Mobile XPages Accelerator

EntwicklerCamp 2012

Table of Contents

Lab objectives

The purpose of this lab is to build your first mobile application using the XPages Extensions Library.

You will create a web application first, then extend this to a hybrid app leveraging PhoneGap features. You will also learn how to package and deploy the resulting app to your device.

Time estimate

You should be able to complete the lab in around 45 minutes.

Prerequisites

Appropriate Skill Level:

- Basic XPages skills

Computer Specification:

- Extensions Library installed on both server and client

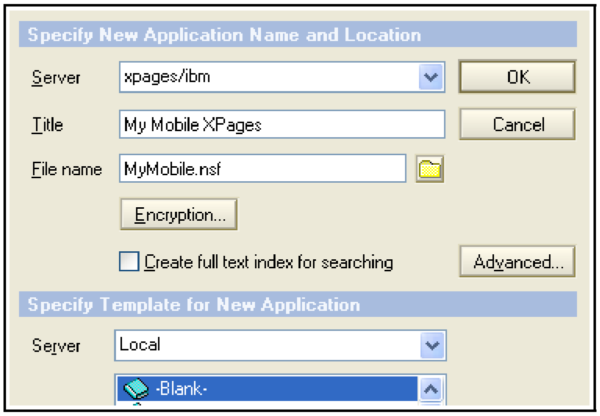
Getting started

This lab includes the following:

- Building your first mobile application with XPages
- Create your first mobile apps quickly by adopting and re-using elements from the Extensions Library
- Integrating XPages technology with PhoneGap features.
- Deploying the app.

1. Copy the **XPagesExt.nsf** from the Extensions Library folder into your Notes data directory.

2. Open **Domino Designer** and create a new sample database on your Domino Server.



Specify New Application Name and Location

Server: xpages/ibm

Title: My Mobile XPages

File name: MyMobile.nsf

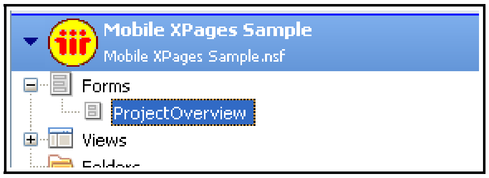
□ Create full text index for searching

Specify Template for New Application

Server: Local

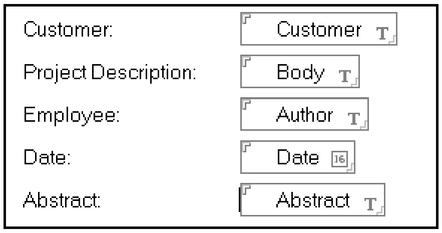
-Blank-

3. Navigate to the **XPagesExt.nsf** and copy the Discussion form into your new database. Rename it to **ProjectOverview** there.



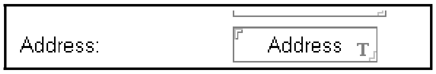
Mobile XPages Sample
Mobile XPages Sample.nsf

4. Open the form and rename the **Title** field to **Customer**. Add some descriptive labels to the fields.



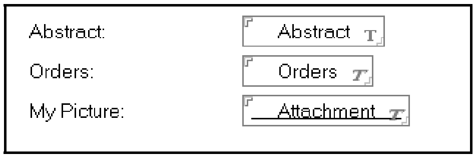
Customer:	Customer
Project Description:	Body
Employee:	Author
Date:	Date
Abstract:	Abstract

5. Add a new text field **Address** to the form.



Address: Address

6. Add two additional **RichText** fields: **Orders** will contain some project documentation in PDF format, **Attachment** is intended to be populated with the employee's picture.

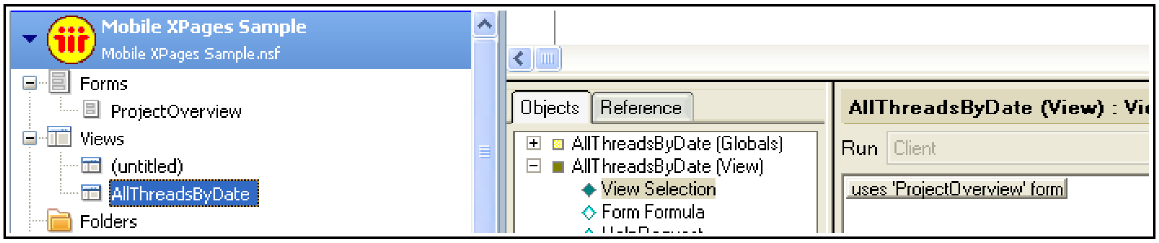


Abstract: Abstract

Orders: Orders

My Picture: Attachment

7. Copy the **AllThreadsByDate** view from the **XPagesExt.nsf** into your new database. Change the view selection to display all documents having the **ProjectOverview** form assigned.

