form action="" method="get">

      <input type="text" name="date1" id="date1"
        value="2011-02-22"
        dojoType="dijit.form.DateTextBox"
        required="true" />

    </form>

  </body>
</html>
```

DateTextBox

base – dojo.addOnLoad()

base

- Registriert eine Funktion, die aufgerufen wird, sobald das DOM geladen ist und alle Widgets intialisiert sind

- `dojo.addOnLoad(functionPointer);`
  Einfacher Funktionsaufruf

- `dojo.addOnLoad(object, "functionName");`
  Aufruf einer Methode eines Objektes

- `dojo.addOnLoad(object, function() { /* ... */});`
  Definition einer neuen Methode in dem Objekt

base – CSS Klassen

base

- `dojo.addClass(DomNode|NodeId, ClassName)`
  fügt dem DOM-Knoten eine CSS-Klasse hinzu
- `dojo.removeClass(DomNode|NodeId, ClassName)`
  entfernt eine CSS-Klasse von einem DOM-Knoten
- `dojo.hasClass(DomNode|NodeId, ClassName)`
  prüft einem DOM-Knoten eine CSS-Klasse zugewiesen wurde

base – DOM Knoten

base

- `dojo.byId(NodeId)`
  liefert den DOM-Knoten mit der gegebenen Id zurück
- `dojo.clone(Object)`
  klont ein Objekt, auch DOM-Knoten, inklusive aller Kinder
- `dojo.isDescendant(ChildDomNode, ParentDomNode)`
  prüft einem DOM-Knoten Kind eines anderen DOM-Knoten ist

base – dojo.query() (1)

base

- `dojo.query(CSS3-Selector)`
  liefert alle DOM-Knoten, die dem CSS3-Selektor entsprechen
  - Klassenname, z.B. „.foo“
  - HTML-Elemente, z.B. „span“
  - CSS Hierarchien, z.B. „table div“
  - Direkte Kindelemente (>), z.B. „#tabular_data > div“
  - Universal Selektor (*)
  - Unmittelbar vorangestellte Geschwisterknoten (~)
  - vorangestellte Geschwisterknoten (+)

base – dojo.query() (2)

base

- `dojo.query(CSS3-Selector)`
  Abfrage nach Attributen
  - [foo] Attribut ist vorhanden
  - [foo='bar'] Attributwert stimmt genau überein
  - [foo~='bar'] Attributwert stimmt mit einem Eintrag in der Liste der Werte überein
  - [foo^='bar'] Attributwert beginnt mit
  - [foo$='bar'] Attributwert endet mit
  - [foo*='bar'] Attributwert enthält

base – dojo.query() (3)

base

- `dojo.query(CSS3-Selector)`
  Pseudoklassen
  - :first-child – erstes Kind
  - :last-child – letztes Kind
  - :only-child – nur ein Kind
  - :empty – kein Kind
  - :checked – aktivierte Radiobuttons & Checkboxes
  - :nth-child(n) – n-te Kind
  - :nth-child(even) – alle „geraden“ Kinder
  - :nth-child(odd) – alle „ungeraden“ Kinder
  - :not(...) - Umkehrungen

base – Aspektorientierte Programmierung – connect

base

- Ruft eine Methode / Funktion auf, sobald ein Event ausgelöst wurde oder eine Methode eines anderen Objektes aufgerufen wurde.

```
dojo.connect(object: Object|null,
event: String, context: Object|
null, method: String|Function);
```

- Verknüpfung mit DOM-Knoten

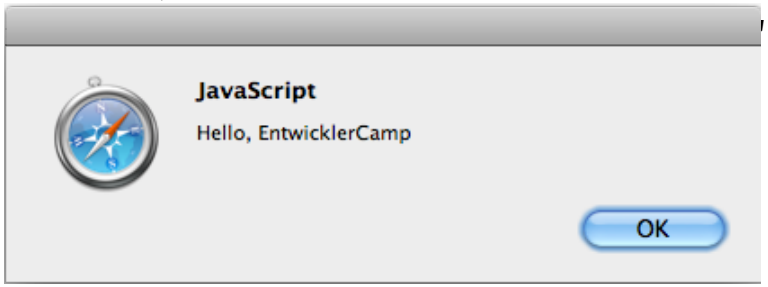
```
dojo.connect(dojo.byId(„foo“),
„onmouseover“, function(evt)
{console.log(evt)});
```

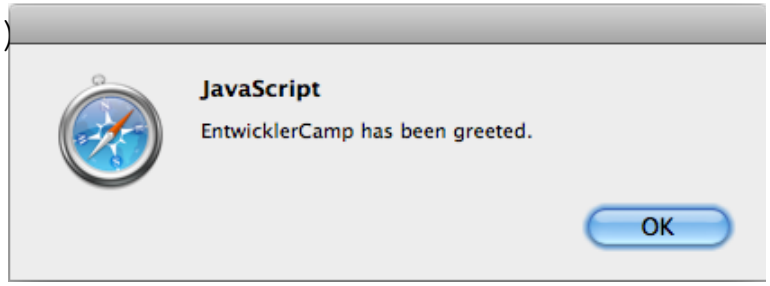
base – Aspektorientierte Programmierung – connect

- Das Event kann auch einfach ein Methodenaufruf eines anderen Objektes sein.

base

