



Introduction to Struts

Sylvain Gibassier

Sylvain536(at)hotmail(dot)com

Disclaimer

- ◆ Views expressed in this presentation are my own and do not reflect those of my employer

Learning objective

- ◆ Understanding struts model

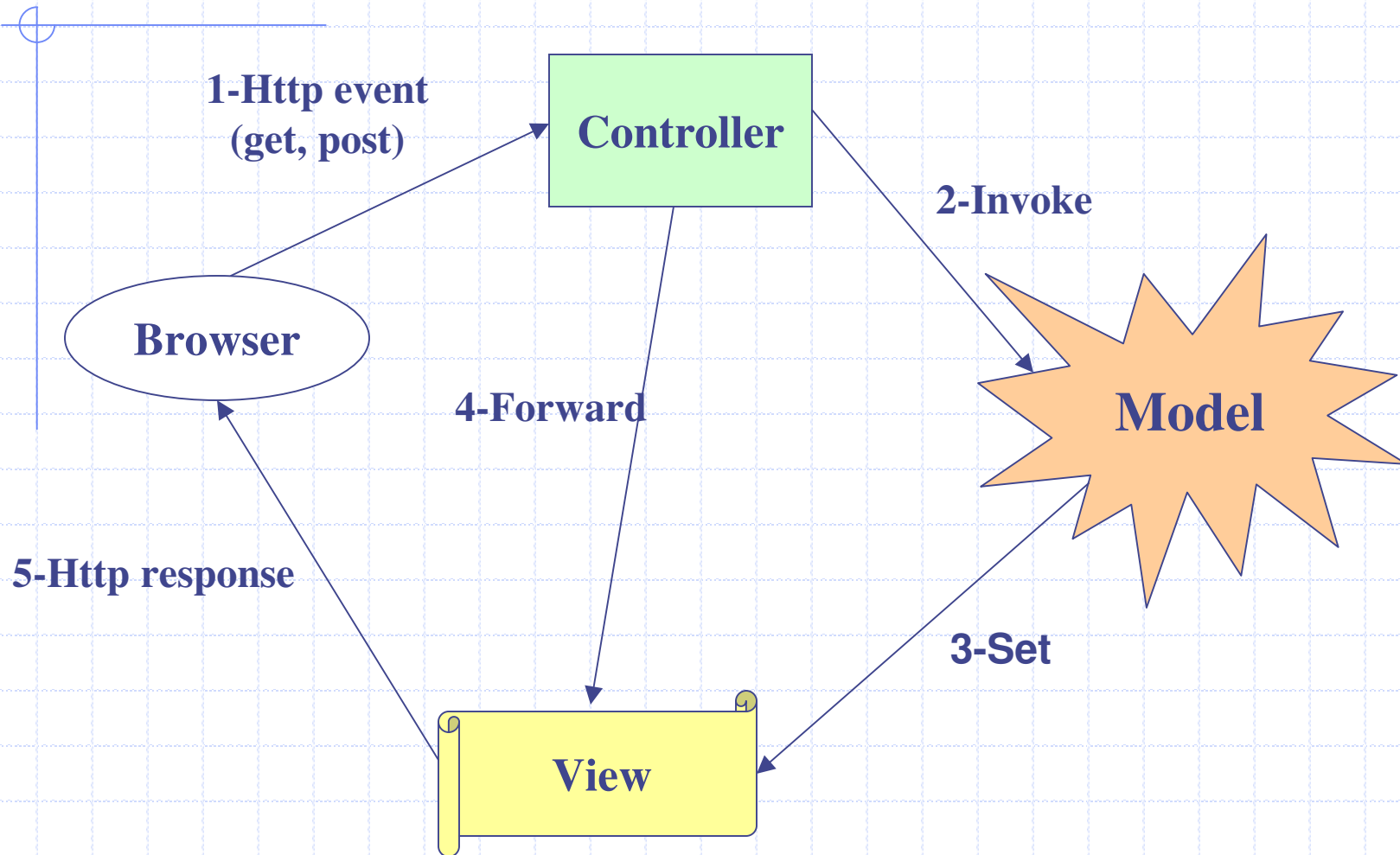
Agenda

- ◆ What is struts?
- ◆ How to install struts in your webapp?
- ◆ Controller, Model
- ◆ View State
- ◆ View

What is Struts?