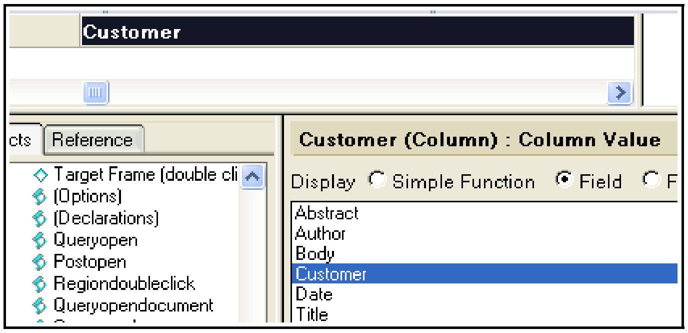


AllThreadsByDate (View) : Vi

Run Client

uses 'ProjectOverview' form

8. Change the second column to display the **Customer** information instead of **Title**.

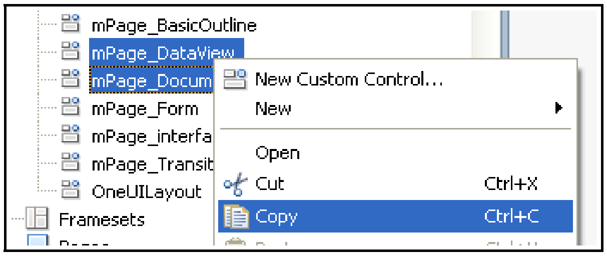


Customer (Column) : Column Value

Display ○ Simple Function ● Field ○ F

Abstract
Author
Body
Customer
Date
Title

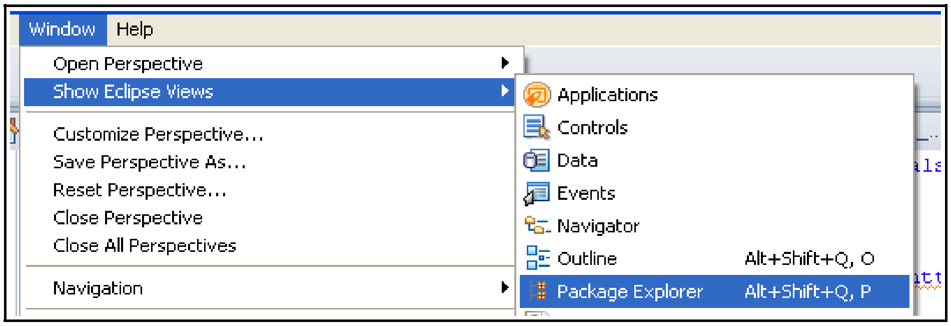
9. Copy the following Custom Controls from the **XPagesExt.nsf** into your database:
   mPage_DataView
   mPage_Document



10. Copy the **mobileDemo** XPage from **XPagesExt.nsf** into your database.

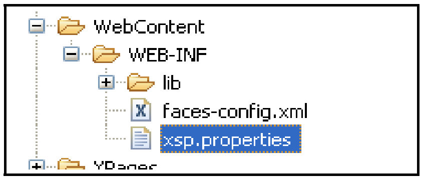
Note: The copied elements still show several errors, so we need to update the XPages properties for this application.

11. From the **Window** menu, add the **Package Explorer** view to your Designer perspective.



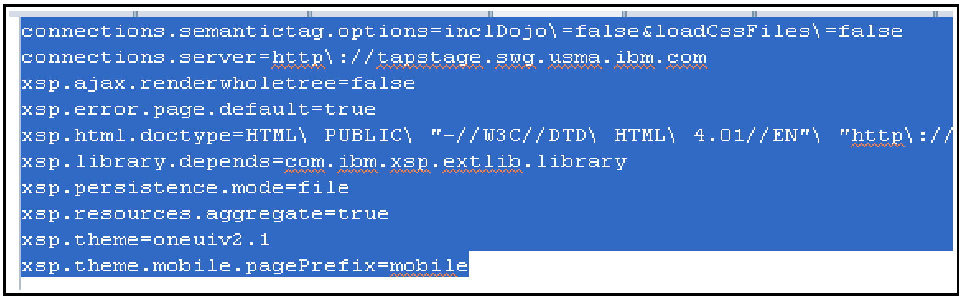
Note: This view shows you the "real" structure of your NSF and provides direct access to the native elements behind your Designer building blocks.

12. Within this new view, navigate to the **XPagesExt.nsf** project. Open the **WebContent/WEB-INF** folder and find the **xsp.properties** file.



13. Open this file and navigate to the **Source** tab.

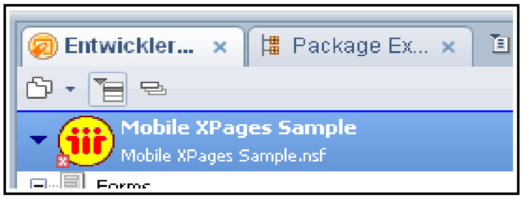
14. Copy all the parameters found here to the clipboard.



```
connections.semantictag.options=inclDojo\=false&loadCssFiles\=false
connections.server=http\://tapstage.swg.usma.ibm.com
xsp.ajax.renderwholetree=false
xsp.error.page.default=true
xsp.html.doctype=HTML\ PUBLIC\ "-//W3C//DTD\ HTML\ 4.01//EN"\ "http\://
xsp.library.depends=com.ibm.xsp.extlib.library
xsp.persistence.mode=file
xsp.resources.aggregate=true
xsp.theme=oneuiv2.1
xsp.theme.mobile.pagePrefix=mobile
```

15. Navigate back to your project and replace the **xsp.properties**' content there with the copied parameters.

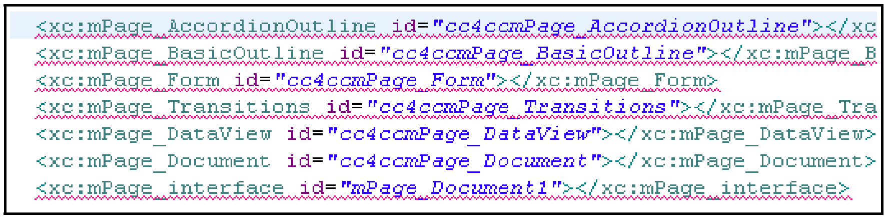
16. Save the **xsp.properties** file and navigate back to your Working Set.



