@EclipseScout
The Future Proof Java Web-framework

OOP ‘16 – Matthias Zimmermann
Agenda

→ Why you need «future-proof»
→ 16 Years of Scout
→ Eclipse Scout Neon: Java and HTML5
  – Model: Java
  – Building: Maven
  – Theming and Styling: CSS3
→ Wrap up
Why you need «Future Proof»
Enterprise Applications

The Good

→ Represents Core Process Knowledge
→ Make business more efficient

The Bad

→ Large initial Costs
→ Large running Costs
Enterprise Applications

The Ugly

- Include Legacy Technologies
- Risk and Cost increasing
- Look old

«Now, let’s look at a typical example»
What did go wrong?

- Focus on **latest technology**
- **Keep adding** new technologies
- **Fail to remove** legacy technologies
Can we do better?

➔ Think long term
➔ Choose boring* technology
➔ Keep technology stack minimal
➔ Keep business logic separate from technology

«And, what do we get in return?»

*http://mcfunley.com/choose-boring-technology-slides (oscon ‘15)
Focus on Business Logic

Modular Architecture

«Boring» Technology

Minimal Stack
Benefits of «Future Proof» Applications

- Protect investment in business logic
- Keep technology healthy
- Keep developers happy
- Avoid rewrites

Want this?
- Eclipse Scout can help
- How? Let’s look at the Scout story ...
16 Years of Scout
Let’s go back to ...
... 1999
... 1999
... 1999
... 2002

<xml />

JAVA™
... 2005

Java
<xml />
... 2007
Eclipse Scout 2011
Eclipse Scout 2012
Eclipse Scout 2016
Our learnings from last 16 years

► Zero rewrites
► Many Scout Upgrades
► UI technologies come and go
► Eclipse Scout helps to «future proof»
Eclipse Scout
Eclipse Scout

Framework

- Built for Enterprise Applications
- Based on Java and HTML5
- Multi Device support

Goals

- Support Long Term strategy (enterprise apps live > 10 years)
- Boost productivity (salaries in Switzerland ...)
- Easy to learn (get productive in 1-2 weeks)
Eclipse Scout – The Model
Eclipse Scout Model

What to model?

➔ User Interface
➔ Services
➔ Security

Level of Detail?

➔ Tradeoff flexibility and efficiency
➔ DON’T depend on any UI technology
➔ DO allow for UI rendering
String Fields – The last 20 years

Type choices, press Enter.

From file ................. > MYFILE_____ Name

Group key

Verwendungszweck (intern) | Test
Antragsteller            | Muster Hans

First name | David

AS400
SAP R3
JSP
Eclipse Scout
OfStringField in Scout

```java
@Order(2)
public class FirstNameField extends AbstractStringField {
    @Override
    protected String getLabel() {
        return TEXTS.get("FirstName");
    }
}
```
Eclipse Scout – Java and Building /w Maven
Until Eclipse Mars ...

myapp-server.product

P2 Update Site

myapp.target

<bundle>

<bundle>.test

plugin.xml

MAVEN TYCHO

MANIFEST.MF
... With Neon

MAVEN
Maven Standard
Multi Module Projects

→ {project name}
→ {project name}.server
→ {project name}.shared
→ {project name}.client
→ {project name}.ui.html

→ {project name}.server.app.dev
→ {project name}.ui.html.app.dev

→ {project name}.server.app.war
→ {project name}.ui.html.app.war
Maven Build
m2e for Eclipse IDE
Maven Build
Command Line

```bash
C:\Users\jbr\code\ugm2015\com.company.sandbox.parent>mvn clean verify
[INFO] Scanning for projects...
[INFO] Reactor Build Order:
[INFO] [INFO] con.comany.sandbox.parent
[INFO] con.comany.sandbox.shared
[INFO] con.comany.sandbox.server
[INFO] con.comany.sandbox.client
[INFO] con.comany.sandbox.server_app.war
[INFO] con.comany.sandbox.ui.html
[INFO] con.comany.sandbox.ui.html_app.war
[INFO] -----------------------------
[INFO] Building com.company.sandbox.parent 1.0.0-SNAPSHOT
[INFO] -----------------------------
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ com.company.sandbox.parent ---
[INFO] --- build-helper-maven-plugin:1.9.1:add-source (add-source) @ com.company.sandbox.parent
[INFO] Source directory: C:\Users\jbr\code\ugm2015\com.company.sandbox.parent\src\main\webapp
[INFO] Source directory: C:\Users\jbr\code\ugm2015\com.company.sandbox.parent\src\generated\java
[INFO] >>> maven-source-plugin:2.4:jar (attach-sources) > generate-sources @ com.company.sandbox
[INFO] --- build-helper-maven-plugin:1.9.1:add-source (add-source) @ com.company.sandbox.parent
[INFO] Source directory: C:\Users\jbr\code\ugm2015\com.company.sandbox.parent\src\main\webapp
[INFO] Source directory: C:\Users\jbr\code\ugm2015\com.company.sandbox.parent\src\generated\java
```
Eclipse Scout – Themes and Styling
Themes and Styling with Scout

Technology

→ CSS3 + Less

CSS Organisation

→ Global Less files (colors, fonts, sizes, ...)
→ Component specific Less files (Form, Field, Button...)
→ Optional: Project specific Less files
→ Theme: Named set of Less files

From Less to final CSS

1. Project specific files override Scout defaults
2. Single big CSS is assembled, processed and minified
Scout Theming
Scout Styling

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
Wrap up
Wrap up

➔ Enterprise Apps: **Long Life** (> 10 years)
➔ UIs Technologies: **Short Life** (<5 years)
➔ Think Longterm

Eclipse Scout

➔ Helps «**future proofing**» your enterprise apps
➔ Based on **Java** and **HTML5**
➔ Styling with **CSS3** is simple
Thanks

Follow us on @EclipseScout

OOP '16 – Matthias Zimmermann