```
function MyClass(){
  this.sayHello = function(inName){
    alert("Hello, " + inName)}
}
function AnotherClass(){
  this.echo = function(message){
    alert(message + " has been greeted.")}
}
foo = new MyClass();
bar = new AnotherClass();
var handle = dojo.connect(foo, "sayHello", bar,
"echo");
```



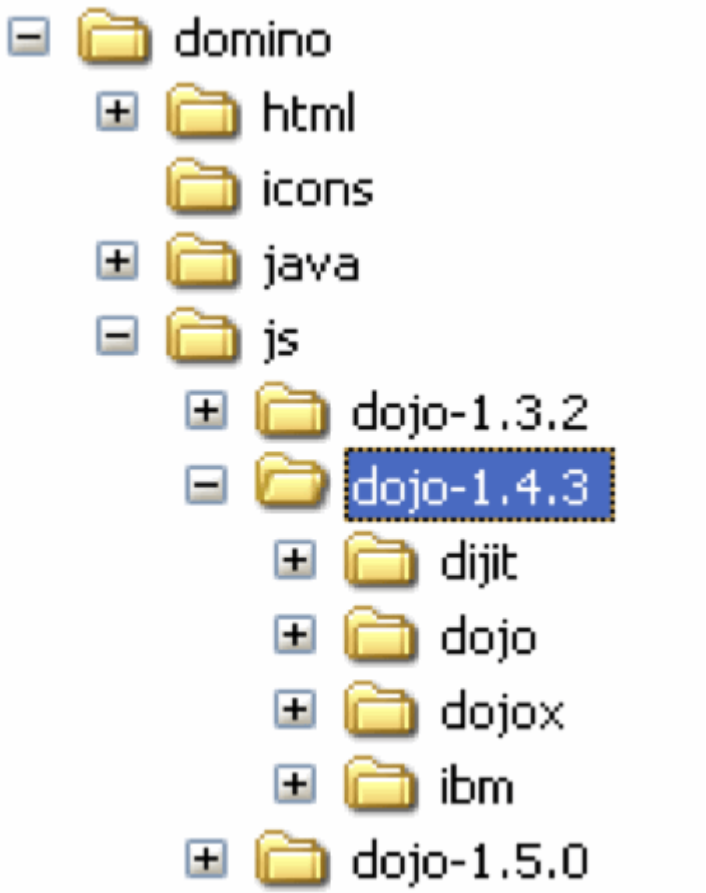


Dojo in XPages

Dojo in XPages

- Verzeichnis „js“ unterhalb von {Domino Data}/domino
- Referenziert als /domjs/
- 8.5.0 = Dojo 1.1.1
- 8.5.1 = Dojo 1.3.2
- 8.5.2 = Dojo 1.4.3
- 8.5.3 = Dojo 1.5.0 *)

*) Beta – kann sich noch ändern (theoretisch)



Dojo in XPages – Standardelemente

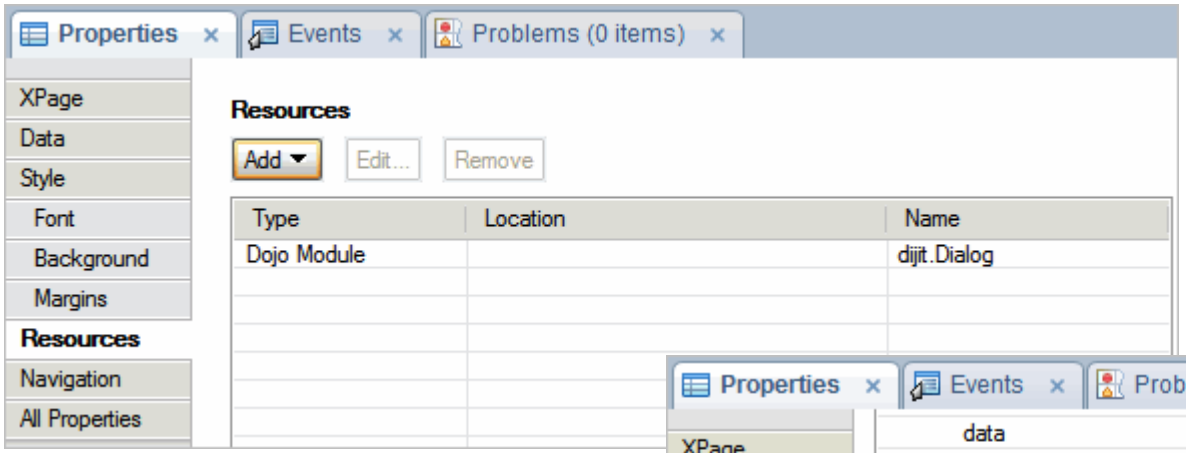
- Dojo ist integraler Bestandteil von XPages
- Partial Refresh
- Type ahead
- Date/Time Picker
- Richtext Editor in den Versionen 8.5 und 8.5.1
- XSP-Object in Clientside JavaScript

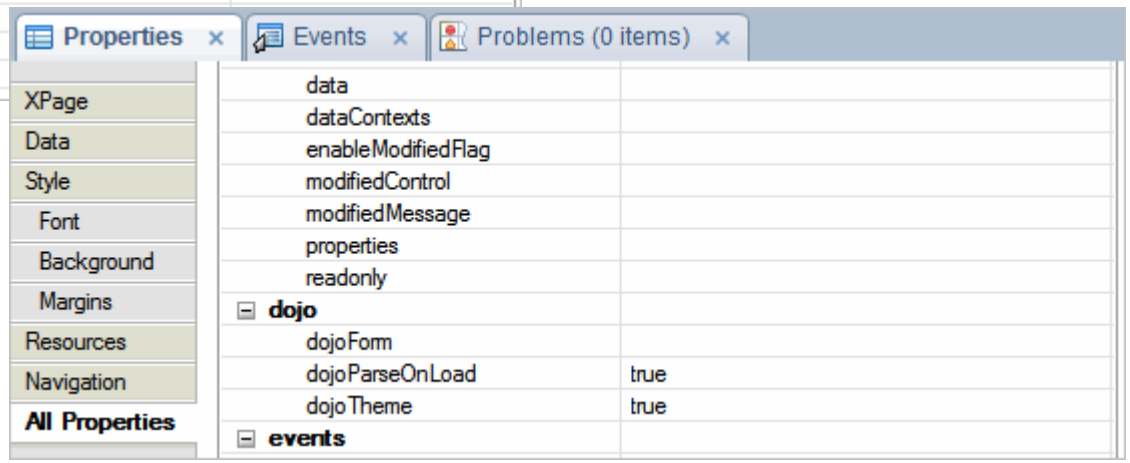
Zugriff auf HTML-Elemente via ID

- Dojo arbeitet vielfach der ID eines HTML-Elementes
- XPages erzeugen die IDs automatisch
- Bestimmen der aktuellen ID via #{id:mycontrol}
- Funktioniert auch in Repeat-Elementen

Dialoge in XPages

- Basis ist das dijit DialogBox (dijit.Dialog)
- Dojo-Ressource dijit.Dialog zur XPage hinzufügen





- Dojo Theme aktivieren
- Gegebenenfalls ParseOnLoad aktivieren

«XPage»
SimpleDialogDemo

Achtung! Positionierung Dialogbox

- Eine Dialogbox wird automatisch direkt unterhalb des Body Elementes positioniert.
