

Using the XPages Mobile Controls

Plus: The pros, cons and alternatives

Richard Sharpe:
Technical Director - Teamstudio



Agenda

- Housekeeping – Setting up testing
- What is the Mobile Controls Project?
- Mobilizing a simple app using the XPage Mobile Controls
- Pros and Cons
- Alternative framework – JQuery Mobile
- Housekeeping – Adding icons and loading screens
- Implementing an 'iPhone spinner' (if time allows)
- Summary

What to Mobilize

- Parts of applications
- Good UI (Angry Birds)
- Ease of Use
- Iterate Frequently – don't overload

For Testing

- For testing in a web browser, Safari or Chrome must be used as IE and Firefox do not support Webkit technologies
 - **Note:** for Windows, Chrome is more stable
- Install Chrome/Safari
- Open Designer and go to:
 - File – Preferences – General – Web Browser
- Click 'New' and in the dialog box for 'Add External Web Browser,' enter a name and the install location (diagram on next slide shows a Safari Install)
- Now in Designer, when you select 'Preview in web browser,' this new browser should appear in the list



Mobile Controls Project

- Created by Niklas Heidloff in Sept 2010
- Design to assist with prototypes of mobile Notes applications
- Not very flexible in usage with existing applications
- Second generation is part of the Extension Library project
- IBM has integrated these controls into the Controls Palette in Designer



Notes on prior slide

Single Page Application (xe:singlePageApp)

This is the container control for the XPages mobile application, controlling its layout and configuration. The property **selectedPageName** must be set here to a Mobile Page control that is to be displayed when the XPage is loaded.

Mobile Page (xe:appPage)

This is the main 'screen' used in a mobile application. Only one of these mobile pages is displayed at a time. The **pageName** property is the mobile page name and is the property used for navigation between the mobile pages as well as being used by the singlePageApp control to decide what page to show initially. The **resetContent** property is used to indicate if the page contents should be recreated each time the page is displayed. Another property affecting performance is **preload**, which can be used to force the Mobile Page to be loaded when the whole xpage is loaded.

Page Heading (xe:djxmHeading)

Used as the header for the Mobile Page. It contains properties for the basic navigation - the **back** property being the label for the Back button and the **moveTo** property to contain the pageName of the destination of the Mobile Page.

The Heading control can also act as a container for other controls like buttons and the mobile applications Tab Bar.



Notes on prior slide (continued)

Rounded List (xe:djxmRoundRectList)

This control displays a rectangle with rounded corners useful for lists that link to documents.

Static Line Item (xe:djxmLineItem)

The Static Line Item control is a dojo control mainly used to link to other mobile pages.

Mobile Switch (xe:djxmSwitch)

The Mobile Switch control is probably best described as a fancy looking checkbox.

Tab Bar (xe:tabBar)

The Tab Bar on its own displays like a banner across the Mobile page, and is mainly used as a container control for the Tab Bar Button. When the **barType** property is set to segmentedControl, the tab bar buttons display together as one though separated into their individual buttons.

The Tab Bar can also be placed inside the Heading.

Tab Bar Button (xe:tabBarButton)

The Tab Bar Button is very much the same as the conventional XPages button (xp:button) but with different styling for XPages Mobile applications.



Employees on iPhone and Android



Not yet implemented

Pros & Cons

- Fast – ideal for proof of concept
- Official – These are now part of Designer and will improve over time
- Supported – As these are on OpenNTF and part of Designer there are resources to help
- Still a bit buggy
- Requires 8.5.3 on both Designer and Server

