Responsive HTML5 single Page Apps

@j2r2b, @ZimMatthias BSI Business Systems Integration AG
Topics for Today

➡ Scout Intro
➡ Resources to get started

Exercises
1. «Hello World» add Form Fields
2. «Hello World» add your Theme
3. «Contacts» add your Entity (the big one ...)

Intro
What is Scout?

Business Application Framework

➡ Based on **Java** and **HTML5**
➡ **Multi Device** support, **Modular** Apps, ...

Framework Goals

➡ **Long Term** Strategy (enterprise apps live > 10 years)
➡ Boosts **Productivity** (producing software in Switzerland ...)
➡ **Easy** to learn (new team members productive in 1-2 weeks)
Framework Features

- Multi Device Support
- Modular Applications
- Java Application Model
- HTML5 Rendering
Multi Device Support

Single Code Base

➔ Desktop
➔ Tablet
➔ Mobile
Modular Applications

Clean Modularization

- Layers - Technical Modules
- Slices - Business Modules
Scout Application Model

Java Application Model

⇒ Clean Business Code
⇒ Long Term Maintainability
HTML5 Rendering

Current Standards

- HTML5, CSS3, JavaScript
- Styling and Theming

![HTML5 Rendering Example]

- Current Standards
- HTML5, CSS3, JavaScript
- Styling and Theming

![HTML5 Rendering Example]
2 Lessons for the Future
Two Lessons for the Future

➔ UI Technologies are «short-lived»
➔ Scout Code is maintainable
UI Technologies are «short-lived»

Scout UI Technologies
➔ 2002 Swing
➔ 2006 SWT
➔ 2011 RAP
➔ 2015 HTML5

Substantial Framework Efforts
New UI Technology every 4 years

Minor Project Efforts
Usually < 3% of Project Codebase affected
«Today’s Trend is tomorrow’s Legacy»
UI Dos and Don'ts

Don’ts
- Don’t mix UI Technology and business code
- Don’t «marry» specific UI Technologies

Dos
- Think long Term
- Abstract UI Technologies
- Use a Framework like Scout that enforces this Abstraction
Scout Code is Maintainable

Example: BSI CRM

- Based on Scout since 1999
- Scout defines an Application Model
- BSI CRM is based on this Model
- Screenshots over Time ...
... 1999
... 2002

<xml />
… 2007
Eclipse Scout 2012
Eclipse Scout 2015/16
Scout Code is Maintainable

Example: BSI CRM
- Based on Scout since 1999
- Scout defines an Application Model
- BSI CRM is based on this Model

Observations
- CRM Code was never thrown away
- Migrations from Tech-Stack to Tech-Stack
- Scout had a big Part in this
Eclipse Scout

Future Proof Business Applications
4.2. Tutorial Overview

This tutorial walks you through the implementation of a Scout application consisting of a frontend and a backend application. The frontend application contains outlines with navigation trees, pages to present information in tabular form and forms to view and edit data.
Eclipse Scout: Technical Guide

Scout Team – scout-dev@eclipse.org – Version 6.0

Introduction

This technical guide documents the Scout architecture and describes important concepts used in Scout.

This document is not complete. Contributions are welcome!
If you like to help, please create a pull request.
Thanks!

Repository:
https://github.com/BSI-Business-Systems-Integration
This branch is 858 commits ahead of eclipse:master.

judithgull remove unnecessary fielddata
<table>
<thead>
<tr>
<th>Topic</th>
<th>Replies</th>
<th>Views</th>
<th>Last message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Doubts operation</td>
<td>2</td>
<td>2315</td>
<td>Wed, 19 October 2016 13:24 By: Nelson Florez</td>
</tr>
<tr>
<td>PlannerField in 6.0</td>
<td>1</td>
<td>1084</td>
<td>Mon, 17 October 2016 22:27 By: Stefan Mutschler</td>
</tr>
<tr>
<td>[neon] M6 mobile support</td>
<td>30</td>
<td>17401</td>
<td>Fri, 14 October 2016 03:22 By: Peter Pfeifer</td>
</tr>
<tr>
<td>[neon] Login page fails to show for french</td>
<td>6</td>
<td>1287</td>
<td>Wed, 12 October 2016 14:29 By: Urs Beeli</td>
</tr>
</tbody>
</table>
Scout Framework

Boost your productivity. Framework and tooling for writing business applications.

