Advanced Servoy Mobile
Custom UI and SQLite

Paolo Aronne

Agenda

Mobile Client architecture

Creating custom UI elements

Integrating 3rd party JavaScript libraries

Using SQLite for Unlimited offline storage

HTML5-based, offline client

Technology stack

- Google Web Toolkit (GWT)
- JQueryMobile
- PhoneGap (optional)

Main features

- Runs in the browser
- Works when offline
- Built-in bi-directional sync
- Deploy as WebApp or Native App

Mobile Client architecture

Any HTML/CSS/JavaScript based component can be used

- JQueryMobile components
- Plain HTML elements
- 3rd party components

Available tools

- Bean element to inject custom HTML
  - plugins.mobile.getMarkupId(element.xxxx)
  - Access DOM directly (plain DOM API or JQuery)
  - Solution.onOpen, Form.onShow/onRecordSelection events
  - Media Library to include 3rd party libs

Servoy events

solution.onOpen:

form.onLoad: DOM not yet be available

form.onShow(firstShow): DOM available, check firstShow to do one-time init

form.onRecordSelection: to update based on current selection

JQuery/JQueryMobile events

documentready: document is ready at firstForm.onShow

create: enchant raw elements into JQM Widgets

PhoneGap events

deviceready:device is ready when solution is open

Display custom UI using the Bean

The Bean element inserts an empty DIV into the HTML markup

Set the bean's innerHTML using:

```
elements.myBean.innerHTML = yourHTMLString
```

Interact with your custom UI

Target the bean element in scripting:

```
var id = plugins.mobile.getMarkupId(elements.myBean)
$('#' + id). //your code here
```

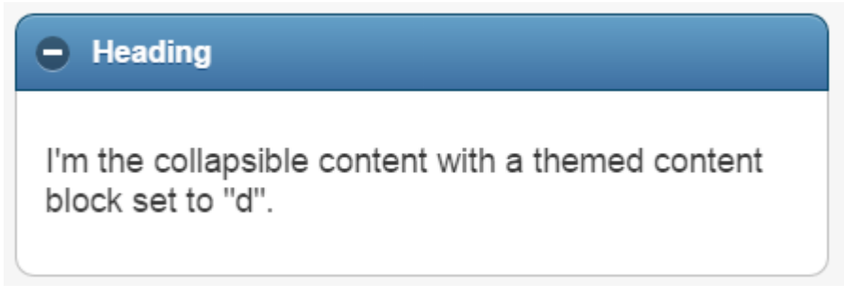
You can assign your own ID's to the injected HTML

Warning: ID's in markup should be unique for all the solution

JQM Widget

It is a feature-rich mobile component which has is own events and methods

JQM process the HTML elements having the **data** attribute to create widgets



JQM create event

when the form is shown for the first time a JQM page is created

JQM processes and transforms the content of the beans into Widgets

Warning: beans will not be processed after the page is loaded.

Create a collapsible Widget

Set the bean’s innerHTML at onRecordSelection event:

```
elements.myBean.innerHTML='<div data-role=”collapsible”>'
                                    <h3>'+ foundset.name +' </h3>
                                    <p>'+ foundset.job_title +' </p>
                                </div>'
```

Trigger on a parent element the JQM create event to display the widget

Note: the activePage is a parent element of the bean.

```
$.mobile.activePage.trigger('create')
```

Bind events to custom UI

Bind Servoy method to HTML/JQM events: **onclick**, **href, swipe**

HTML events are executed from root scope: no form context

Provide fullpath to the method

```
elements.myBean.innerHTML='<div data-role=”button”
onclick= ”window.forms.formName.methodName('+ foundset.pk +')”/>
```