- ◆ Open source
- ◆ Apache Jakarta project
- ◆ <http://jakarta.apache.org/struts/>
- ◆ Current version: 1.1
- ◆ Framework for developing java web based applications
- ◆ Based on the JSP Model 2 architecture (MVC)

The JSP Model 2



Agenda

- ◆ What is struts?
- ◆ How to install struts in your webapp?
- ◆ Controller, Model
- ◆ View State
- ◆ View

Struts installation

1. Download the zip file from the Apache Jakarta web site
2. Copy the jar files from the lib directory of the zip file in WEB-INF/lib
3. Copy the .tld files from the lib directory of the zip file in the WEB-INF
4. Editing web.xml
 - Struts servlet configuration and mapping
5. Creating an empty struts-config.xml
6. Start your server
7. Test

Editing web.xml

```
<servlet>
  <servlet-name>action</servlet-name>
  <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>0</param-value>
  </init-param>
  <init-param>
    <param-name>detail</param-name>
    <param-value>0</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>action</servlet-name>
  <url-pattern>/controller/*</url-pattern>
</servlet-mapping>
```

Creating an empty struts-config.xml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

```
<!DOCTYPE struts-config PUBLIC "-//Apache Software  
Foundation//DTD Struts Configuration 1.1//EN"  
"http://jakarta.apache.org/struts/dtds/struts-  
config_1_1.dtd">
```

```
<struts-config>  
  <form-beans>  
  </form-beans>  
  
  <action-mappings>  
  </action-mappings>  
</struts-config>
```

Start your server and test



Message from the struts controller meaning that it does not know anything about that http event

Agenda

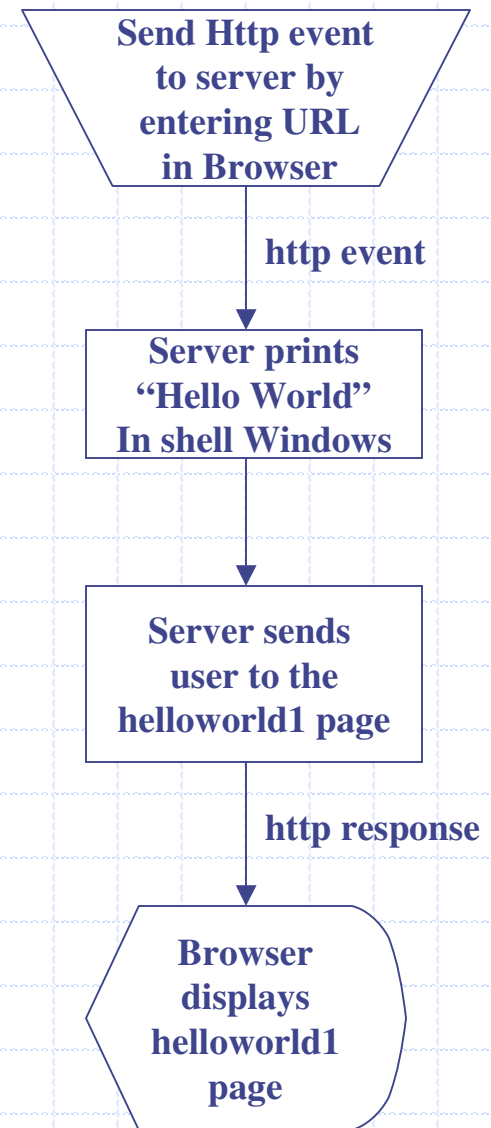
- ◆ What is struts?
- ◆ How to install struts in your webapp?
- ◆ Controller, Model
- ◆ View State
- ◆ View