@EclipseScout

Switzerland
eclipse.org/scout

Joined November 2009

129 Photos and videos

Tweets

the style.css is EPL you may do with it what fits your needs ;-)
Exercise 1
«Hello World» add Form Fields
1. Find form class
2. Find layout container class
3. Position cursor at location to add the field
4. Use Content Assist <Ctrl><Space>
5. Select form field type and fill in boxes
6. Update the form service method
Exercise 1: Add Form Fields

Recommended

- Add `DateTimeField «ServerTime»`
  - Overwrite `getConfiguredGridW`
  - Set the current time in the form service method `load`
- Add `BooleanField «ShowTime»`
  - Overwrite `execChangedValue` to control visibility of time field. **Hint:** `getServerTimeField().setVisibility(...)`

Optional

- Add some more fields
- Change rendering to tablet or mobile. **Hint:** Use browser developer tools (e.g. `<Ctrl><Shift><i>` with Chrome)
Exercise 2
«Hello World» add your Styling
Message from Server

Message: Hello mzi!

Normal Text: *hello*

Console Text: >hello

Set text mandatory

Server Time: 10/21/2016 1:47 PM

Show Time
Scout Styling and Theming

Technology

→ CSS3 + Less

CSS Organisation

→ Global Less files (colors, fonts, sizes, ...)
→ Component specific Less files (Form, Field, Button...)
→ Optional: Project specific Less files
→ Themes: Named set of Less files
«Contacts» Styling Demo
Two Examples

Global Styling

➤ Globally change application colors, fonts, ...
➤ Simple and low effort
➤ **Example**: Color of data fields

Styling of Components

➤ Change individual elements
➤ Powerful but more CSS knowledge required
➤ **Example**: Appearance of dialogs
Message from Server

Message
Hello mzi!

Normal Text
*hello*

Set text mandatory

Server Time
10/21/2016
1:47 PM

Show Time
1. CSS/Less files in module `helloworld.ui.html` folder `src/main/js`
2. Global styling CSS files in folder `scout/style`
3. Component specific CSS in folders `scout/...` (eg. `Scout/form/fields`)
4. Theme specified in app’s `config.properties`

→ Check `org.eclipse.scout.rt.ui.html-6.0.100.RC4.jar` (see Maven dep.)
→ Work with browser’s developer (e.g. `<Ctrl><Shift><i>` with Chrome)
Exercise 2: Add your Theme and Style

Recommended

➔ Add Theme «my-theme»
  ➔ Copy colors-application.css to colors-my-theme.css
  ➔ In config.properties, set scout.ui.theme=my-theme

➔ Global Styling: Set navigation background color to orange
  ➔ @navigation-background-color: #FE9A0D;

➔ Component Styling: Set form field input color to blue
  ➔ Add file FormField-my-theme.css
  ➔ @import (less) "scout/form/fields/FormField.css";

Optional

➔ Add fields «Normal Text», «Console Text» and «Set mandatory» and set background for mandatory fields to yellow

➔ Style console text field different to normal text field
  ➔ Hint: Overwrite getConfiguredCssClass on console text field
Exercise 3
«Contacts» add your Entity
«Contacts» add your Entity (Page + Form)

Topics

➔ Install «Contacts»
➔ Repo Organization
➔ Exercise Steps in Events Module

1. Client: Add *LocationTablePage* and Columns
2. Client: Add *LocationForm*
3. Client: Add `{New|Edit}Menu` for Location Table Page
4. Client: Add Form Fields
5. Server: Implement *LocationService* .*getLocationTableData*, .*create*, .*load* and .*store*
«Contacts» Repo Organisation
«Contacts» add LocationTablePage

1. New package ...event.client.location
2. New Wizard, select Scout Page
1. Events Client: **LocationTablePage**: Set Title, add Columns to Table
2. Events Server: **LocationService**
3. Events Client: **EventOutline**: In `execCreateChildPages` Add location table page → `pageList.add(new LocationTablePage());`
LocationTablePage
Columns and Link to Event Outline

