TDC2012
the developer's conference

Aplicativos JavaEE 6 modularizados com Web Fragments

Kleber Xavier
Instrutor e Arquiteto de Software

Vinicius Senger
Instrutor e Arquiteto De Software

Globalcode

Agenda

- Introdução
- Componentização com JavaEE 6
- Plugabilidade
- DEMO

Globalcode

Introdução

- Componentes web em um aplicativo até JavaEE 5:
  - Frameworks (JSF, RichFaces, etc)
  - Módulos de aplicativo: contendo classes e recursos (páginas, scripts, imagens, css, etc)
- Para instalação não basta apenas adicionar um .jar
- Configuração intrusiva:
  - Alterações no deployment descriptor (web.xml)
  - Adição de descritores proprietários em WEB-INF

Introdução

TDC2012 the developer's conference

- Configurações do web.xml necessárias para JSF:

```
<servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>
    javax.faces.webapp.FacesServlet
  </servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.jsf</url-pattern>
</servlet-mapping>
```

Globalcode

Introdução

TDC2012 the developer's conference

- Configurações do web.xml necessárias para RichFaces:

```
<filter>
    <display-name>Ajax4jsf Filter</display-name>
    <filter-name>ajax4jsf</filter-name>
    <filter-class>org.ajax4jsf.Filter</filter-class>
</filter>
<filter-mapping>
    <filter-name>ajax4jsf</filter-name>
    <servlet-name>Faces Servlet</servlet-name>
    <dispatcher>REQUEST</dispatcher>
    <dispatcher>FORWARD</dispatcher>
    <dispatcher>INCLUDE</dispatcher>
</filter-mapping>
```

Globalcode

Introdução

- Combinação de frameworks aumenta a complexidade do arquivo web.xml.
- As páginas HTML de módulos devem ser publicadas juntamente às páginas do aplicativo principal
- Não é possível empacotar as páginas HTML em um arquivo .jar

Java EE 6

TDC2012 the developer's conference

- Foco na extensibilidade
  - Possibilidade de configuração de aplicativos Web e JSF via anotações
  - Possibilidade de quebra do deployment descriptor (web.xml) em vários fragmentos
  - Páginas HTML podem ser adicionadas em arquivos .jar

Globalcode

Declarando um Servlet

- Exemplo de Servlet com anotação **`@WebServlet`**, usando o atributo **`value`** para definir várias URLs:

```
package br.com.globalcode.tdc2012.exemplo.servlet;

// imports omitidos

@WebServlet({"/faces/*", "*.jsf","*.faces"})
public class FacesServlet extends HttpServlet {

  @Override
  protected void doGet(HttpServletRequest request,
                       HttpServletResponse response)
          throws ServletException, IOException {
    // ...
  }

}
```

Globalcode

Declarando um Filter

TDC2012 the developer's conference

- Exemplo de filtro com anotação **`@WebFilter`**:

```
package br.com.globalcode.tdc2012.exemplo.filter;

// imports omitidos

@WebFilter(servletNames={"FacesServlet"},dispatcherTypes={REQUEST,FORWARD})
public class AjaxFilter implements Filter {

  public void doFilter(ServletRequest request, ServletResponse response,
          FilterChain chain)
          throws IOException, ServletException {
    // ...
  }

}
```

Globalcode

Declarando um Listener

TDC2012 the developer's conference

- Exemplo de listener com anotação **`@WebListener`**:

```
package br.com.globalcode.tdc2012.exemplo.listener;

import javax.servlet.ServletContextEvent;
import javax.servlet.ServletContextListener;
import javax.servlet.annotation.WebListener;

@WebListener
public class ExemploListener implements ServletContextListener {

  @Override
  public void contextInitialized(ServletContextEvent event) {
    // ...
  }

  @Override
  public void contextDestroyed(ServletContextEvent event) {
    // ...
  }

}
```

Globalcode

Plugabilidade

- Mecanismo que permite adicionar a uma aplicação novos componentes (Servlets, Filters, Listeners) contidos em pacotes JARs através do classpath da aplicação Web.
- Suporta componentes anotados em JARs separados;
- Suporta arquivos XML de configuração que complementam o web.xml, chamados de "web fragments";
- Suporta a configuração programática (carregamento dinâmico) de componentes através de códigos em Listeners.