Hello World 1



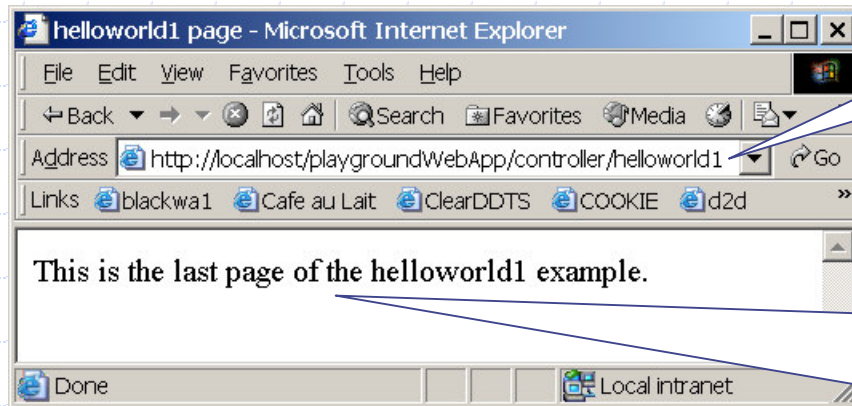
Application requirements:

- Invoke the business logic that is going to write "Hello World!" in the shell windows you have started your server in.
- Show the user a page with "Hello World!" in his browser.



Demo

1



This creates the http event the controller is expecting

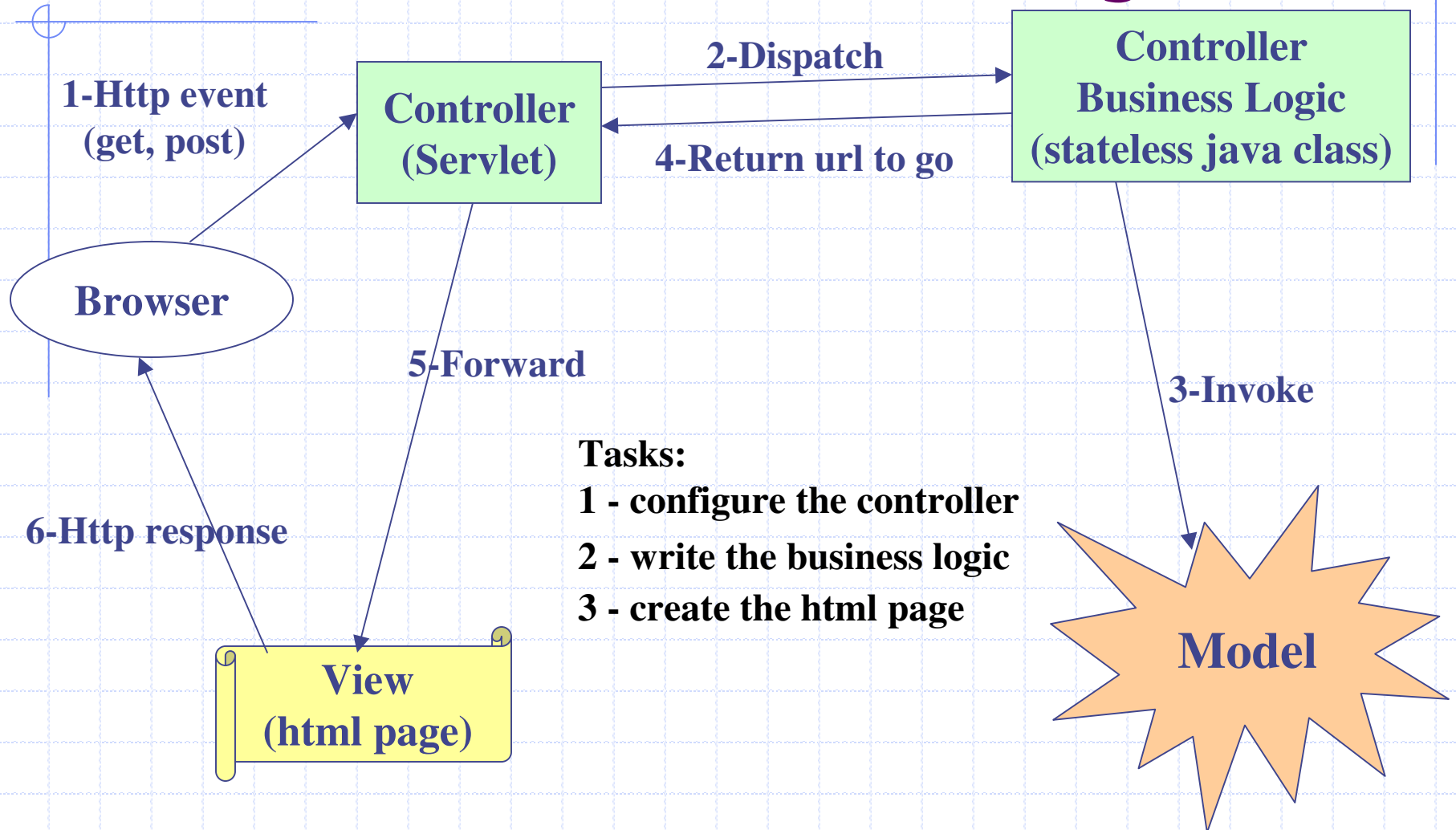
The struts controller has forwarded the user to helloworld1.html after executing the business logic