→ Read «Beginners Guide»: Creating the Person Page
→ Create the following Columns:
  - **LocationIdColumn**: Primary Key
    see `EventTablePage.EventIdColumn`
  - **NameColumn**: String column
  - **CityColumn**: String column
  - **CountryColumn**: Smart column
    see `EventTablePage.CountryColumn`
→ Add the Location Table Page to the Events Outline
  - Read Link the Person Page to the Contacts Outline
  - Adapt method `execCreateChildPages`
1. Select package `...events.client.location`
2. New Wizard: Select Scout Form → «LocationForm»
3. Result: `LocationForm` and new `LocationService` methods
1. Add member field `locationId` to `LocationForm`
2. Add `EditMenu` and `NewMenu` to `LocationPage`
   Hint: Copy from `EventPage` and `EventForm` and modify...
«Contacts» LocationForm

Form Layout

- Scout Form Layout Intro
- Target Layout for LocationForm
- Add Container Fields and Value Fields

Form Field Validation

- NameField: getConfiguredMandatory
- Read «Beginners Guide»: Form Field Validation
- CityField: execValidateValue, city to start with «L»
Scout Form Layout Intro
# Scout Form Layout Intro

![Form Layout Diagram](image)

<table>
<thead>
<tr>
<th>Field</th>
<th>Input Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Name</td>
<td>Text</td>
</tr>
<tr>
<td>Last Name</td>
<td>Text</td>
</tr>
<tr>
<td>Date of Birth</td>
<td>Date</td>
</tr>
<tr>
<td>Gender</td>
<td>Radio</td>
</tr>
<tr>
<td>Street</td>
<td>Text</td>
</tr>
<tr>
<td>Phone</td>
<td>Text</td>
</tr>
<tr>
<td>Location</td>
<td>Text</td>
</tr>
<tr>
<td>City</td>
<td>Text</td>
</tr>
<tr>
<td>Country</td>
<td>Text</td>
</tr>
<tr>
<td>Mobile</td>
<td>Text</td>
</tr>
<tr>
<td>Email</td>
<td>Text</td>
</tr>
</tbody>
</table>
Scout Form Layout Intro
«Contacts» LocationForm Layout

MainBox (GroupBox)

TopBox (GroupBox with getConfiguredGridColumnCount = 1)

Name

City | Country

LocationBox (SequenceBox)
«Contacts» LocationForm Layout

```java
@Order(1000)
public class MainBox extends AbstractGroupBox {

    @Order(1000)
    public class TopBox extends AbstractGroupBox {
        protected int getConfigureGridColumnCount() {
        }
    }

    @Order(1000)
    public class NameField extends AbstractStringField {
        protected String getConfigureLabel() {
        }
    }

    @Order(2000)
    public class LocationBox extends AbstractSequenceBox {
        protected boolean getConfigureAutoCheckFromTo() {
        }
    }

    @Order(1000)
    public class CityField extends AbstractStringField {
        protected String getConfigureLabel() {
        }
    }

    @Order(2000)
    public class CountryField extends AbstractSmartField<String> {
        protected String getConfigureLabel() {
        }
    }

    @Override
    protected Class<?> extends ILookupCall<String>> getConfigureLookupCall() {
        return CountryLookupCall.class;
    }
}
```
Farben:
- 254/153/21
- 0/130/161
- 127/127/127

Für vollflächige Inhalte geeignet

Special:
- Ausserhalb der Linien erlaubt

The following fields are required:
- Name

The following fields are invalid:
- City: Values for the City field need to start with "L".

OK
1. NameField: `getConfiguredMandatory`
2. Read «Beginners Guide»: Form Field Validation
3. CityField: `execValidateValue`, city to start with «L»
«Contacts» update Persistence for Locations

1. Uncomment `DatastoreLocation`
2. Implement Methods in `LocationService`  
   `.getLocationTableData`, `.create`, `.load` and `.store`
Thanks
and please, cast your Vote

Evaluate the Sessions
Sign in and vote at eclipsecon.org

-1 0 +1