Empacotamento de componentes anotados

- Arquivos **JARs** armazenados no **/WEB-INF/lib** da aplicação Web podem conter componentes anotados para definir **Servlets**, **Filters** e **Listeners**;
- A presença do JAR com componentes anotados permitirá a detecção e deploy automáticos;
- Na ausência do JAR, os componentes não serão instalados;

Fragmentos de configuração

- **Componentes Web** podem ser configurados através de arquivos **XML** contidos **em JARs**;
- O arquivo deve se chamar **`web-fragment.xml`** e ser armazenado em **/META-INF/** do pacote **JAR**;
- Ao colocar o **JAR no /WEB-INF/lib** da aplicação Web resultará na interpretação do arquivo de fragmentos no /META-INF;

Fragmentos de configuração

TDC2012 the developer's conference

```
<web-fragment version="3.0"
              xmlns="http://java.sun.com/xml/ns/javaee"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
              http://java.sun.com/xml/ns/javaee/web-fragment_3_0.xsd">

    <servlet>
        <servlet-name>exemplo01</servlet-name>
        <servlet-class>
                br.com.globalcode.tdc2012.ExemploServlet
        </servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>exemplo01</servlet-name>
        <url-pattern>/exemploServletFragmentado1</url-pattern>
    </servlet-mapping>

</web-fragment>
```

Globalcode

Páginas nos fragmentos

- Além da configuração e classes, também é possível disponibilizar páginas nos fragmentos do aplicativo web;
- As páginas devem estar localizadas dentro do .jar, no subdiretório META-INF/resources
- Isso permite a criação fácil de módulos para aplicativos Web

Criação de módulos

TDC2012 the developer's conference

- Para criação de módulos com web fragments basta seguir os passos:
  - Criar o .jar com o módulo ou plugin desejado, incluindo as páginas e classes;
  - Criar um listener de inicialização dentro do .jar que faz a publicação do plugin em um repositório (por exemplo um EJB @Singleton)
  - Empacotar o .jar junto ao aplicativo Web principal

Globalcode

Exemplo

TDC2012 the developer's conference

- Página do aplicativo principal que carrega menus dinamicamente de acordo com os módulos instalados:

```
<h:form>
    <rich:toolbar width="1000px">
       <c:forEach itens="#{menuController.menus}"
                     var="menu" />
              <ui:include src="#{menu}"/>
       </c:forEach>
    </rich:toolbar>
</h:form>
```

Globalcode

Exemplo

TDC2012 the developer's conference

- EJB Singleton utilizado como registro de módulos:

```
@Singleton
public class MenuRegistryImpl implements MenuRegistry {

        private Set<String> menus = new LinkedHashSet<String>();

        public void addMenu(String caminhoMenu) {
                menus.add(caminhoMenu);
        }

        public Set<String> getMenus() {
                return menus;
        }

        public void removeMenu(String caminhoMenu) {
                menus.remove(caminhoMenu);
        }

}
```

Globalcode

Exemplo

TDC2012 the developer's conference

- Listener de registro de módulos:

```
@WebListener
public class ComprasWebListener implements ServletContextListener {

        @EJB
        MenuRegistry registry;

        public void contextInitialized(ServletContextEvent sce) {
                registry.addMenu("/includes/menu-compras.xhtml");
        }


        public void contextDestroyed(ServletContextEvent sce) {}

}
```

Globalcode

DEMO

Dúvidas

TDC2012
the developer's conference

?

kleber@globalcode.com.br
vinicius@globalcode.com.br

Globalcode