

wabion

led by sense

Nutzung des
Dojo Toolkits zur
Optimierung
bestehender
Notes-WebAnwendungen

Mathias BierlWabion GmbH







Agenda



- Vorstellung
- Was ist das Dojo Toolkit?
- Warum sollte ich es nutzen?
- Einbindung des Dojo Toolkits
- Feldvalidierung
- Nutzung von Ajax
- Tips & Tricks

Über Wabion



Gegründet 2004
Firmensitz in Esslingen –
Einsatz weltweit
10 feste Mitarbeiter
15 freie Mitarbeiter

Größter Google Enterprise Partner in Deutschland.

Focus Bereiche:

- Ganzheitliche Groupwarelösungen und deren Einsatz im Unternehmen (IBM und Google)
 - Migrationen zwischen Groupwarelösungen und zu Cloud Lösungen
 - Enterprise Search auf Basis Google Search Appliance
 - Anwendungsentwicklung auf Basis Google Apps Engine und Lotus Domino
- Beratungsunternehmen mit dem Focus jeweils Technologieführer auf seinem Gebiet zu sein.

Wer bin ich?



Mathias Bierl, Staatl. Gepr. Wirtschaftsinformatiker (mathias.bierl@wabion.com)
Notesentwicklung / Administration seit 1997
(R 4 – ND 8.5.x)
Anti-Spam/Anti-Virus Lösungen
Google Lösungen (Search, Apps,...)
Server-Virtualisierung

Wer bin ich?



- Projekte mit Lotus / Domino:
 - Lotus Notes R4 ND 8.5.x
 - Sametime, Domino.Doc, Lotus Workflow, LEI, DECS, DGW
 - Formelsprache, Skript, Klassen, Java, JS, HTML, AJAX
 - Migration R4-R5-ND6-ND7-ND8, Microsoft Exchange,...
 - C-/C++-API, WIN 3.11-WIN 7, OS/2, AIX, AS/400, Linux, ...
 - Datenanbindungen an SAP, RDBS
 - XML/XSL, Servlets, Web Services
 - Mobile Datenanbindung (Blackberry, OBMG,...)
- Projekte mit Google:
 - Google Search Appliance
 - Google Message Security/Discovery
 - Google Apps inkl. Migrationen
 - Google Apps Engine
 - Google Maps/Earth

Was ist das Dojo Toolkit?



- Das Dojo Toolkit ist ein JavaScript Framework zur einfachen Entwicklung von Rich Internet Applikationen
- Unterstützung von Ajax Technik
- Dynamische Gestaltung der Oberfläche
- Visuelle Effekte
- Themes
- Widgets
- Datenverwaltung
- Grafikunterstützung, Charts

Warum sollte ich es nutzen?



- Bietet einfache Möglichkeiten dem Nutzer erweiterte Möglichkeiten zur Verfügung zu stellen
 - z.B. DatePicker
- Browsereigenheiten werden durch das Toolkit verarbeitet
 - Keine eigenen Browserweichen mehr nötig
- Unterstützung auch mobiler Endgeräte
- Wird bei aktuellen Domino Versionen mitgeliefert
 - Neuere/andere Versionen können installiert werden

Warum sollte ich es nutzen ?w

- Warum das Dojo Toolkit direkt nutzen anstatt gleich ganz auf XPages zu wechseln ?
 - XPages sind anders strukturiert, d.h. es muss eventuell viel Entwicklungszeit in die Umstellung von existierenden Anwendungen gesteckt werden
 - Bestehende Anwendungen sollen einfach nur ein wenig komfortabler gemacht werden

Dokumentation



- Homepage
 - http://dojotoolkit.org/
- API Dokumentation
 - http://dojotoolkit.org/api/
- Reference Guides
 - http://dojotoolkit.org/reference-guide/
- Tutorials
 - http://dojotoolkit.org/documentation/
- Vorsicht: Auf korrekte Versionsauswahl achten

Aufbau des Dojo Toolkits



- Das Dojo Toolkit liegt auf dem Domino Server im Datenverzeichnis unter domino/js
- Jede Dojo Version besteht aus verschiedenen Unterverzeichnissen
- Zur Einbindung eines Dojo Elements ist die Verzeichnishierarchie wichtig:
 - dojo/fx/Toggler.js wird zu dojo.fx.Toggler

