

互联网应用开发技术

Web Application Development

第11课

WEB后端框架-STRUTS MVC

Episode Eleven

Struts MVC

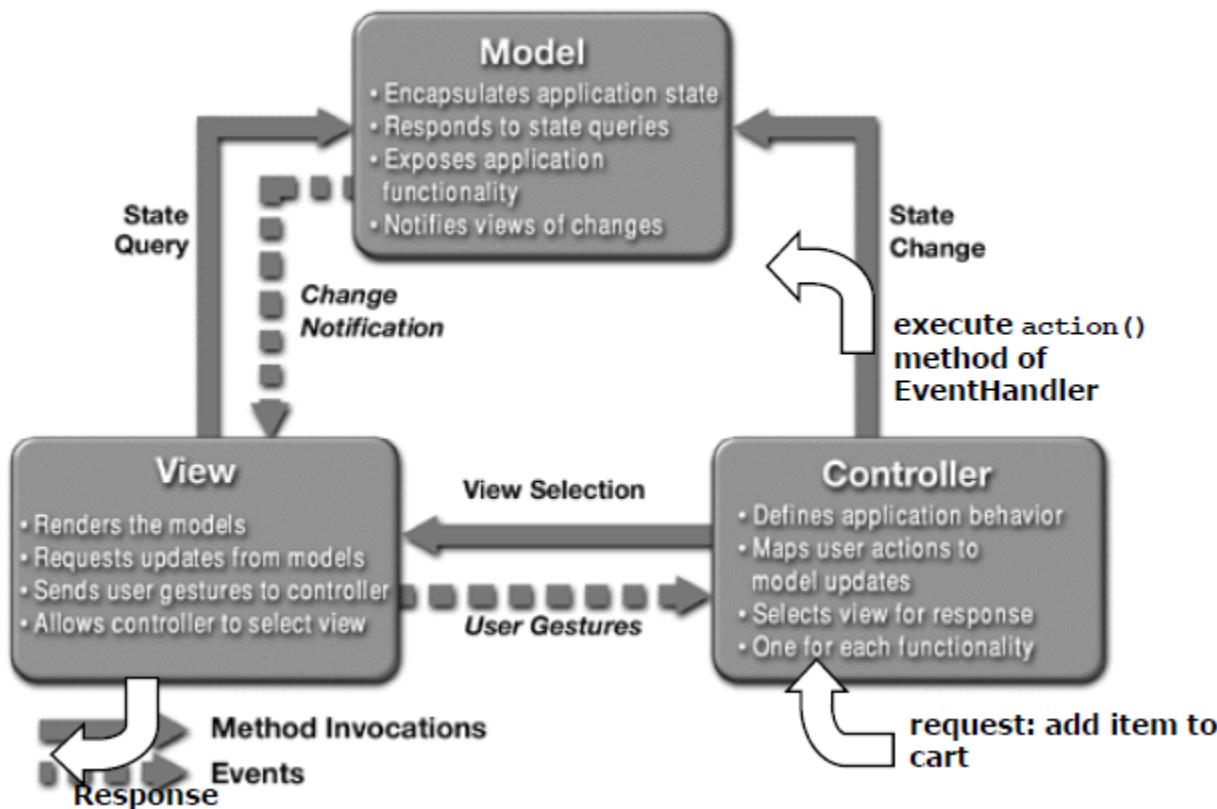
陈昊鹏

chen-hp@sjtu.edu.cn



Model-View-Controller Architectures

- Model-View-Controller (MVC) is an old design pattern for user applications from Smalltalk