The object window holds the Servoy **forms** object

Creating custom UI elements

Demo

- JQueryMobile Widgets
- JQueryMobile swipe

- Other JQueryMobile widgets

http://view.jquerymobile.com/1.3.1/dist/demos/

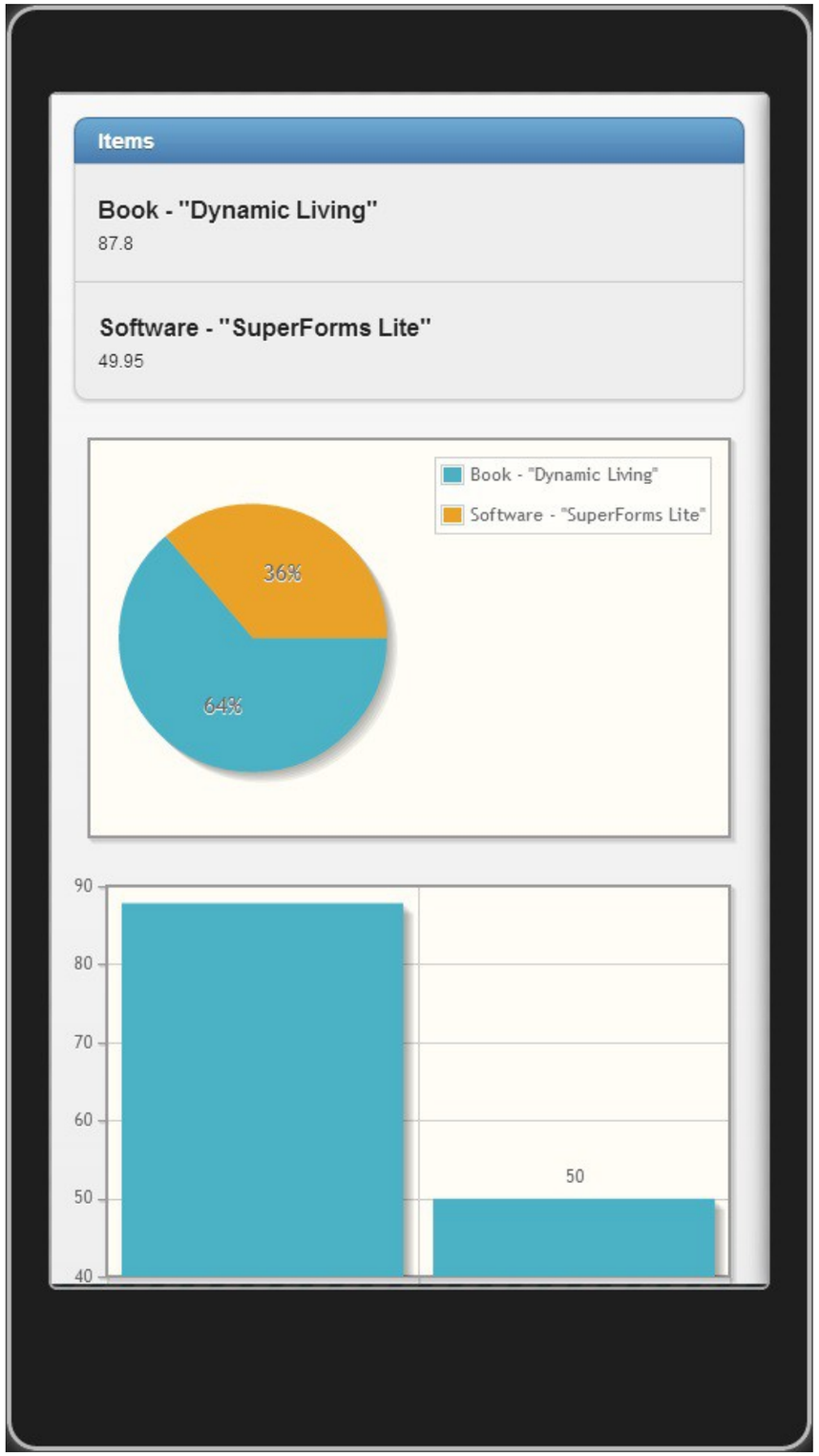
Creating custom UI elements

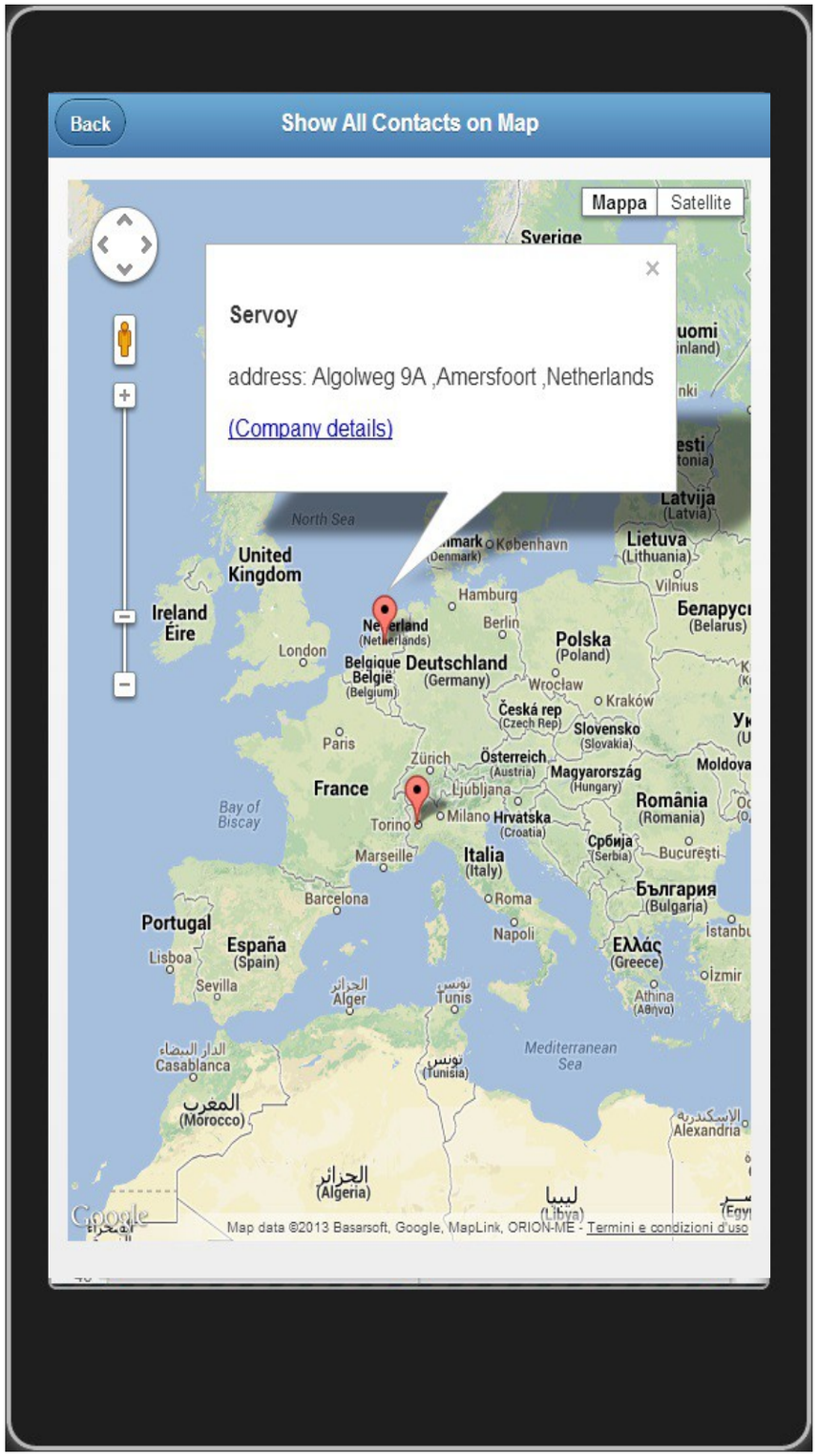
Bundling additional (3rd party) JavaScript and CSS resources

- .js and .css files can be added to the media library
- Any .js and .css file in the media lib will be added to the DOM automatically
- During export define on which order add the .js and .css to the DOM.
- If the library is not available offline append it to the DOM at onOpenSolution

Access the 3rd party library anywhere in your code

```
var map = new google.maps.Map($('#map-canvas'), mapOptions);
```





Integrating 3rd party libraries

Use PhoneGap/Cordova to

- Wrap Servoy Mobile Solution as Native App
- Integrate with Device API’s: Camera, File...

Use PhoneGapBuild to

- Create builds for supported platforms w/o SDK’s
- Include plugins: Bluetooth, Push..