- Damit „Partial Refresh“ und andere Interaktion mit dem Server funktionieren kann, muss sie sich aber innerhalb des Form Elementes befinden.
- Entweder einfach per JavaScript dort positionieren oder dojo Erweiterung von Jermey Hodge nutzen.
  http://xpagesblog.com/xpages-blog/2010/4/10/xpages-compatible-dojo-dialog-reusable-component.html

Picklist mit dojo

- XPage Button hinzufügen
  - CSJS: Dialog erzeugen
- Custom Control mit Ansicht erstellen und einbetten
  - 1. Spalte mit CheckBoxes versehen
  - Button hinzufügen
    - CSJS: Selektierte Werte in Felder schreiben und Dialog schließen
  - Alternativ
    - SSJS: Selektierte Ids auslesen und in den sessionScope schreiben
    - CSJS: Dialog schließen
    - http://xpageswiki.com/web/youatnotes/wiki-xpages.nsf/dx/How_to_create_a_picklist_style_dialog

«XPage»
PicklistDemo

XPages Name Picker

- Tim Tripcony – Erweiterung des Type Ahead „XPages namepicker using standard typeahead“
  http://www.timtripcony.com/blog.nsf/d6plinks/TTRY-7XD5P9
- OpenNTF – XPages Namepicker - Julian Buss (GPL!)
  http://www.openntf.org/Projects/pmt.nsf/0/BB49635F04CB3924862575B1002D5F9C
  - Kommerzielle Version verfügbar
- OpenNTF Extension Library – Name Picker Control
  http://extlib.openntf.org

dojo Widgets in Xpages – Accordion Container

- Navigationsleiste mit AccordionContainer
  - Zusammenklappbare Bereiche
  - Inhalte statisch oder Custom Control
- Bestandteile
  - Package dijit.layout.AccordionContainer
  - Package dijit.layout.ContentPane
  - Verschachtelte DIVs

