

# Server Pages

Web Engineering  
Dierk König  
Christian Ribeaud



# Technology Overview

**Grails**



***Web app platform***

Groovy

*"PHP" for Java*

Geb & Spock

*Test support*

Gradle

*Build system*

Spring Boot

*Enterprise Web*



# The story so far

Static Pages - HTML, CSS

MVC - Model, View, Controller

Static Page

Server Page



# Engineering Aspects

Well-formed & valid HTML, CSS

Avoid duplication (in CSS)

Testing pages and navigation

Validating Models (imp., decl.)

MVC separation of concerns

# Request - Response

server

controller

```
class MyController
  def myAction(en, exam)
```

client

view

Grade Calculator

localhost:8080/static/GradeCalculator.html

continuous assessment grade

final exam

Daten absenden

request

response

# Server Page 1: GSP

use of view binding

```
<p> Your average is <output>${ result }</output>.</p>
```

```
<💡> Back to the <a href="/static/GradeCalculator.html">calculator</a>.</p>
```

# Page structure: four ways

GSP

*Server page with values*

Template

*Local composition*

TagLib

*Global composition*

Layout

*Inverse composition*



# Server Page 2: Template

gsp

```
<tmpl:form_row name="en" label="En" calculatorInstance="${calculatorInstance}" />
```

\_form\_row.gsp

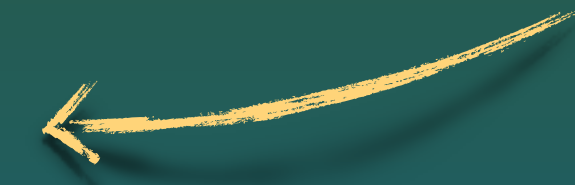
```
<label for='${name}'>${label}</label>
<input type="number decimal" name="${name}" value="${calculatorInstance.getProperty(name)}"
      required="true" min="1.0" max="6.0" id="${name}"
```



# Templates

*static/structural*

Parts of a page - as a page



Avoid in-page duplication

Sep. of concerns - abstraction

Compositional

# Server Page 3: Taglib

gsp

```
<mvc:decorate grade="${calculatorInstance.result}">  
  <output>${calculatorInstance.result}</output>  
</mvc:decorate>
```

## DecorationTagLib

```
def decorate = { attributes, body ->  
  String grade = attributes.grade  
  // ...  
  out << body()
```

# TagLibs

*dynamic content*



Parts of content - as a method

Can enclose a body

Can be used inside tags

All Grails tags are so defined

# Templates & TagLibs

either can be used



as tag

as method

# Server Page 4: Layout

gsp

```
<head>  
  <meta name="layout" content="form"/>
```

layout/form.gsp

```
<h1><g:layoutTitle default="Form"/></h1>  
  
<g:layoutBody/>
```

# Layouts

Imposed embedding - as a page

Inverse composition (sitemesh)

Avoid duplication

Improve consistency

Allow multi-tenant apps

# Page structure: four ways

GSP	<i>Server page with values</i>
Template	<i>Local composition</i>
TagLib	<i>Global composition</i>
Layout	<i>Inverse composition</i>