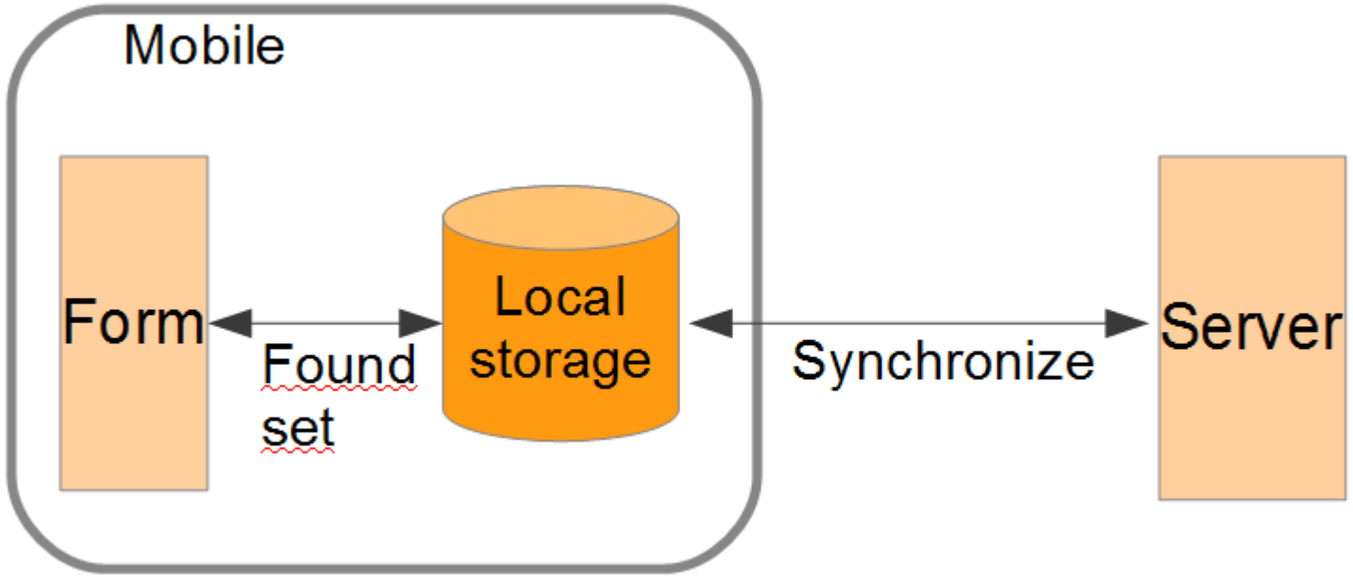
Available Plugins at:

https://build.phonegap.com/plugins

Mobile working with data

- Store data into offline storage
- Simplified interface: not suitable to navigate big amount of data
- Selective synchronization of data
- Retrieve additional data with online search

Built-in process



SQLite on Mobile

Extend the offline storage capacity

- Web SQL: W3C Specification
- PhoneGap SQLite plugin: available for PhoneGapBuild

Web SQL Support

- Android : 2.0 +
- Apple iOS: 3.2 +
- Desktop Browsers: Chrome, Safari, Opera

SQLite plugin Support

- Phonegap 3.0+

Web SQL/SQLite plugin

- 'Unlimited' offline storage (Web SQL limited to 50 Mb on iOS)
- relational database: SQLite dialect
- transactional
- asynchronous

Use Case

- as user I would like to have available in offline mode all information of interest
- e.g. all products, all customers data, all locations...