17. Notice that the **mobileDemo.xsp** still shows some errors, so open this XPage now.

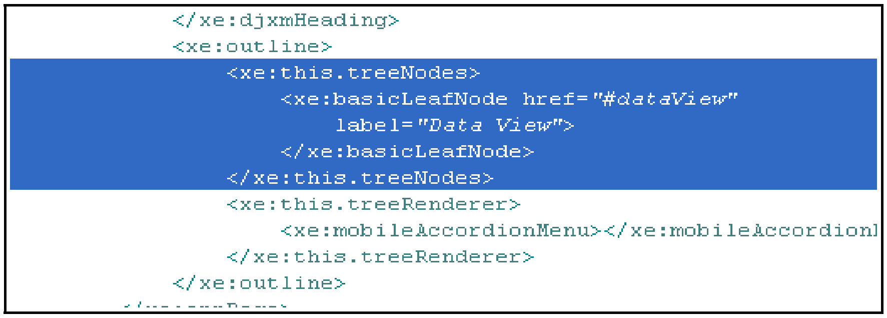
18. Remove the lines with a red underline – we do not need the required Custom Controls for our example.

Keep only the **mPage_DataView** and **mPage_Document** references.



```
<xc:mPage_AccordionOutline id="cc4ccmPage_AccordionOutline"></xc
<xc:mPage_BasicOutline id="cc4ccmPage_BasicOutline"></xc:mPage_B
<xc:mPage_Form id="cc4ccmPage_Form"></xc:mPage_Form>
<xc:mPage_Transitions id="cc4ccmPage_Transitions"></xc:mPage_Tra
<xc:mPage_DataView id="cc4ccmPage_DataView"></xc:mPage_DataView>
<xc:mPage_Document id="cc4ccmPage_Document"></xc:mPage_Document>
<xc:mPage_interface id="mPage_Document1"></xc:mPage_interface>
```

19. Remove the corresponding **basicLeafNode** references and keep only the **#dataView** enty here.



```
            </xe:djxmHeading>
            <xe:outline>
                <xe:this.treeNodes>
                    <xe:basicLeafNode href="#dataView"
                        label="Data View">
                    </xe:basicLeafNode>
                </xe:this.treeNodes>
                <xe:this.treeRenderer>
                    <xe:mobileAccordionMenu></xe:mobileAccordionl
                </xe:this.treeRenderer>
            </xe:outline>
```

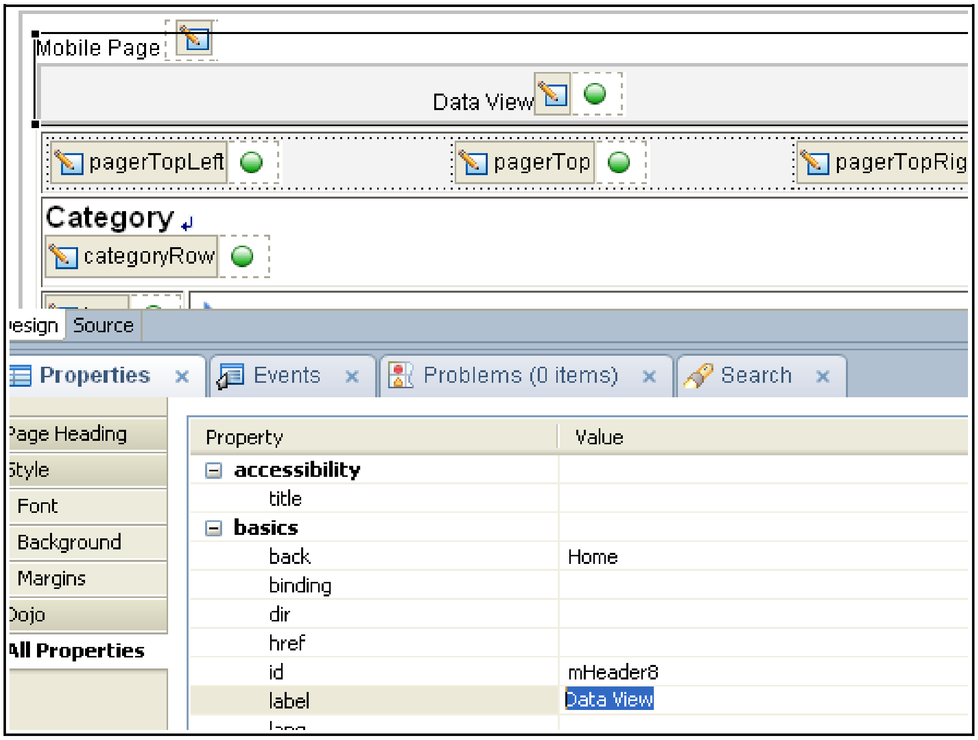
20. Change the **basicLeafNode's** label from **Data View** to **Projektberichte**.