Aufbau des Dojo Toolkits



- Das im Domino mitgelieferte Dojo besteht aus folgenden Teilbereichen:
 - dojo: Core Elemente von Dojo
 Browser Erkennung, JSON, Ajax, Basic CSS Styles,
 Formatierung, Lokalisierung, Cookie Handling, Event
 Handling
 - dijit: Dijit Widgets Cross-Browser Unterstützung, CSS Themes
 - dojox: Erweiterte Elemente
 Experimentelle Erweiterungen, Charts, Data Access Layer,
 Offline Features, Verschlüsselung
 - ibm: Domino spezifische Dojo Erweiterungen

Demo



- Einbinden des DojoToolkits
 - Anhand von
 - Text/Nummer
 - Datum
 - Checkboxen
 - Radio Buttons
 - RichText
 - Listboxen/Comboboxen

Demo



Titel:		Titel: * Anzahl: *	2
A 11		Anzani. " Betrag:	0
Anzahl:	0.00	Datum:	28.03.2012
Betrag:	0.00	Kategorien:	Kategorie 1
	0.00		Kategorie 2
Datum:	28.03.2012	Status:	☐ Kategorie 3
Kategorien:	☐ Kategorie 1	Status.	Zur Genehmigung
reacegorica.	-		Genehmigt
	☐ Kategorie 2	Bereiche:	Bereich 1 ▼
	☐ Kategorie 3	Bemerkungen:	♥ ♥ ★ □ □ B I U S 計 日 極 極 三 三 三 ■
Status:	© Entwurf		
	C Zur Genehmigung		
	C Genehmigt		
D 11			
Bereiche:	Bereich 1 ▼		
Bemerkungen:			
	0		
	Speichern		One-inhorn
			Speichern

Einbinden des Dojo Toolkits w

- Einbinden der Dojo Library (HTML Header)
 - <script type="text/javascript" src="/domjs/dojo-1.4.3/dojo/dojo.js" djConfig="parseOnLoad: true"/>
 - djConfig: definiert Konfigurationsparameter
 - parseOnLoad: parst die HTML Eigenschaften auf Dojo Einstellungen
 - isDebug: aktiviert den Debug Modus
- Einbinden des Basis Stylesheets (HTML Header)
 - type="text/css" href="/domjs/dojo-1.4.3/dojo/resources/dojo.css">
 - Basis Styles

Einbinden des Dojo Toolkits w

- Einbinden der Dijit Widget Basis (HTML Header)
 - Einbinden eines Themes für die Darstellung
 - type="text/css" href="/domjs/dojo-1.4.3/dijit/themes/nihilo/nihilo.css">
 - Notwendig f
 ür die Darstellung der Widgets
 - Auch eigene Themes möglich
 - Festlegen des verwendeten Themes (HTML Body)
 - class="nihilo"

Festlegen von Feldtypen



- Einbinden der JavaScript Elemente im JS Header
 - dojo.require("dijit.form.DateTextBox");
 - Hiermit wird dann die JavaScript Datei DateTextBox.js aus dem Unterverzeichnis dijit\form eingebunden
- Festlegen des Feldtyps in den HTML Eigenschaften
 - dojoType="dijit.form.DateTextBox"
 - Fügt dem Feld die entsprechenden Eigenschaften hinzu. In diesem Fall wird beim Klick auf das Feld ein DatePicker geöffnet

Festlegen von Feldtypen



• Typische Domino Feldtypen

Domino Feld	Dijit Klasse	
Text	dijit.form.TextBox dijit.form.ValidationTextBox	
Nummern	dijit.form.NumberTextBox dijit.form.NumberSpinner dijit.form.CurrencyTextBox	
Datum/Zeit	dijit.form.DateTextBox dijit.form.TimeTextBox	
Checkboxen	dijit.form.CheckBox	
Radiobuttons	dijit.form.RadioButton	
RichText	dijit.Editor	
Listboxen/Comboboxen	dijit.form.ComboBox dijit.form.FilteringSelect	



- Die Festlegung von Feldtypen fügt dem Feld dann auch gleich eine Basis Validierung hinzu. z.B. auf ein gültiges Datum
 - Dies verhindert allerdings nicht ein Speichern des Dokumentes
 - Daher muss das jeweilige Feld noch auf validen Inhalt geprüft werden
 - dijit.byId(ElementID)
 - Methode isValid()