```
<div dojoType="dijit.layout.AccordionContainer" style="height: 300px;">
   <div dojoType="dijit.layout.ContentPane" title="Panel 1">Content 1</div>
   <div dojoType="dijit.layout.ContentPane" title="Panel 2">Content 2</div>
   <div dojoType="dijit.layout.ContentPane" title="Panel 3">Content 3</div>
</div>
```

dojo Widgets in Xpages – dojox.widget.Rating (1)

- Rating anhand von Sternen
- Sterne sind an ein DIV und nicht an ein Feld gebunden
- Wert aus dem Dijit auslesen und in ein verstecktes Feld schreiben
- Zusätzliches CSS benötigt
  <xp:styleSheet href="/.ibmxpres/dojoroot/dojox/form/resources/Rating.css"></xp:styleSheet>
- Package dojox.form.Rating

dojo Widgets in Xpages – dojox.widget.Rating (2)

```
<div id="rateControl"
  dojoType="dojox.form.Rating"
  numStars="5"
  value="2"
  onChange=
  "dojo.byId('#{id:editControlName}').value
  = dijit.byId('#{id:rateControl}').value">
</div>
```

dojo Widgets in Xpages – dijit.layout.TabContainer (1)

- Standard XPages Tab Container
  - Rendered nur aktiven Tab
  - Andere Tabs stehen für Client Side JavaScript nicht zur Verfügung
  - Jeder Tab-Wechsel bedeutet Serveraufruf
  - Eher langsam
- Dojo's Tab Container
  - Alle Tabs werden gerendered
  - Dojo übernimmt Verstecken/Anzeigen
  - Sehr schnell

dojo Widgets in Xpages – dijit.layout.TabContainer (2)

- Bestandteile
  - Package dijit.layout.TabContainer
  - Package dijit.layout.ContentPane
  - Verschachtelte DIVs oder Panel

```
<xp:panel dojoType="dijit.layout.TabContainer" style="width: 350px; height:
300px">
   <xp:panel dojoType="dijit.layout.ContentPane" title="dojo Tab 1">
      Content Tab 1
   </xp:panel>
   <xp:panel dojoType="dijit.layout.ContentPane" title="dojo Tab 2">
      Content Tab 2
   </xp:panel>
   <xp:panel dojoType="dijit.layout.ContentPane" title="dojo Tab 3">
      Content Tab 3
   </xp:panel>
</xp:panel>
```

«XPage»
WidgetsDemo

dojo DataGrids

- Darstellung von Datenmengen in Tabellen
- dojox.grid.DataGrid (dojo 1.1)
  - Einfache Darstellung
  - Dynamisches Sortieren
  - Vertauschen von Spalten
  - Dynamische Filter
- dojox.grid.EnhancedGrid (dojo 1.4)
  - Checkboxes und dynamische Selektion
  - Kontextmenüs
- Daten können aus vielfältigen Quellen stammen
  - JSON, XML, CSV und weitere

DojoX DataGrid - Ressourcen