- struts.apache.org

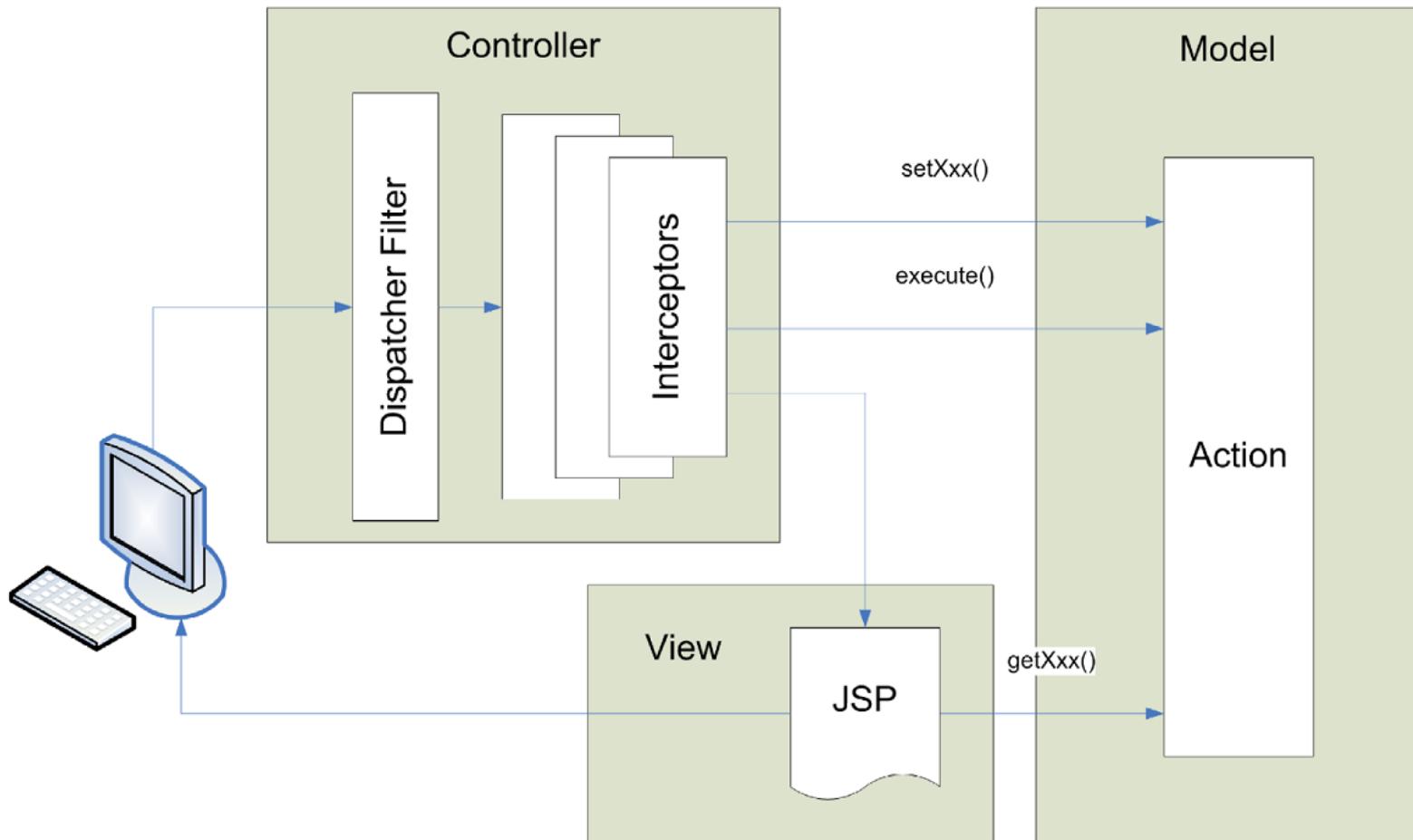


Apache Struts

Apache Struts is a free, open-source, MVC framework for creating elegant, modern Java web applications. It favors convention over configuration, is extensible using a plugin architecture, and ships with plugins to support REST, AJAX and JSON.