- Die Festlegung von Feldtypen fügt dem Feld dann auch gleich eine Basis Validierung hinzu. z.B. auf ein gültiges Datum
 - Dies verhindert allerdings nicht ein Speichern des Dokumentes
 - Daher muss das jeweilige Feld noch auf validen Inhalt geprüft werden
 - dijit.byId(ElementID)
 - Methode isValid()



- Eigenschaft "regExp" um den Inhalt anhand eines regulären Ausdrucks zu validieren
 - z.B. ,,\\d{5}" validiert auf Zahl mit 5 Stellen z.B. Postleitzahl
- Eigenschaft "required" um ein Feld zum Pflichtfeld zu machen
- Weitere Validierung ist im dojox.validate verfügbar
- Natürlich besteht weiterhin die bisherige Validierungsmöglichkeit per JavaScript



- Anpassen der Fehlermeldungen
 - Eigenschaft "invalidMessage"
- Hinzufügen eines Hinweises
 - Eigenschaft "promptMessage"

Hinweise



- Datumsfelder werden im Edit Modus nicht richtig dargestellt, da Dojo den Datumswert im Format XXXX-XXX erwartet
 - Datumsfeld im Edit Modus entsprechend formatieren—
- Bei einem FilteringSelect müssen auf jeden Fall Alias Werte generiert werden
 - -> Feldoption setzen
- Zur leichten Umstellung auf neuere Dojo Versionen sollte der Pfad konfigurierbar sein



- Dynamische Abfrage von Daten
 - Für Änderungen der Abfragebasis muss nicht jedes Mal die ganze Maske neu geladen werden
- Senden von Daten im Hintergrund
- Einsatzzwecke
 - Füllung von abhängigen Dialogfeldern
 - Dynamische Validierung
 - Aktualisierung von Teildaten
 - Senden von Daten
 - Senden von Abfragen

Demo



- Ajax mit dem DojoToolkit
 - Anhand von
 - Namensabfrage gegen das Domino Verzeichnis



- Abfrage von Daten per Ajax geschieht über
 - dojo.xhrGet
- Grundaufbau

```
dojo.xhrGet(
        url: "URL",
        handleAs: "Ausgabeformat",
        timeout: 1000,
        load: function(Parameter) {
        error: function(Parameter) {
```



- url
 - URL, die die Daten im gewünschten Format zur Verfügung stellt
 - Nur URLs der gleichen Domain sind möglich
- handleAs
 - Ausgabeformat
 - text
 - json
 - javascript
 - xml
- timeout
 - Zeit in ms nachdem der Request als fehlgeschlagen abgebrochen wird



- Weitere Parameter
 - sync: synchroner (true) oder asynchroner Aufruf (false)
 - preventCache: Hängt bei true einen eindeutigen Parameter an den Aufruf an um ein Caching zu verhindern
 - content: URL Parameter als Name/Wert JavaScript Objekt
 - headers: Header Parameter als Name/Wert JavaScript Objekt
 - user/password: Benutzerdaten für HTTP Basic Authentifizierung



- load
 - Funktion, die aufgerufen wird nachdem die Daten erfolgreich geladen wurden
- error
 - Funktion, die aufgerufen wird wenn der Abruf fehlschlägt
- handle
 - Funktion, die immer aufgerufen wird. Der Status muss in dieser dann separat geprüft werden



- Die Anzahl der Parameter ist abhängig vom handle As Parameter
 - Minimal werden zwei Parameter übergeben:
 - Die Daten im entsprechenden Format
 - I/O Argumente, z.b Status Codes,...
 - Der Status Code kann über xhr.status aus den I/O Argumenten abgefragt werden



- Senden von Daten per Ajax geschieht über
 - dojo.xhrPost bzw dojo.xhrPut
 - Es können sowohl reine Textdaten wie auch HTML Formulare gesendet werden
- Mit dojo.xhr kann auch dynamisch anhand des ersten Parameters die Methode zur Kommunikation gewählt werden

Debugging



- Verwenden von JavaScript Debuggern
 - z.B. Firebug, Tools in Chrome und neueren IEs
- Nutzung des console-Objektes (Firebug, Chrome,...)
 - warn
 - log
 - info
 - error
 - debug







Kontakt



Wabion GmbH

Katharinenstrasse 57

D-73728 Esslingen

Phone: +49 (0)711 – 25 25 52 24

Fax: +49 (0)711 – 25 25 52 14

Web: http://www.wabion.com

Mathias Bierl

eMail: Mathias.Bierl@wabion.com

Mobil: +49 (0) 172 877 27 65