21. Save and close the XPage.

22. Open the **mPage_DataView** Custom Control.

23. Click on the **Page Heading** element and navigate to **Properties > All Properties**.

24. Change the **label** attribute from **Data View** to **Projektberichte**.

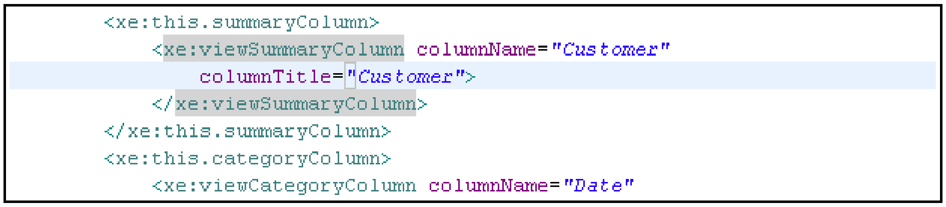


Property	Value
⊟ **accessibility**	
title	
⊟ **basics**	
back	Home
binding	
dir	
href	
id	mHeader8
label	Data View

25. Change the **Home** attribute to **Zurück**.

26. Click on the **Load More** link at the end of the page and change the text attribute to **Mehr**...

27. Change the **Title** column reference to point to the new **Customer** data.



```
        <xe:this.summaryColumn>
            <xe:viewSummaryColumn columnName="Customer"
                columnTitle="Customer">
            </xe:viewSummaryColumn>
        </xe:this.summaryColumn>
        <xe:this.categoryColumn>
            <xe:viewCategoryColumn columnName="Date"
```

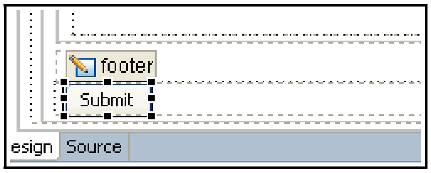
28.Save and close the Custom Control.

29. Open the **mPage_Document** Custom Control.

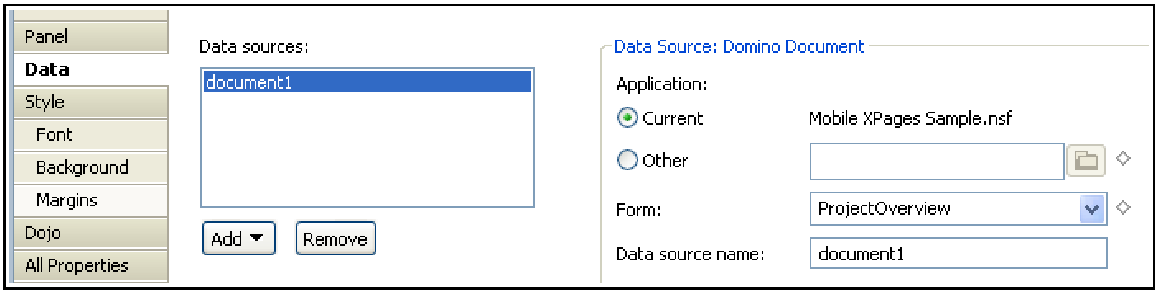
30. Change the Page Header label from **Document** to **Details** and the **back** attribute to **Zurück**.

31. Click on the **Form Table** and change the **formDescription** from **Edit the document's properties** below to **Anpassungen werden per Aktualisieren zurückgeschrieben**.

32. Change the **Submit** button's label to **Aktualisieren**.



33. Change the **Data source** for this document to point to the correct form.



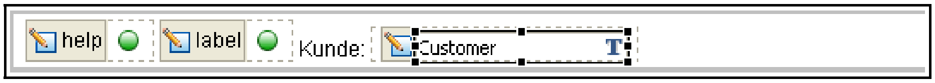
Data sources: document1

Application:
● Current Mobile XPages Sample.nsf
○ Other

Form: ProjectOverview

Data source name: document1

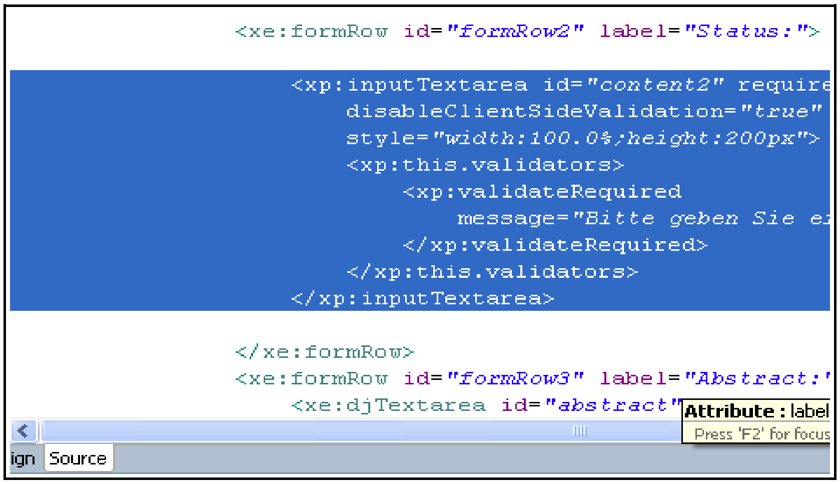
34. Change the first row of the **Form Layout Row** to display the **Customer** information.