Fork me on GitHub

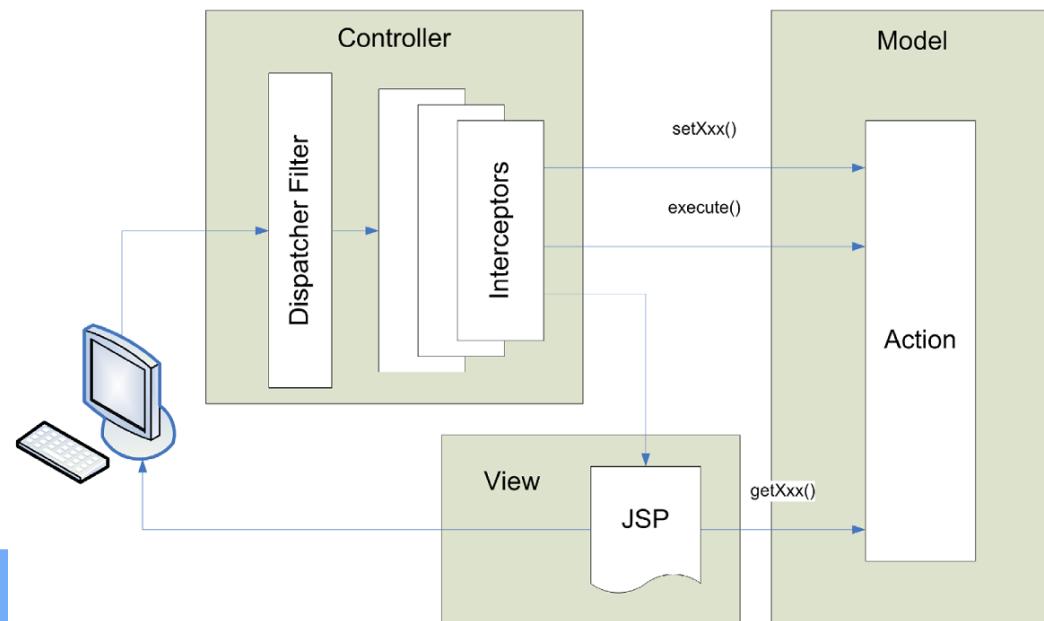
- Struts was originally developed by Craig McClanahan and donated to the Apache Foundation in May 2000.
- Struts has been a de facto framework with a strong and vibrant user community.
- Struts uses and extends the Java Servlet API to adopt the "Model 2" approach, a variation of the classic Model-View-Controller (**MVC**) design pattern.



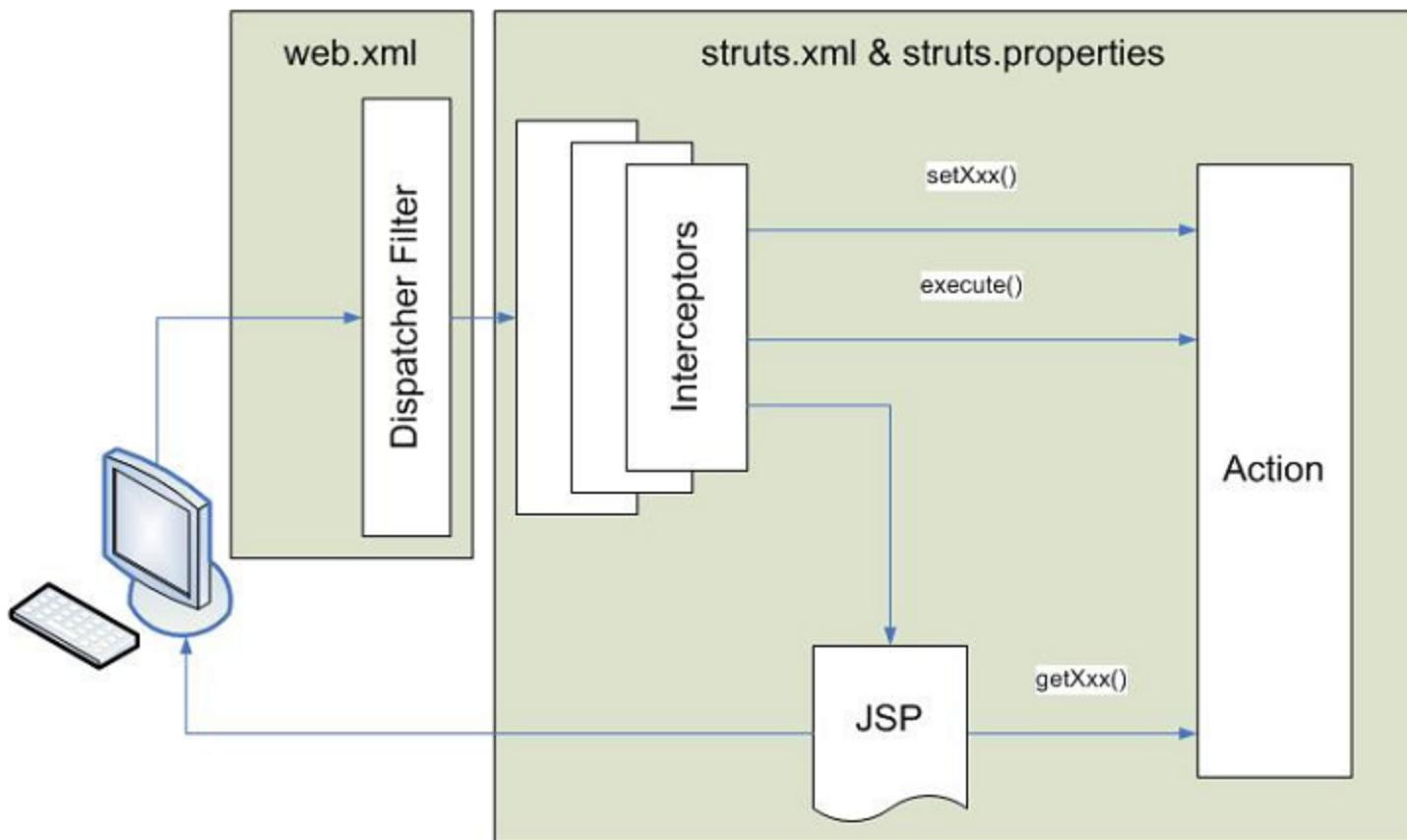
- Here are some of the features that may lead you to consider Struts2:
 - Action based MVC web framework
 - Mature with a vibrant developer and user community
 - Annotation and XML configuration options
 - POJO-based actions that are easy to test
 - Spring, SiteMesh and Tiles integration
 - OGNL expression language integration
 - Themes based tag libraries and Ajax tags
 - Multiple view options (JSP, Freemarker, Velocity and XSLT)
 - Plug-ins to extend and modify framework features