Why Alternatives

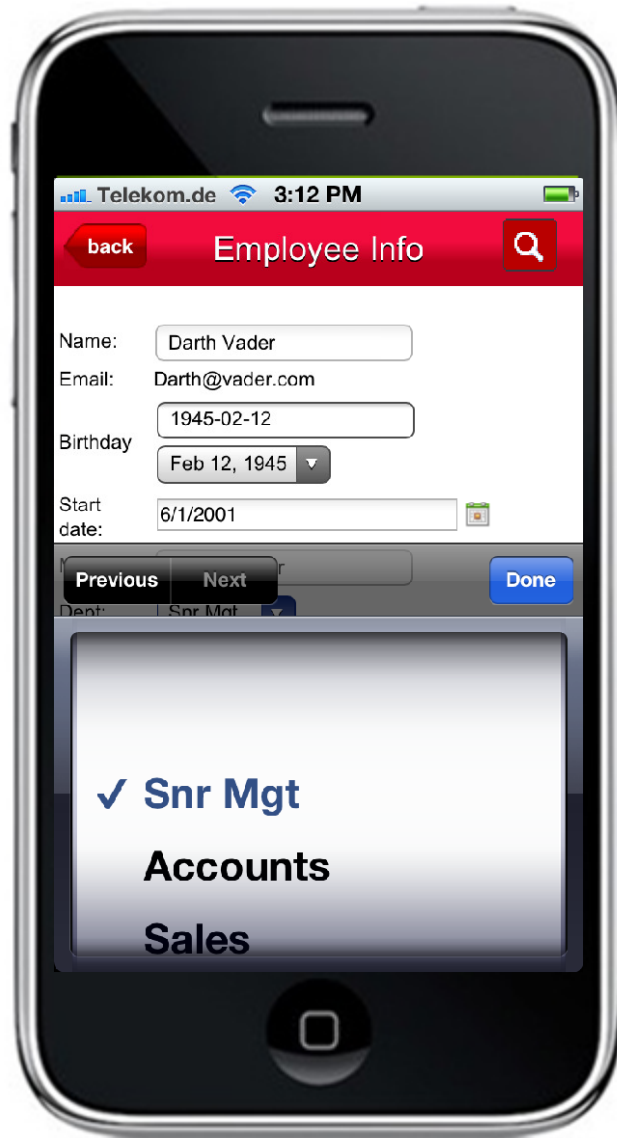
- Not everyone is on 8.5.3 (client and server)
- A few small bugs
- Development methodology
- Other frameworks are more established



Using jQueryMobile

- Based on JQuery but separate library
- Client side rendering and functional JS library specifically designed for mobile UI
- Architected on a “Single Page Application” model or separate pages
- “Hijacks” the HTML elements and applies UI and functionality based on the HTML 5 ‘data-role’ property
- Comes with a set of pre-defined ‘themes’ and icons

Implementing a Spinner



Exercise – Implementing a Spinner

- Switch to the Source tab and add the code to display the current Birthday in the Spinner
- Create the Spinner using the `<input type=date>` tag
 - This must be a string in the form of 'yyyy-mm-dd'

```
<xp:tr>
  <xp:td>
    <xp:label value="Birthday" id="label1"></xp:label>
  </xp:td>
  <xp:td>
    <xp:inputText id="inputText1" styleClass="realDateField" value="#
{document1.Birthday}">
      <xp:this.converter>
        <xp:convertDateTime type="date" pattern="yyyy-MM-dd">
          </xp:convertDateTime>
        </xp:this.converter>
      </xp:inputText>
      <input id="DOB" type="date" value="#{javascript:
        var birthday:NotesDateTime = document1.getItemValueDate('Birthday');
        return @Year(birthday) + '-' + @Right('0' + @Month(birthday),2) + '-' + @Right('0' +
          @Day(birthday),2);}" />
    </xp:td>
  </xp:tr>
```

