irian THE SOFTWARE EXPERTS

JSF mit GET-Requests und lesbaren URLs

Michael Kurz | IRIAN

@michi_kurz

http://jsflive.wordpress.com

Agenda

- Motivation
- GET-Unterstützung in JSF 2.x
- RESTful URLs mit OCPsoft Rewrite
- Demonstrationen

Das Web versus JSF

WWW

HTTP GET

HTTP POST

GET-Unterstützung in JSF 2

- **h:link** und **h:button**
- View-Parameter
- View-Actions (ab JSF 2.2)

Navigation mit h:link und h:button

- GET-Navigation
- **h:form** nicht erforderlich

Navigation mit h:link

```
<h:link outcome="/page.xhtml" value="Page"/>
```

```
<a href="/page.jsf">Page</a>
```

h:link mit Parametern

```
<h:link outcome="/page.xhtml" value="Page">
  <f:param name="para" value="1"/>
</h:link>
```

```
<a href="/page.jsf?para=1">Page</a>
```

Navigation mit h:button

```
<h:button outcome="/page.xhtml" value="Page"/>
```

```
<input type="button" value="Page"
       onclick="window.location.href='/page.jsf'"/>
```

View-Parameter

```
<f:metadata>
    <f:viewParam name="id" value="#{myBean.id}"/>
</f:metadata>
```

```
@Named @RequestScoped
public class MyBean {
    private long id;
    public long getId() {return id;}
    public void setId(long id) {this.id = id;}
}
```

View-Actions

- Echte View-Actions mit JSF 2.2
- System-Event **PreRenderViewEvent**
- CODI: **@PreRenderView**

View-Actions mit JSF 2.2

```
<f:metadata>
    <f:viewParam name="id" value="#{bean.id}"/>
    <f:viewAction action="#{bean.load}"
        onPostback="false"
        phase="INVOKE_APPLICATION"/>
</f:metadata>
```

```
@Named @RequestScoped
public class MyBean {
    private long id;
    public long getId() {return id;}
    public void setId(long id) {this.id = id;}
}
```

PreRenderViewEvent

```
<f:event type="preRenderView"
    listener="#{bean.preRender}"/>
```

```
public void preRender(ComponentSystemEvent event) {
    mail = mailService.findById(id);
}
```

Demo: JSF 2 GET-Support

https://github.com/jsflive/jsf-get-examples (jsf-get01)

Post-Redirect-Get mit JSF 2

```
<h:form>
  <h:commandButton action="#{myBean.save}" value="Save"/>
  <h:commandButton value="Cancel" immediate="true"
      action="myPage?faces-redirect=true&amp;
                     faces-include-view-params=true"/>
</h:form>
```

```
@Named @RequestScoped
public class MyBean {
  public String save() {
    return "myPage?faces-redirect=true"
        + "&faces-include-view-params=true";
  }
}
```

Demo: Post-Redirect-Get

https://github.com/jsflive/jsf-get-examples (jsf-get02)

RESTful URLs

http://myshop.at/node.html?mode=list&cat=1234

http://myshop.at/products/books

Warum RESTful URLs?

- Lesbarkeit
- Search Engine Optimization (SEO)
- Vertrauen
- Sieht besser aus

OCPsoft Rewrite

- RESTful URLs für Java EE
- Nachfolger von PrettyFaces

Beispiel 1: Mapping für JSF-URL

http://myshop.at/faces/account.xhtml

http://myshop.at/account

Beispiel 1: Konfiguration mit Java API

```
@RewriteConfiguration
public class MyConfigurationProvider
    extends HttpConfigurationProvider {
  @Override
  public Configuration getConfiguration(ServletContext c) {
    return ConfigurationBuilder.begin()
        .addRule(Join.path("/account/")
        .to("/faces/account.xhtml"));
  }
  @Override
  public int priority() {return 10;}
}
```

Beispiel 1: Konfiguration über Annotation

```
@Named
@RequestScoped
@Join(path="/account", to="/faces/account.xhtml")
public class AccountPage {}
```

Beispiel 2: Mapping für JSF-URL mit Parameter

http://myshop.at/faces/products/details.xhtml?id=123

http://myshop.at/product/123

Beispiel 2: Konfiguration mit Java API

```
@RewriteConfiguration
public class MyConfigurationProvider
    extends HttpConfigurationProvider {
  @Override
  public Configuration getConfiguration(ServletContext c) {
    return ConfigurationBuilder.begin()
        .addRule(Join.path("/product/{id}")
        .to("/faces/products/details.xhtml"));
  }
  @Override
  public int priority() {return 10;}
  }
```

Beispiel 2: Konfiguration über Annotation

```
@Named
@RequestScoped
@Join(path = "/product/{id}",
    to="/faces/products/details.xhtml")
public class ProductDetailsPage {}
```

Demo: Rewrite

https://github.com/jsflive/jsf-get-examples (jsf-get03)

Fazit

- JSF 2 vereinfacht GET-Unterstützung
- JSF 2.2 bringt View-Actions
- OCPsoft Rewrite
- Notwendig fürs Intranet?

Weitere Informationen

Michael Kurz, Martin Marinschek
JavaServer Faces 2.2, dpunkt.verlag

IRIAN JSF@Work Online-Tutorial
http://jsfatwork.irian.at

JSFlive Weblog
http://jsflive.wordpress.com