Core Components

- The Model-View-Controller pattern in Struts2 is realized with five core components
 - **Actions**: The model is implemented with actions
 - **Interceptors**: The controller is implemented with a Struts2 dispatch servlet filter as well as interceptors
 - **Value stack / OGNL**: provide common thread, linking and enabling integration between the other components.
 - **Result types**
 - **Results** / view technologies.



Configuration



Configuration

- The web application configuration for the **FilterDispatcher** servlet filter needs to be configured in your “**web.xml**” file:

```
<welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
```

```
<filter>
    <filter-name>struts2</filter-name>
    <filter-class>
        org.apache.struts2.dispatcher.filter.StrutsPrepareAndExecuteFilter
    </filter-class>
</filter>
```

```
<filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

- The `struts.xml` File

- This configuration file contains the configuration information that you will be modifying as actions are developed.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation
//DTD Struts Configuration 2.0//EN"
"http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
  <constant name="struts.devMode" value="true" />
  <package name="basicstruts2" extends="struts-default">
    <action name="index">
      <result>/index.jsp</result>
    </action>
    .....
  </package>
</struts>
```

- Actions are a fundamental concept in most web application frameworks, and they are the most basic unit of work that can be associated with a HTTP request coming from a user.
- Single result

```
class MyAction {  
    public void String execute() throws Exception {  
        return "success";  
    }  
}
```

```
<action name="my" class="com.fdar.infoq.MyAction" >  
    <result>view.jsp</result>  
</action>
```

- Multiple results

```
class MyAction {  
    public void String execute() throws Exception {  
        if( myLogicWorked() ) {  
            return "success";  
        } else {  
            return "error";  
        }  
    }  
}  
  
<action name="my" class="com.fdar.infoq.MyAction" >  
    <result>view.jsp</result>  
    <result name="error">error.jsp</result>  
</action>
```

- Interceptors are conceptually the same as **servlet filters** or the JDKs Proxy class.
 - They provide a way to supply pre-processing and post-processing around the action.

```
<interceptors>
```

```
...
```

```
<interceptor name="autowiring"  
            class="interceptor.ActionAutowiringInterceptor"/>
```

```
</interceptors>
```

```
<action name="my" class="com.fdar.infoq.MyAction" >  
    <result>view.jsp</result>  
    <interceptor-ref name="autowiring"/>  
</action>
```

- Interceptors are conceptually the same as servlet filters or the JDKs Proxy class.