35. Change the second **Form Layout Row** to display the project's status information. Remove the **Author** Edit Box.

36. Add a new **Multiline Edit Box** control to the **Form Layout Row**. Here is the code for that, including validation:

```
<xp:inputTextarea id="content2" required="true"
    disableClientSideValidation="true" value="#{document1.Body}"
                        style="width:100.0%;height:200px">

    <xp:this.validators>
          <xp:validateRequired
                message="Bitte geben Sie eine Projektbeschreibung an.">
          </xp:validateRequired>
    </xp:this.validators>
</xp:inputTextarea>
```



```
<xe:formRow id="formRow2" label="Status:">

    <xp:inputTextarea id="content2" require
        disableClientSideValidation="true"
        style="width:100.0%;height:200px">
        <xp:this.validators>
            <xp:validateRequired
                message="Bitte geben Sie ei
            </xp:validateRequired>
        </xp:this.validators>
    </xp:inputTextarea>

</xe:formRow>
<xe:formRow id="formRow3" label="Abstract:'
    <xe:djTextarea id="abstract"
```

37. We now want to provide access to some documentation that has been attached to the project before. So, after formRow2, insert an additional Form Row with the following code:

```
<xe:formRow id="formRow4" label="Unterlagen:">
    <xp:link escape="true" id="link2">

    <xp:this.value><![CDATA[#{javascript:var imageName:String = "";
    if(!document1.isNewNote()){
           var al:java.util.List = document1.getAttachmentList("orders");
           if(!al.isEmpty()){
                  var eo:NotesEmbeddedObject = al.get(0);
                  imageName = eo.getHref();
           }
    }
    return(imageName);}]]></xp:this.value>
    <xp:this.rendered><![CDATA[#{javascript:var imageName:String = "";
    if(!document1.isNewNote()){
           var al:java.util.List = document1.getAttachmentList("orders");
           if(!al.isEmpty()){
                  var eo:NotesEmbeddedObject = al.get(0);
                  imageName = eo.getHref();
           }
    }
    imageName !="";}]]></xp:this.rendered>
    <xp:image id="image1" url="/adobe.gif" style="height:50px;width:50px">
    </xp:image>
    </xp:link>
    <xp:label id="label5" value="nicht verfügbar">
    <xp:this.rendered><![CDATA[#{javascript:var imageName:String = "";
    if(!document1.isNewNote()){
           var al:java.util.List = document1.getAttachmentList("orders");
           if(!al.isEmpty()){
                  var eo:NotesEmbeddedObject = al.get(0);
                  imageName = eo.getHref();
           }
    }
    imageName=="";}]]></xp:this.rendered>
    </xp:label>

 </xe:formRow>
```

38. Navigate to the last **Form Row**. Change the label **Abstract**: to **Kontakt**:. Remove the **Dojo Text Area** from this row.

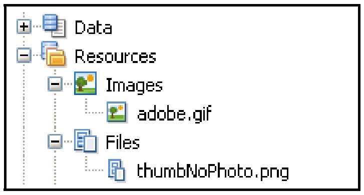
39. Add the following code between the existing formRow tags:

```
   <xe:formRow id="formRow3" label="Kontakt:">
       <xp:br />
       <xp:table style="width:100%">
       <xp:tr>
       <xp:td align="left" valign="top">
       <xp:text escape="false" id="computedField1" value="#{document1.Author}">
       </xp:text>
       </xp:td>
       <xp:td>
       <xp:image id="image4" style="height:50px;width:40px">
             <xp:this.url>
             <![CDATA[#{javascript:var imageName:String = "thumbNoPhoto.png";
             if(!document1.isNewNote()){
             var al:java.util.List = document1.getAttachmentList("attachment");
             if(!al.isEmpty()){
                    var eo:NotesEmbeddedObject = al.get(0);
                    imageName = eo.getHref();
             }
             }
             return(imageName);}]]></xp:this.url>
       </xp:image>
       </xp:td>
       </xp:tr>
       </xp:table>
</xe:formRow>
```

40. Save and close the Custom Control.

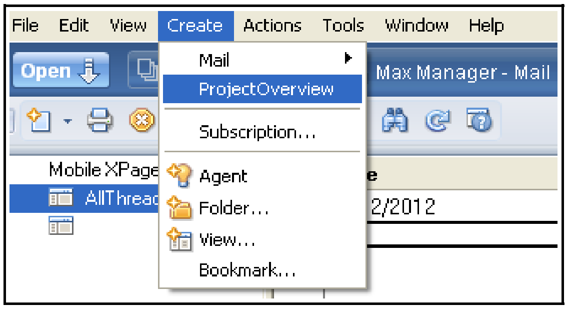
41. Finally navigate to the files you got from your moderator. There should be two image files: **adobe.gif** and **thumbNoPhoto.png**.