2



**Congratulations!
Our "business logic"
has been successfully
executed!**

Helloworld 1 interaction diagram



Introduction to the struts controller

- ◆ Class: `org.apache.struts.action.ActionServlet`
- ◆ Is the traffic cop of your web application: determines which action to call for the http event (get/post) it is getting based on its configuration
- ◆ Configured by: `struts-config.xml`

Step 1: Configuration of the struts controller

For the “Hello World!” application:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>  
<!DOCTYPE struts-config PUBLIC "-//Apache Software  
Foundation//DTD Struts Configuration 1.1//EN"  
"http://jakarta.apache.org/struts/dtds/struts-config_1_1.dtd">
```

```
<struts-config>  
  <form-beans/>
```

```
  <action-mappings>  
    <action path="/helloworld1" type="presentation.HelloWorld1Action"/>  
  </action-mappings>  
</struts-config>
```

Http event

Business logic to call when receiving
the “helloworld” http event

Struts business logic introduction

- ◆ Stateless java class that must extend:
 - `org.apache.struts.action.Action`
- ◆ Must be thread safe
- ◆ Override the `execute(...)` method to call the business logic
- ◆ Tell struts which page to display next

Step 2: writing the business logic to invoke the model

```
public class HelloWorld1Action extends Action {

    public ActionForward execute(ActionMapping arg0,
        ActionForm arg1, HttpServletRequest arg2,
        HttpServletResponse arg3) throws Exception {

        // start business logic
        System.out.println("Hello World!");
        // end business logic

        // tell which page to go next
        return new ActionForward("/helloworld1.html");
    }
}
```

Step 3: create helloworld1.html

◆ Cannot be more simple ;-):