- They provide a way to supply pre-processing and post-processing around the action.

```
<interceptor-stack name="basicStack">
    <interceptor-ref name="exception"/>
    <interceptor-ref name="servlet-config"/>
    <interceptor-ref name="prepare"/>
    <interceptor-ref name="checkbox"/>
    <interceptor-ref name="params"/>
    <interceptor-ref name="conversionError"/>
</interceptor-stack>
```

```
<action name="my" class="com.fdar.infoq.MyAction" >
    <result>view.jsp</result>
    <interceptor-ref name="basicStack"/>
</action>
```

- Implementing Interceptors

```
public interface Interceptor extends Serializable {  
    void destroy();  
    void init();  
    String intercept(ActionInvocation invocation)  
        throws Exception;  
}
```

Return Types



REliable, INtelligent & Scalable Systems

```
<action name="my" class="com.fdar.infoq.MyAction" >
  <result type="dispatcher">view.jsp</result>
</action>
```

```
<result-types>
  <result-type name="dispatcher" default="true"
    class="....dispatcher.ServletDispatcherResult"/>
  <result-type name="redirect"
    class="....dispatcher.ServletRedirectResult"/>
  ...
</result-types>
```

- To create a new result type, implement the Result interface.

```
public interface Result extends Serializable {
  public void execute(ActionInvocation invocation)
    throws Exception;
}
```

Hello World Using Struts 2



REliable, INtelligent & Scalable Systems

- To create a "Hello World" example, you need to do four things:
 - Create a class to store the welcome message (the model)
 - Create a server page to present the message (the view)
 - Create an Action class to control the interaction between the user, the model, and the view (the controller)
 - Create a mapping (struts.xml) to couple the Action class and view

Hello World Using Struts 2



REliable, INtelligent & Scalable Systems

- Step1: Create a class to store the welcome message (the model)

```
public class MessageStore {  
    private String message;  
  
    public MessageStore() {  
        setMessage("Hello Struts User");  
    }  
  
    public String getMessage() { return message; }  
  
    public void setMessage(String message) { this.message = message; }  
  
    public String toString() {  
        return message + " (from toString)";  
    }  
}
```

Hello World Using Struts 2



REliable, INtelligent & Scalable Systems

- Step 2 - Create The Action Class HelloWorldAction.java

```
public class HelloWorldAction extends ActionSupport {  
  
    private MessageStore messageStore;  
  
    public String execute() throws Exception {  
        messageStore = new MessageStore();  
        return SUCCESS;  
    }  
  
    public MessageStore getMessageStore() {  
        return messageStore;  
    }  
  
    public void setMessageStore(MessageStore messageStore) {  
        this.messageStore = messageStore;  
    }  
}
```

Hello World Using Struts 2



REliable, INtelligent & Scalable Systems

- Step 3 - Create The View HelloWorld.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="s" uri="/struts-tags" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Hello World!</title>
</head>
<body>
    <p>I've said hello <s:property value="helloCount" /> times!</p>
    <h2><s:property value="messageStore.message" /></h2>
    <p>I've said hello to you <s:property value="#session.helloCount" /> times!</p>
</body>
</html>
```

Hello World Using Struts 2

- Step 4 - Add The Struts Configuration In struts.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
  "-//Apache Software Foundation//DTD Struts Configuration 2.5//EN"
  "http://struts.apache.org/dtds/struts-2.5.dtd">
<struts>
  <constant name="struts.devMode" value="true" />
  <package name="basicstruts2" extends="struts-default">
    <action name="index">
      <result>/index.jsp</result>
    </action>

    <action name="hello" class="org.reins.mvc.struts.sample.action.HelloWorldAction"
      method="execute">
      <result name="success">/HelloWorld.jsp</result>
    </action>
  </package>
</struts>
```