Import these files as shown below:

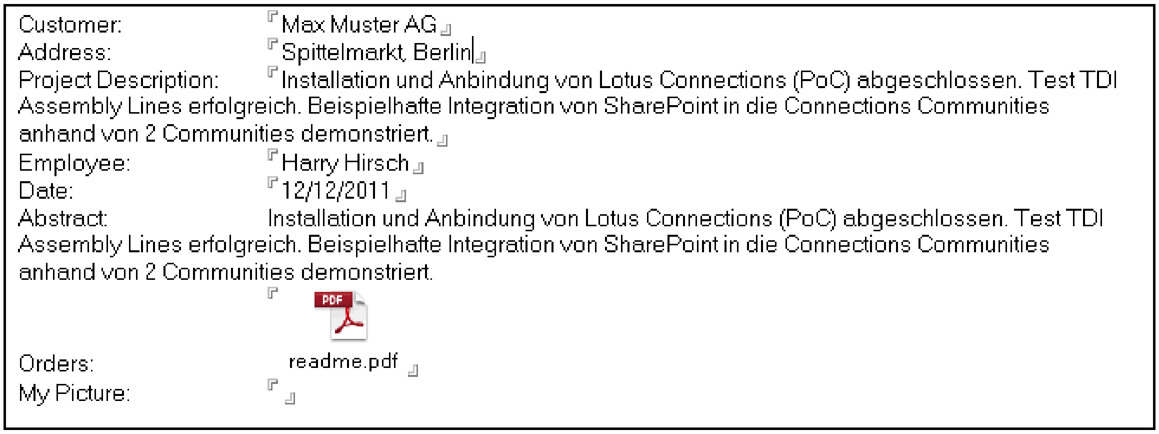


42. Navigate to the **AllThreadsByDate** view and open it in Lotus Notes.

43. Create a new document by selecting **Create > ProjectOverview** from the menu.

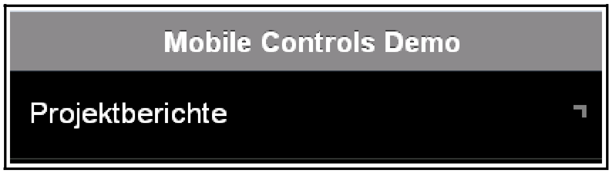


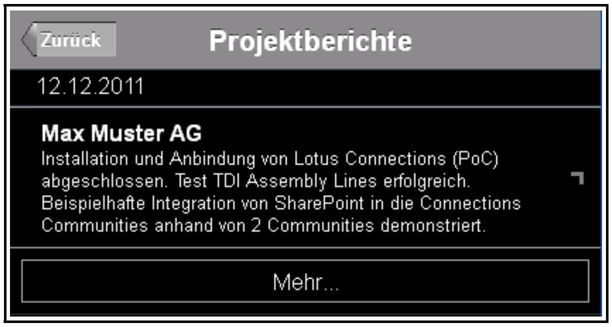
44. Add some sample data and save the document.



Customer: Max Muster AG
Address: Spittelmarkt, Berlin
Project Description: Installation und Anbindung von Lotus Connections (PoC) abgeschlossen. Test TDI Assembly Lines erfolgreich. Beispielhafte Integration von SharePoint in die Connections Communities anhand von 2 Communities demonstriert.
Employee: Harry Hirsch
Date: 12/12/2011
Abstract: Installation und Anbindung von Lotus Connections (PoC) abgeschlossen. Test TDI Assembly Lines erfolgreich. Beispielhafte Integration von SharePoint in die Connections Communities anhand von 2 Communities demonstriert.
Orders: readme.pdf
My Picture:

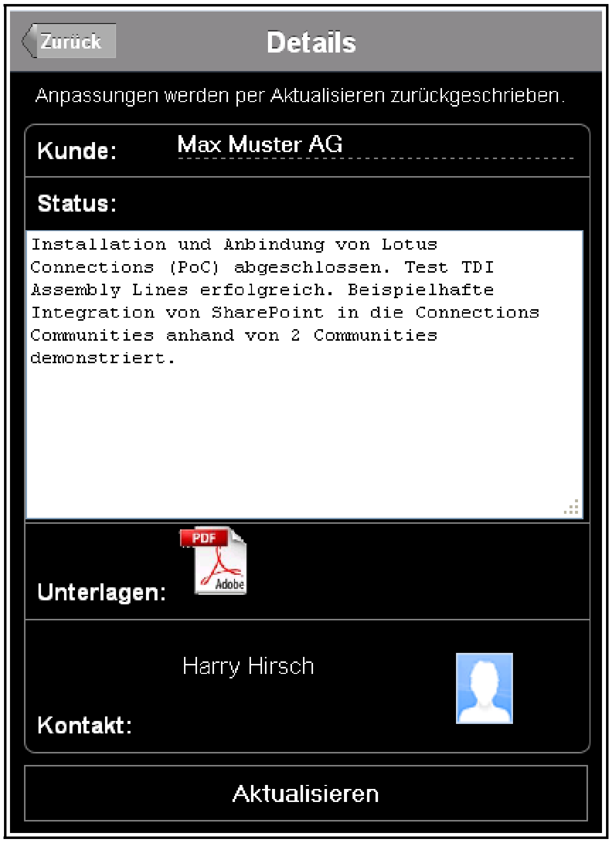
45. Open the **mobileDemo.xsp** and run it within a Browser.





12.12.2011

Max Muster AG
Installation und Anbindung von Lotus Connections (PoC) abgeschlossen. Test TDI Assembly Lines erfolgreich. Beispielhafte Integration von SharePoint in die Connections Communities anhand von 2 Communities demonstriert.



Anpassungen werden per Aktualisieren zurückgeschrieben.

Kunde: Max Muster AG

Status:

Installation und Anbindung von Lotus Connections (PoC) abgeschlossen. Test TDI Assembly Lines erfolgreich. Beispielhafte Integration von SharePoint in die Connections Communities anhand von 2 Communities demonstriert.

