Eclipse Scout
Future Proof Business Applications

@ZimMatthias  @EclipseScout
Scout Hello World
Scout Beginner’s Guide

Preface

The goal of this book is to get you familiar with the Scout framework in a short time. Scout’s core features and concepts are introduced and explained by writing actual Scout applications. As Scout applications are written in Java, we make the assumption that you are familiar with the Java language and its core concepts.

We hope that this book helps you to get started quickly and would love to get your feedback. This feedback is very valuable to us as it helps to improve both the book’s content and the quality for all future readers.

To allow for contributions to this book, the technical setup and the book’s licence have been selected to minimize restrictions. According to the terms of the Creative Commons (CC-BY) license, you are allowed to freely use, share and adapt this book. All source files of the book including the Scout projects described in the book are available on Github.

1. Introduction
## Scout Demo Application

### Contacts
- **First name**: Marius
- **Last name**: Brunnial
- **City**: Baden
- **Country**: Switzerland
- **Events**: 0

<table>
<thead>
<tr>
<th>First name</th>
<th>Last name</th>
<th>City</th>
<th>Country</th>
<th>Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marius</td>
<td>Brunnial</td>
<td>Baden</td>
<td>Switzerland</td>
<td>0</td>
</tr>
</tbody>
</table>

### Organizations
- **BSI Business Systems Integration AG**

### Events
- **Name**: Stefan Leicht Vogt
- **Date of birth**: 01/01/1987
- **Gender**: Male

### Contact Info
- **Street**: Mutschweg 3
- **Location**: Gebenstock, Switzerland

### Home Page
- [https://www.bsi-software.com](https://www.bsi-software.com)
Scout Integrating JavaScript Libraries
Scout Commercial Application
What about Migration?

Mars

```java
@Order(1000.0)
public class MessageField extends AbstractStringField {

    @Override
    protected String getConfigurerLabel() {
        return TEXTS.get("Message");
    }
}

@Override
protected void execLoad() throws ProcessingException {
    IDesktopService service = SERVICES.getService(DesktopService.class);
    DesktopFormData formData = new DesktopFormData();
    exportFormData(formData);
    importFormData(formData);
}

@Override
public DesktopFormData load(DesktopFormData formData) throws ProcessingException {
    formData.getMessage().setValue("Hello World");
    return formData;
}
```
Eclipse Scout Neon

What’s new?

➡ Scout Platform: Plain Java
➡ Building: Plain Maven
➡ UI: Plain HTML5

What stays?

➡ Developer Productivity
➡ Application Model (minimizes migration efforts)
➡ Long Term Strategy (enterprise apps live > 10 years)
Thanks

@EclipseScout, @ZimMatthias