

Zaawansowane Aplikacje Internetowe

Struts 2 framework

Piotr Mazur

Katedra Mikroelektroniki i Technik Informatycznych

Łódź, 19 października 2010

- 1 Struts 2
 - Interceptory

- 2 Tagi JSP

Interceptory

- Pozwalają na wykonanie dodatkowych operacji przed lub po wykonaniu akcji
- Przed wykonaniem każdej akcji wykonuje się zwykle kilka/kilkanaście interceptorów
- Grupę interceptorów określamy mianem **stosu**
- Stos interceptorów można przypisać do **pakietu**

Przykład

```
public class TestInterceptor extends AbstractInterceptor{

    @Override
    public String intercept(ActionInvocation invocation) throws Exception {
        System.out.println("Przed wykonaniem akcji");

        String result=invocation.invoke();

        System.out.println("Po wykonaniu akcji");

        return result;
    }
}
```

- Interceptor powinien implementować interfejs **com.opensymphony.xwork2.interceptor.Interceptor**
- Klasa **com.opensymphony.xwork2.interceptor.AbstractInterceptor** implementuje ten interfejs - można ją rozszerzać
- Interceptory **MUSZA** być thread-safe

Dodatkowe informacje na temat definiowania interceptorów można znaleźć na stronie

<http://struts.apache.org/2.x/docs/writing-interceptors.html>

Uzyskanie dostępu do akcji która ma być (lub była) wykonywana dla danego interceptora

```
ActionSupport action=(ActionSupport) invocation.getAction();
```

Wartość zwrócona przez **getAction** jest klasy **Object**

Lista tagów

- Tagi pozwalają na korzystanie z mechanizmów oferowanych przez **Struts 2** z poziomu strony JSP
- Korzystanie z tagów nie jest wymagane jednak ułatwiają one budowanie widoku aplikacji
 - Automatyczne dodawanie informacji validujących formularz
 - Obsługa języka **OGNL**
- Tagi bazują na wzorcach opracowanych w języku **freemarker**

Kompletna lista tagów dostępna jest na stronie

<http://struts.apache.org/2.x/docs/tag-reference.html>

Tagi **Struts 2** dostępne są w kilku wariantach określanych mianem motywów **theme**

- Wszystkie tagi dostępne są w kilku wariantach
- W zależności od motywu tag może wygenerować inny kod HTML
- Można definiować własne motywy
- Domyślnym motywem jest **xhtml**

Aby korzystać z biblioteki tagów oferowanych przez **Struts 2** należy je dołączyć do strony JSP

```
<%@ taglib prefix="s" uri="/struts-tags" %>
```


Przykłady

s:if

```
<s:if test="%{object.property}">Treść1</s:if>  
<s:elseif test="%{object.otherproperty}">Treść2</s:if>  
<s:else>Treść 3</s:else>
```

s:property

```
<s:property value="object.property" />
```

s:iterator

```
<s:iterator value="lista" >  
  Element listy: <s:property /> <br/>  
</s:iterator>
```

W przykładzie iteratora wykorzystywana jest domyślna wartość przechowywana na stosie

OGNL

Object-Graph Navigation Language - język wykorzystywany przez **Struts 2** w widoku JSP

- Jest to język typu **Expression Language** mający zastąpić wykorzystywany w jsp język **EL**
- Pozwala na dostęp oraz manipulację **Grafem Obiektów** z poziomu strony JSP

Dokumentację można znaleźć na stronie

```
http://www.opensymphony.com/ognl/
```

The End