```
<xp:this.resources>
        <xp:dojoModule name="dojox.grid.DataGrid"></xp:dojoModule>
        <xp:dojoModule name="dojox.data.XmlStore"></xp:dojoModule>
        <xp:styleSheet
href="/.ibmxspres/dojoroot/dojox/grid/resources/Grid.css">
        </xp:styleSheet>
        <xp:styleSheet
href="/.ibmxspres/dojoroot/dojox/grid/resources/tundraGrid.css">
        </xp:styleSheet>
</xp:this.resources>
```

DataGrid – Einrichtung (1)

- Datenquelle definieren

```
var store = new dojox.data.XmlStore({
  url:
'#{javascript:facesContext.getExternalContext().
getRequestContextPath();}/CustomersInXml.xsp'
});
```

- Layout definieren

```
var layout = [{field: 'customername', name: 'Kundenname',
width: '200px'}, { field: 'customernr', name: 'Kundennr.',
width: '50px'}, {field: 'country', name: 'Land', width:
'200px'}];
```

DataGrid – Einrichtung (2)

- Grid definieren

```
grid = new dojox.grid.DataGrid({
        query: { customername: '*'},
        store: store,
         autoHeight: 10,
        autoWidth:true,
        columnReordering: true,
        noDataMessage: "<span class='dojoxGridNoData'>Keine passenden
        Daten gefunden!</span>",
        structure: layout},
             "#{id:gridContainer}");
        // Grid rendern
       grid.startup();
```

«XPage»
DataGridDemo

dojo Charts

- Darstellung von Daten in Diagrammen
- Reichhaltige Darstellungsform
- Interaktivität
- Daten werden als JSON übergeben
- Intec Blog - Dojox Charting Tutorial
  http://hermes.intec.co.uk/Intec/Blog.nsf/archive?openview&title=Dojox%20Charting%20Tutorial&type=cat&cat=Dojox%20Charting%20Tutorial

DojoX Charts - Ressourcen

```
<xp:this.resources>
        <xp:dojoModule name="dojox.charting.Chart2D"></xp:dojoModule>
        <xp:dojoModule name="dojox.charting.plot2d.Pie"></xp:dojoModule>
        <xp:dojoModule
        name="dojox.charting.action2d.Highlight"></xp:dojoModule>
        <xp:dojoModule
        name="dojox.charting.action2d.MoveSlice"></xp:dojoModule>
        <xp:dojoModule
        name="dojox.charting.action2d.Tooltip"></xp:dojoModule>
        <xp:dojoModule
        name="dojox.charting.themes.PlotKit.blue"></xp:dojoModule>
        <xp:dojoModule name="dojox.charting.widget.Legend"></xp:dojoModule>
        <xp:script src="/dojoXChartingData.jss"
        clientSide="false"></xp:script>
</xp:this.resources>
```

DojoX Chart – Daten als JSON

- Serverside JavaScript Funktion erstellen, um die Daten im JSON Format zu liefern
  {y: [Wert], text: "[Beschreibung]", stroke: "black", tooltip: "[Hinweis]"}
- Im Script-Output-Block die Funktion aufrufen und in eine Variable schreiben
  var chartData =
  [#{javascript:getDojoXChartData();}];

DojoX Chart – Chart erzeugen