Unterlagen:

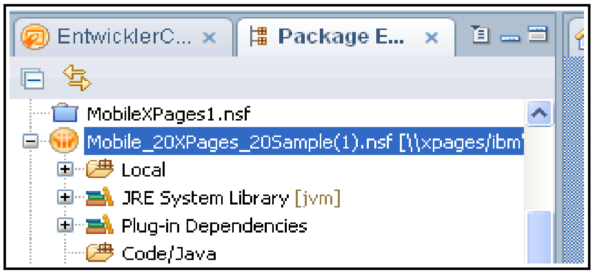
Kontakt: Harry Hirsch

46. Now, point your Android Emulator to the XPage and have the “real” mobile experience...

Use PhoneGap in XPages

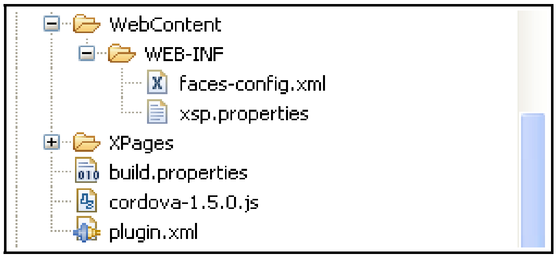
Now, we will demonstrate how to build hybrid applications using PhoneGap for the native stuff and XPages as the web back-end.

1. Navigate to the project's root folder within your Project Explorer perspective in Domino Designer.



MobileXPages1.nsf
Mobile_20XPages_20Sample(1).nsf [\\xpages/ibm
Local
JRE System Library [jvm]
Plug-in Dependencies
Code/Java

2. Add **cordova-1.5.0.js** from your PhoneGap (Android) folder into the **WebContent** folder (via **Import > General > File System**).

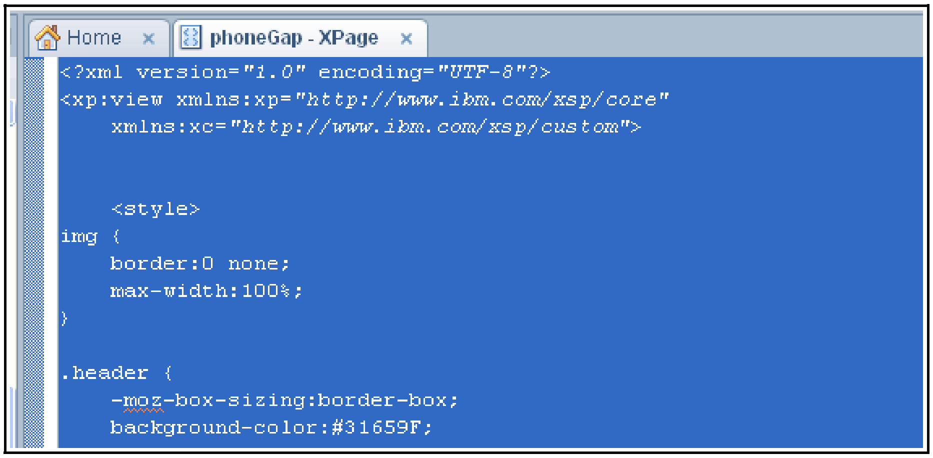


WebContent
WEB-INF
faces-config.xml
xsp.properties
XPages
build.properties
cordova-1.5.0.js
plugin.xml

3. Create a new XPage within your project.

4. Navigate to the **CodeSnippets** (you'll get this file from your moderator) and navigate to the text from Lab 3 / Page 17 in a text editor.

5. **Copy** and **paste** the Lab 3 / Page 17 code into your XPage.

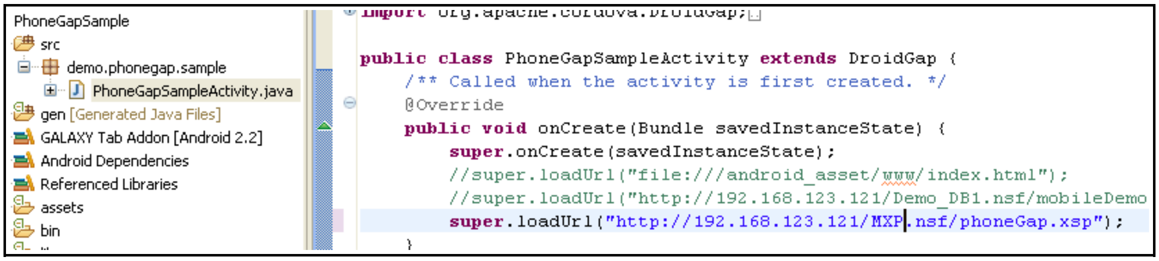


```
<?xml version="1.0" encoding="UTF-8"?>
<xp:view xmlns:xp="http://www.ibm.com/xsp/core"
    xmlns:xc="http://www.ibm.com/xsp/custom">


    <style>
img {
    border:0 none;
    max-width:100%;
}

.header {
    -moz-box-sizing:border-box;
    background-color:#31659F;
```

6. Open Eclipse and navigate to your PhoneGap project that we have build in a previous lab.

7. Navigate to the Activity class we have created and change the code to point to your XPage.

   Note: You might have to run **ipconfig** to find out your IP address.



```
import org.apache.cordova.DroidGap;

public class PhoneGapSampleActivity extends DroidGap {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //super.loadUrl("file:///android_asset/www/index.html");
        //super.loadUrl("http://192.168.123.121/Demo_DB1.nsf/mobileDemo
        super.loadUrl("http://192.168.123.121/MXP.nsf/phoneGap.xsp");
    }
```

8. Open your mobile application using the Android emulator.

   Note: The Android emulator doesn't provide any coordinates to display, so the XPage now should display a map with only one pin linked to the customer address you provided in the Notes document.