Hello World Using Struts 2



REliable, INtelligent & Scalable Systems

- Step 5 - Create The URL Action Index.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="s" uri="/struts-tags" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Hello World!</title>
</head>
<body>
    <h2><s:property value="messageStore.message" /></h2>
</body>
</html>
```

Hello World Using Struts 2



REliable, INtelligent & Scalable Systems

- Step 6 - Create The web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
    <display-name>Struts Blank</display-name>
    <welcome-file-list>
        <welcome-file>index.jsp</welcome-file>
    </welcome-file-list>
    <filter>
        <filter-name>struts2</filter-name>
        <filter-class>
            org.apache.struts2.dispatcher.filter.StrutsPrepareAndExecuteFilter
        </filter-class>
    </filter>
    <filter-mapping>
        <filter-name>struts2</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
</web-app>
```

Hello World Using Struts 2



REliable, INtelligent & Scalable Systems

- Step 7 - Build the WAR File and Run The Application

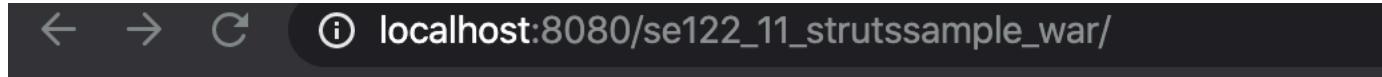
```
se122_11_strutssample ~/IdeaProjects/se122_11_strutssample
  .idea
  src
    main
      java
        org.reins.mvc.struts.sample
          action
            HelloWorldAction
            Register
          model
            MessageStore
            User
          util
            HibernateUtil
      resources
        hibernate.cfg.xml
        log4j2.xml
        struts.xml
      webapp
        WEB-INF
          web.xml
        HelloWorld.jsp
        index.jsp
        register.jsp
        thankyou.jsp
  target
```



Using Struts 2 Tags

- index.jsp

```
<%@ taglib prefix="s" uri="/struts-tags"%>
<body>
    <h1>Welcome To Struts 2!</h1>
    <p>
        <a href="
```



Welcome To Struts 2!

[Hello World](#)

[Hello Bruce Phillips](#)

Get your own personal hello by filling out and submitting this form.

Your name:

[Please register](#) for our prize drawing.

Coding Struts 2 Actions



REliable, INtelligent & Scalable Systems

- HelloWorld.java(Solution 1)

```
public class HelloWorldAction extends ActionSupport {  
  
    private String userName;  
  
    public String execute() throws Exception {  
        messageStore = new MessageStore();  
        if (userName != null) {  
            messageStore.setMessage(messageStore.getMessage() + " " + userName);  
        }  
        return SUCCESS;  
    }  
  
    public String getUserName() { return userName; }  
  
    public void setUserName(String userName) {  
        this.userName = userName;  
    }  
}
```

- **HelloWorld.java (Solution 2)**

```
public class HelloWorldAction extends ActionSupport {  
  
    public String execute() throws Exception {  
        messageStore = new MessageStore();  
  
        ActionContext context = ActionContext.getContext();  
        Map params = context.getParameters();  
        String[] users = (String[]) params.get("userName");  
  
        if (userName != null) {  
            messageStore.setMessage(messageStore.getMessage() + " " + userName);  
        }  
        return SUCCESS;  
    }  
}
```



localhost:8080/se122_11_strutssample_war/hello.action

I've said hello 2 times!

Hello Struts User Bruce Lee

I've said hello to you 2 times!

Processing Forms



REliable, INtelligent & Scalable Systems

- index.jsp

```
<html>
<body>
  <p>
    <a href=".//register.jsp">Please register</a> for our prize drawing.
  </p>
</body>
</html>
```

Processing Forms

- register.jsp

```
<?xml version="1.0" encoding="UTF-8"?>
<%@ taglib prefix="s" uri="/struts-tags"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<body>
    <h3>Register for a prize by completing this form.</h3>
    <s:form action="register">
        <s:textfield name="personBean.firstName" label="First name" />
        <s:textfield name="personBean.lastName" label="Last name" />
        <s:textfield name="personBean.email" label="Email" />
        <s:textfield name="personBean.age" label="Age" />
        <s:submit />
    </s:form>
</body>
<s:head />
</html>
```

← → C ⓘ localhost:8080/se122_11_strutssample_war/register.jsp

Register for a prize by completing this form.