```
var dc = dojox.charting;
var chartTwo = new dc.Chart2D("#{id:chartTwo}");
chartTwo.setTheme(dc.themes.PlotKit.blue).addPlot("default", {
            type: "Pie",
            font: "normal normal 11pt Tahoma",
            fontColor: "black",
            labelOffset: -30,
            radius: 80
        }).addSeries("Series A", chartData);
var anim_a = new dc.action2d.MoveSlice(chartTwo, "default");
var anim_b = new dc.action2d.Highlight(chartTwo, "default");
var anim_c = new dc.action2d.Tooltip(chartTwo, "default");
chartTwo.render();
var legendTwo = new dojox.charting.widget.Legend({chart: chartTwo},
        "#{id:legendTwo}");
```

«XPage»
ChartDemo

Ressourcen

Offizielle Webseite
http://www.dojotoolkit.org

API Dokumentation
http://api.dojotoolkit.org

Demos
http://demos.dojotoolkit.org

DojoCampus
http://dojocampus.org/

Dojo Dokumentation
http://docs.dojocampus.org/

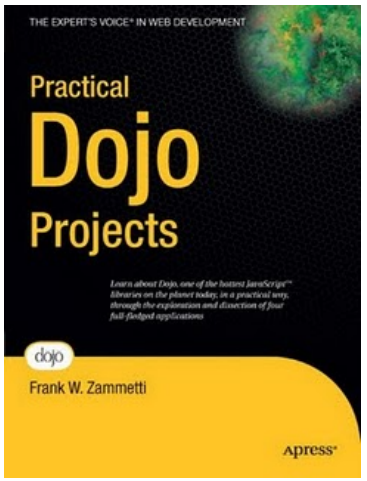
Alte Webseite
http://o.dojotoolkit.org

Ressourcen

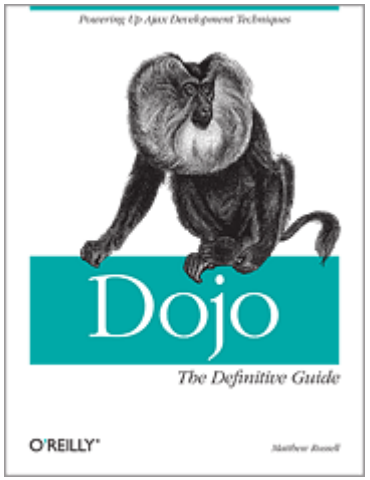
- IBM Widget Gallery
  http://www-01.ibm.com/software/ucd/widgetgallery/widget_home.html
- OpenNTF CodeBin - Dojo - Easy As 123
  http://www.openntf.org/Projects/codebin/codebin.nsf/0/A2A3F3DB69E21F2A8625740E005B8EC9
- Dojomino – Dojo Domino Framework
  http://dojomino.com/
- Sitepen Labs Dojo
  http://o.sitepen.com/labs/dojo.php
- Dojo Toolbox – Adobe AIR Anwendung mit Offline API
  http://o.sitepen.com/labs/toolbox/

Bücher

- Frank W. Zammetti
  "Practical Dojo Projects"
  Apress Verlag



- Matthew A. Russell
  "Dojo – The Definitive Guide"
  O´Reilly Verlag



Fragen?

jetzt stellen – oder später:

bhort@assono.de
http://www.assono.de/blog
040/73 44 28-315

assono
IT-Consulting & Solutions

Folien & Beispieldatenbank unter
http://www.assono.de/blog/d6plinks/EntwicklerCamp-2011-dojo-XPages

dijit - Einführung

- Dijit bezeichnet zwei Dinge
  - Das Widget Framework basierend auf den base und core Elementen
  - Ein einzelnes Widget wird auch als dijit bezeichnet
- Dijits sind voll „theme-able“ - können im Aussehen angepasst werden
  - Gewünschtes „Theme“ als CSS einbinden
  - `<body class="{Theme}">`
  - Beliebtestes „Theme“ ist „Tundra“

Zwei Arten dijits einzubinden

- Dijits können auf zwei Arten eingebunden werden
  - Deklarativ
  - Programmatisch
- Beim deklarativen Ansatz wird der HTML-Code nach Attributen „dojoType“ abgesucht
- Gefundene Elemente werden automatisch um die notwendigen JavaScript-Aufrufe ergänzt
- Vorsicht! HTML Validierungs-Services melden den HTML-Code als nicht valide!

Dijits deklarativ einbinden


- ```
  var djConfig = {
      baseScriptUri : "js/dojo/",
      parseOnLoad : true
  };
  ```
  Konfiguration zum Parsen des HTML-Codes beim Laden

- `dojo.require("dojo.parser");`
  Notwendiges Package zum Parsen

- `dojo.require("dijit.form.Button");`
  Das zum dijit gehörige Package laden

- ```
  <button
  dojoType="dijit.form.Button">ok</button>
  ```
  Dojo Attribute hinzufügen

Dijit programmatisch einbinden

- `dojo.require("dijit.form.Button");`
  Das zum dijit gehörige Package laden

- ```
  var myDijit =
  new dijit.form.Button(
      {label : "Ok"}
  );
  dojo.byId("divButton").appendChild(myDijit.do
  mNode);
  ```

- Der programmatische Ansatz ist der performantere Ansatz.
  - Nur bei großen HTML-Dokumenten wichtig
  - Oder bei hoch dynamischen Webseiten
  ```

Zugriff auf dijits

- Jedes dijit wird vom dijit Manager verwaltet
- Jedes dijit hat eine eindeutige Id
  - Entweder manuell definiert
  - Oder automatisch generiert
- Die Methode `dijit.byId(<someID>);` liefert das dijit Objekt zurück
- Nicht zu verwechseln mit `dojo.byId(<someID>);`
  - Die Methode liefert den DOM-Knoten zurück