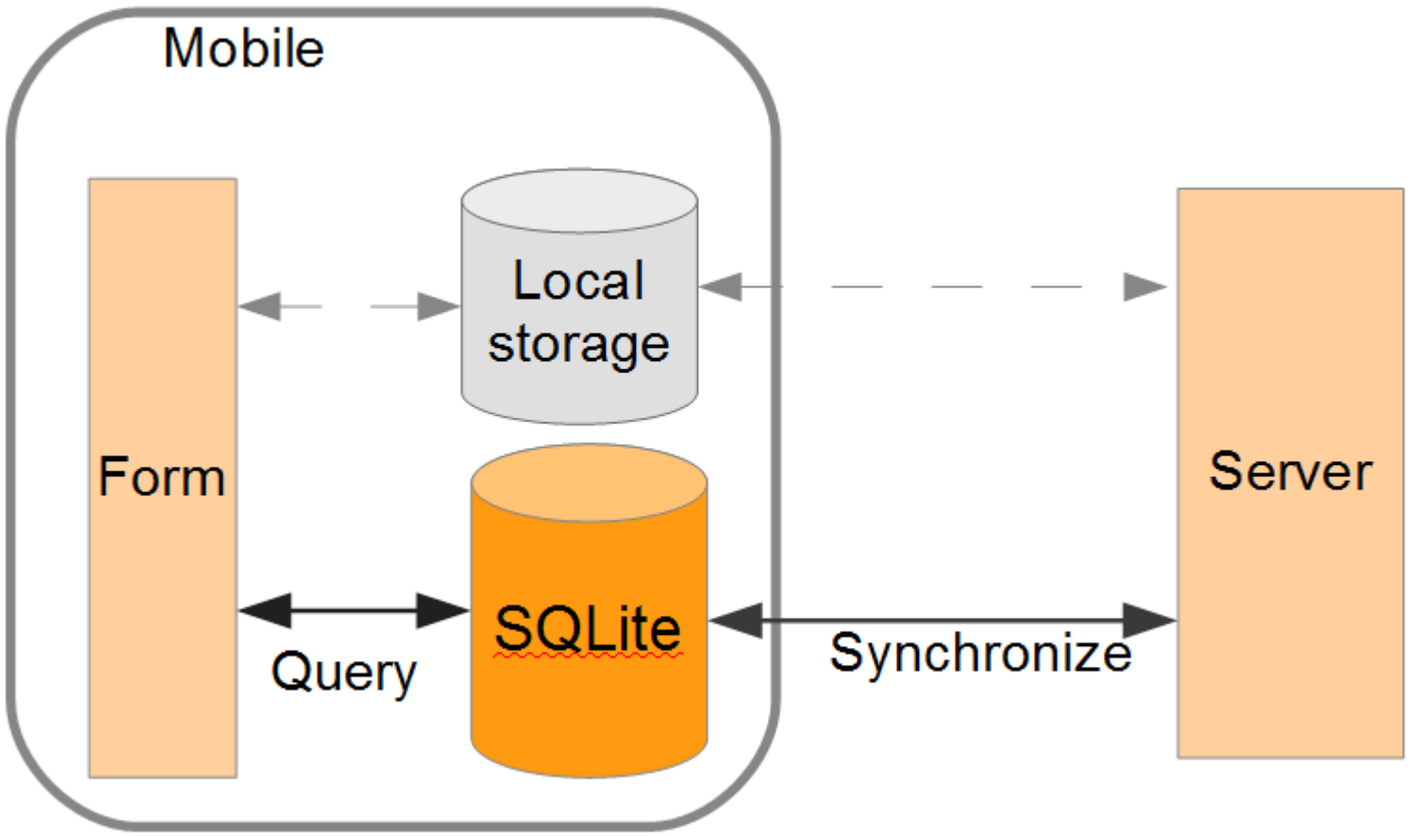
How to use SQLite in Servoy

SQLite setup: create database

scripting to synchronize data with server

SQLite scripts for querying

r/w to localstorage using foundset



SQLite on Mobile

Setup SQLite in Servoy

Open connection to SQLite at onSolutionOpen

Create tables

```
var db = window.openDatabase('myWebSQL', '1.0', 'this is my
first db', 10000000);

db.transaction(
   function (tx) {
   tx.executeSql('CREATE TABLE IF NOT EXISTS foo (id unique,
text)');
}, SQLsuccessCallback, SQLerrorCallback )
```

Using SQLite in Servoy

Execute SQLite Query

Store result rows as new record in foundset or use directly in custom UI

```
db.transaction( function (tx) {
   tx.executeSql('SELECT * FROM foo', null, errorCallback,
   successCallback);
})
```

```
function successCallback (tx, result) {
      for (i=0; i < result.rows.lenght, i ++) {
            var record = results.rows.item(i);
          // do something with record
      }
}
```

Servoy Synchronization Framework (Concept Car)

- use Servoy Framework to synchronize data on SQLite Mobile
- standardize synchronization process and fill the gap with the built-in sync
- Keep focus on your own business logic

Other considerations

- think about using the SQLite database in read-only mode
- you can use localstorage to push changes into server
- keep foundset size small, limit SQLite query results
- Web SQL and SQLite plugin have similar API structure, use Web SQL during development and test on the Browser

Now:

Multitenancy with PostgresSQL

12.10 pm

Integrating WebSpeech into your application