First name:

Last name:

Email:

Age:

Processing Forms

- Register.java

```
public class Register extends ActionSupport {  
    private User personBean;  
  
    @Override  
    public String execute() throws Exception {  
        Session session = HibernateUtil.getSessionFactory().getCurrentSession();  
        session.beginTransaction();  
        personBean.getEmailAddresses().add(personBean.getEmail());  
        session.save(personBean);  
        session.getTransaction().commit();  
        return SUCCESS;  
    }  
  
    public User getPersonBean() { return personBean; }  
  
    public void setPersonBean(User person) {  
        personBean = person;  
    }  
}
```

Processing Forms

- User.java

```
@Entity
@Table(name = "user")
public class User
{
    private Long id;
    private String firstName;
    private String lastName;
    private String email;
    private int age;
    private Set<String> emailAddresses = new HashSet();

    public User() { }

    @Id
    @GeneratedValue(generator = "increment")
    @GenericGenerator(name = "increment", strategy = "increment")
    public Long getId() { return id; }
    private void setId(Long id) { this.id = id; }
```

Processing Forms

- User.java

```
public String getFirstName() { return firstName; }
public void setFirstName(String firstName) { this.firstName = firstName; }
public String getLastName() { return lastName; }
public void setLastName(String lastName) { this.lastName = lastName; }
public String getEmail() { return email; }
public void setEmail(String email) { this.email = email; }
public int getAge() { return age; }
public void setAge( int age) { this.age = age; }
```

```
@ElementCollection
@CollectionTable( name="emailaddress",
    joinColumns = { @JoinColumn(name = "id", referencedColumnName = "id")})
@Column(name="emailaddress")
public Set<String> getEmailAddresses() { return emailAddresses; }
public void setEmailAddresses(Set emailAddresses) {
    this.emailAddresses = emailAddresses;
}
```

Processing Forms



REliable, INtelligent & Scalable Systems

- struts.xml

```
<struts>
```

```
  <action name="register" class="org.reins.mvc.struts.sample.action.Register"  
    method="execute">  
    <result name="success">/thankyou.jsp</result>  
    <result name="input">/register.jsp</result>  
  </action>
```

```
</struts>
```

Processing Forms



REliable, INtelligent & Scalable Systems

- thankyou.jsp

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<body>
```

```
    <h3>Thank you for registering for a prize.</h3>
```

```
    <p>
```

```
        Your registration information:
```

```
        <s:property value="personBean" />
```

```
    </p>
```

```
    <p>
```

```
        <a href="<s:url action='index' />">Return to home page</a>.
```

```
    </p>
```

```
</body>
```

```
</html>
```

Processing Forms



REliable, INtelligent & Scalable Systems

← → ⌛

ⓘ localhost:8080/se122_11_strutssample_war/register.action

Thank you for registering for a prize.

Your registration information: First Name: Yi Last Name: Sima Email: royal@wei.com Age: 70

[Return to home page.](#)

Form Validation

- Constraints
 - User must provide a first name
 - User must provide an email address
 - User younger than 18 cannot register
- Register.java

```
public void validate(){
    if ( personBean.getFirstName().length() == 0 ){
        addFieldError( "personBean.firstName",
                      "First name is required." );
    }
    if ( personBean.getEmail().length() == 0 ){
        addFieldError( "personBean.email",
                      "Email is required." );
    }
    if ( personBean.getAge() < 18 ){
        addFieldError( "personBean.age",
                      "Age is required and must be 18 or older" );
    }
}
```

- If any errors have been added then Struts 2 will not proceed to call the execute method.
 - Rather the Struts 2 framework will return "**input**" as the result of calling the action.



localhost:8080/se122_11_strutssample_war/register.action

Register for a prize by completing this form.

First name is required.

First name:

Last name:

Email is required.

Email:

Age is required and must be 18 or older

Age:

- Your Struts 2 application may need to access the HTTP session object.
 - Struts 2 provides an interface, **SessionAware**, that your Action class should implement to obtain a reference to the HTTP session object.
 - The **SessionAware** interface has one method, **setSession**, that your Action class will need to override.
- `HelloWorldAction.java`