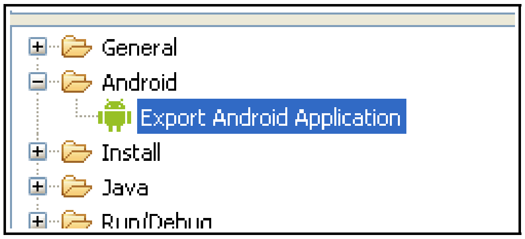
   If you click **Show my current position**, the second pin will show the Maritim location in Gelsenkirchen – because this is the fallback scenario in case of no data available.

A “real” Android device would display both your current and the customer's coordinates on the map.

Deploying your Android project as a native app

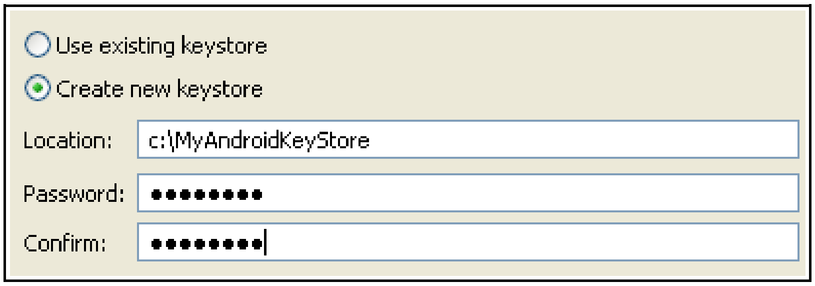
You could export your application as an APK from Eclipse and use this package for installation on your Android device.

1. Right-click on your project in Eclipse and select **Export > Android > Export Android Application.**



2. Click **Next > Next**.

3. You will need to have to setup a new key store that will contain the key for signing your application. The signature is required for deployment on the device.

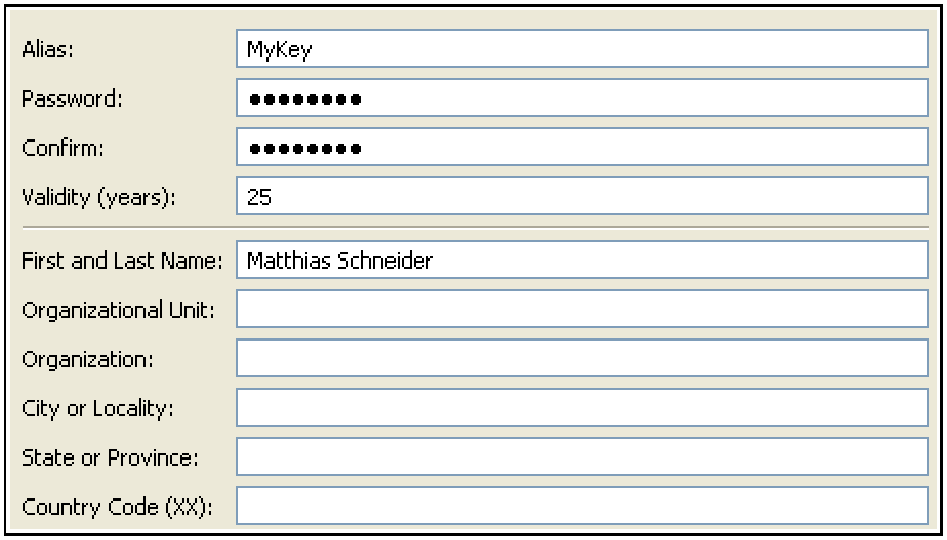


○ Use existing keystore
● Create new keystore

Location: c:\MyAndroidKeyStore
Password: ●●●●●●●●
Confirm: ●●●●●●●●

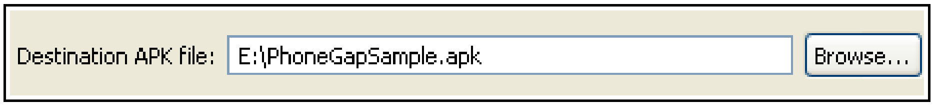
4. Click Next.

5. You will need to provide some basic information for the party you'd like to create a key for



Alias:	MyKey
Password:	●●●●●●●●●
Confirm:	●●●●●●●●●
Validity (years):	25
First and Last Name:	Matthias Schneider
Organizational Unit:	
Organization:	
City or Locality:	
State or Province:	
Country Code (XX):	

6. Provide an APK file name for your packege and click **Finish.**



Destination APK file: E:\PhoneGapSample.apk

7. Now, you could copy the APK to your Domino server (e.g. to **<Domino>\data\domino\html\download\my_apks**) and access it from an Android browser for installation.

Summary

In this lab, you created a mobile XPages project derived from the Extensions Library. You customized the provided application components and built an XPage coupled with the PhoneGap app we previously created. The result was a hybrid application where most of the application logic remains on the server.

As the last step, you exported your app for installation on your mobile device.

End of exercise