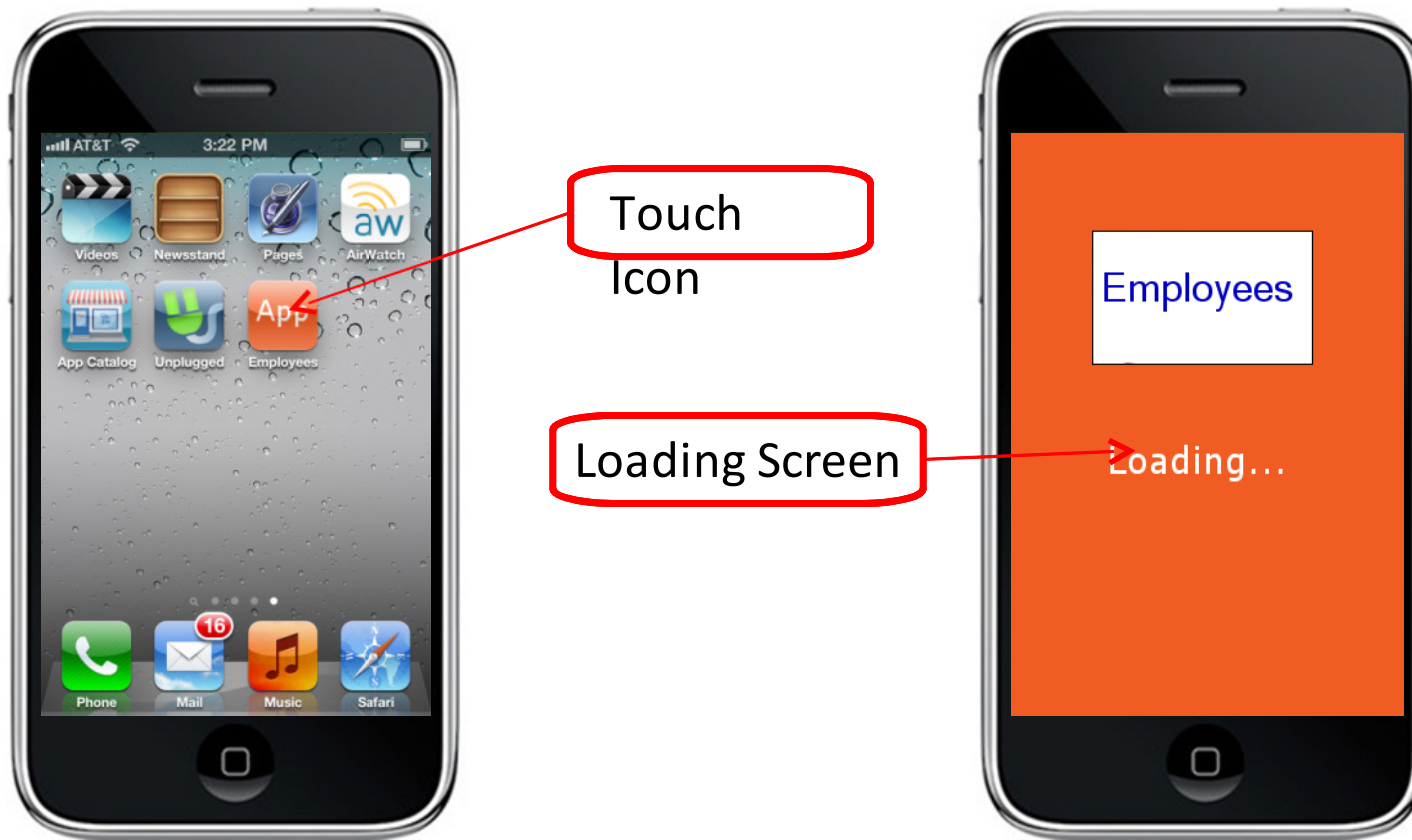


Exercise – Implementing a Spinner

- Bind these values to back end document
- Switch back to the Design Pane and select the ‘Update Details’ button
- In the Events tab, select the onClick event and the Client tab
- Edit the JavaScript (client side) to set the value of the spinner to the Edit Box control with id=“DOB”

```
var theBday = document.getElementById("DOB");  
dojo.byId("#{id:inputText1}").value = theBday.value;  
//$('.realDateField').val(theBday.value);  
return true;
```

Preparing for Distribution



Preparing for distribution

- This is performed by the use of meta-tags in the `<xp:this.resources>` area
- Works on Android (even though it says Apple!!)

```
<xp:metaData name="viewport"
content="width=device-width, initial-scale=1.0, user-scalable=no, minimum-
scale=1.0, maximum-scale=1.0">
</xp:metaData>
<xp:metaData name="apple-mobile-web-app-status-bar-style"
content="black-translucent">
</xp:metaData>
<xp:metaData name="apple-mobile-web-app-capable" content="yes">
</xp:metaData>
<xp:linkResource rel="apple-touch-startup-image" href="loading.png">
</xp:linkResource>
<xp:linkResource rel="apple-touch-icon" href="AppIcon.png">
</xp:linkResource>
```



Preparing for distribution

- Image dimensions:
 - Icon = 72px x 72px – Do not add curved edges or the sheen, iOS does that for you
 - Loading = 320 x 460
- To activate open the web page in the device browser and from options, select “Add To Home Page” – an option to name the app is presented (this can be overridden by another method)



Teamstudio Mobile XPages Workshop

One Day Event

- Xpages Mobile Controls and JQuery Mobile
- 'Better' Headers and Footers
- Mobile Widgets
- Form/View Decomposition
- Designing for Tablets and Phones simultaneously
- Offline support

Richard_Sharpe@Teamstudio.com



Resources

- unplugged.teamstudio.com
- jquerymobile.com
- openNTF.org
- demos.dojotoolkit.org/demos/mobileGallery/

Richard_Sharpe@Teamstudio.com