```
private Map<String, Object> userSession;

public void setSession(Map<String, Object> session) {
    userSession = session;
}
```

- HelloWorldAction.java

```
public String execute() throws Exception {  
    increaseHelloCount();  
}  
  
private void increaseHelloCount() {  
    Integer helloCount = (Integer) userSession.get("helloCount");  
    if (helloCount == null) {  
        helloCount = 1;  
    } else {  
        helloCount++;  
    }  
    userSession.put("helloCount", helloCount);  
}
```

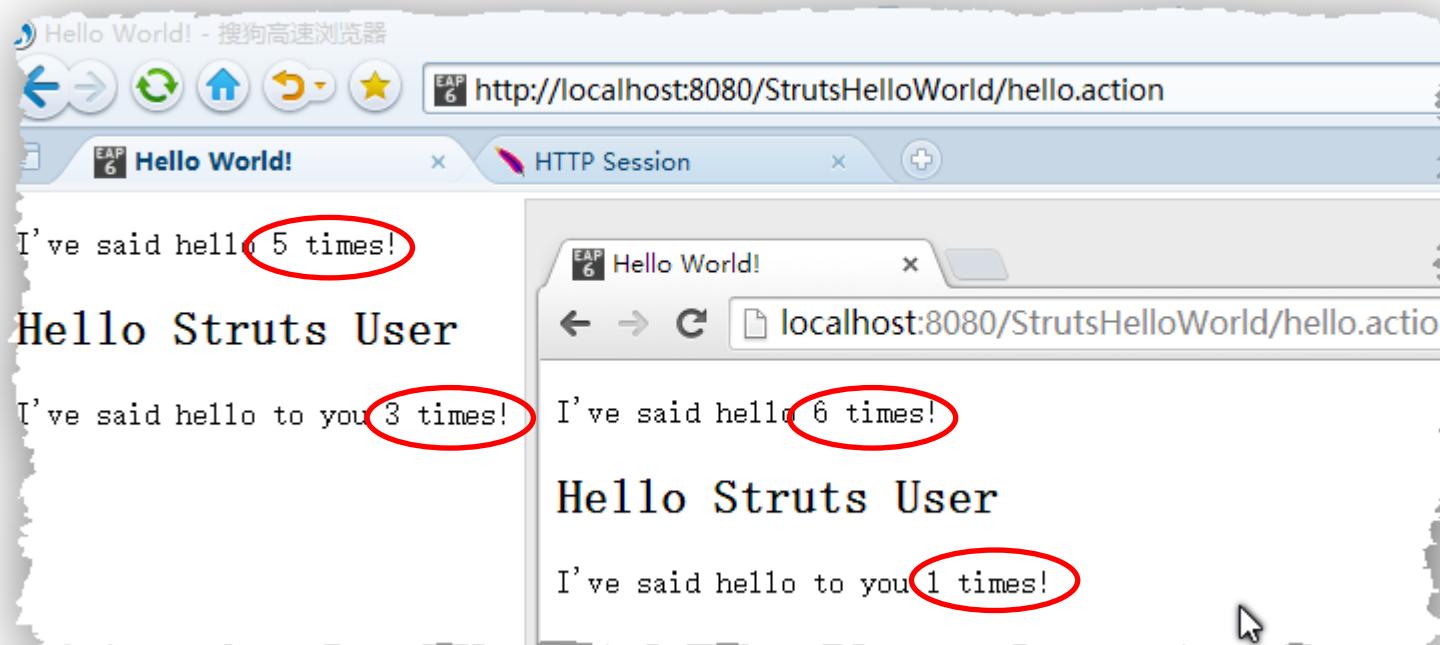
Http Session

- HelloWorld.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Hello World!</title>
</head>
<body>
    <p>I've said hello <s:property value="helloCount" /> times!</p>
    <h2><s:property value="messageStore.message" /></h2>
    <p>I've said hello to you <s:property value="#session.helloCount" /> times!</p>
</body>
</html>
```

Http Session



- Apache Struts 2 Documentation: Getting Started,
 - <https://struts.apache.org/getting-started/>



- *Web*开发技术
- *Web Application Development*

Thank You!