```
<html>
```

```
<head>
```

```
<title>helloworld1 page</title>
```

```
</head>
```

```
<body>
```

```
This is the last page of the  
helloworld1 example.
```

```
</body>
```

```
</html>
```

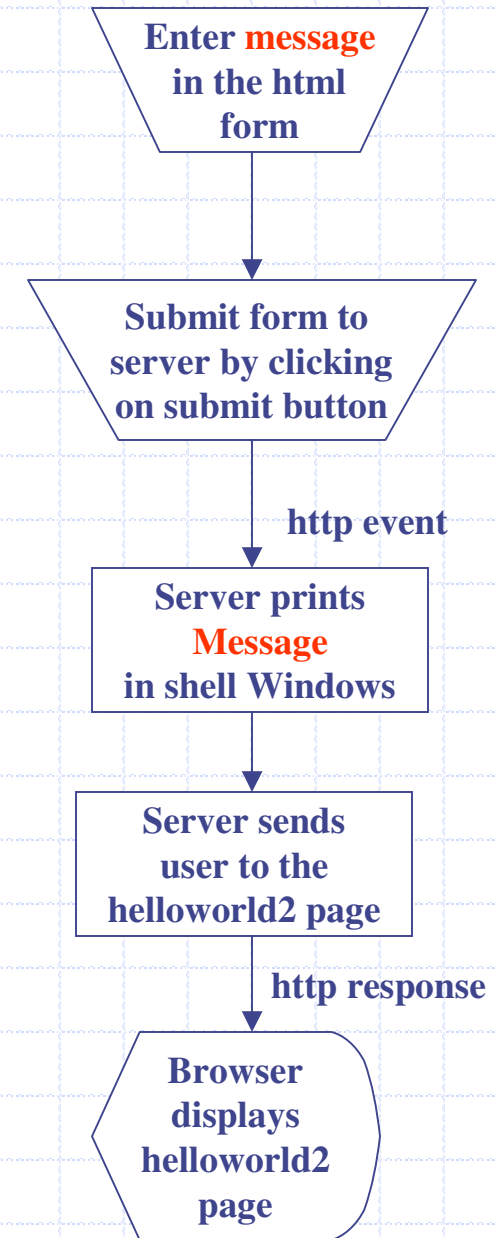
Agenda

- ◆ What is struts?
- ◆ How to install struts in your webapp?
- ◆ Controller, Model
- ◆ View State
- ◆ View

"Hello World!" 2

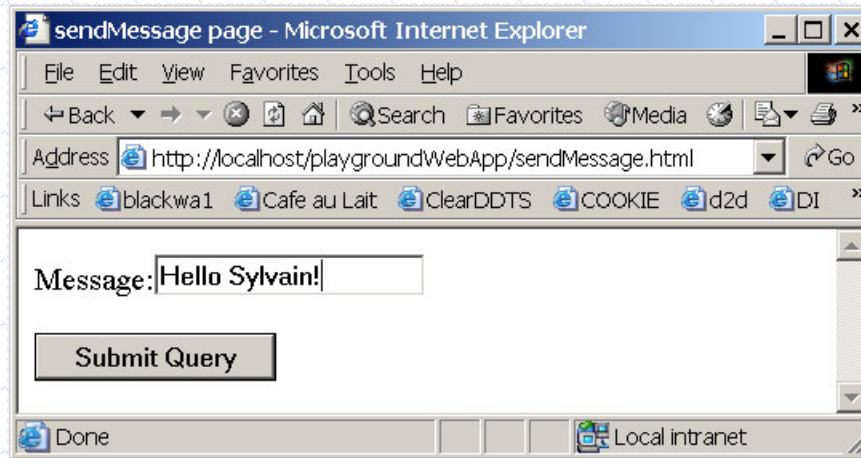
◆ Addition to previous requirements:

- Instead of displaying "Hello World!" in the shell windows, your application is now going to show a custom message that will be entered in a html form.

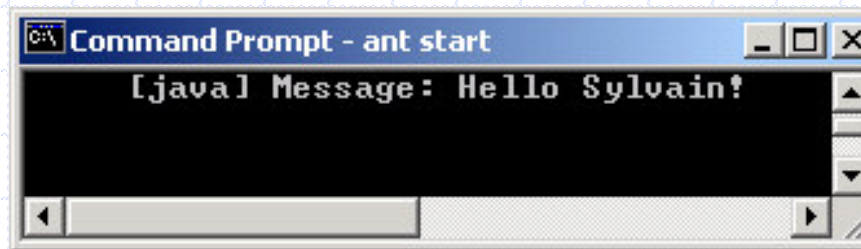


Demo

1



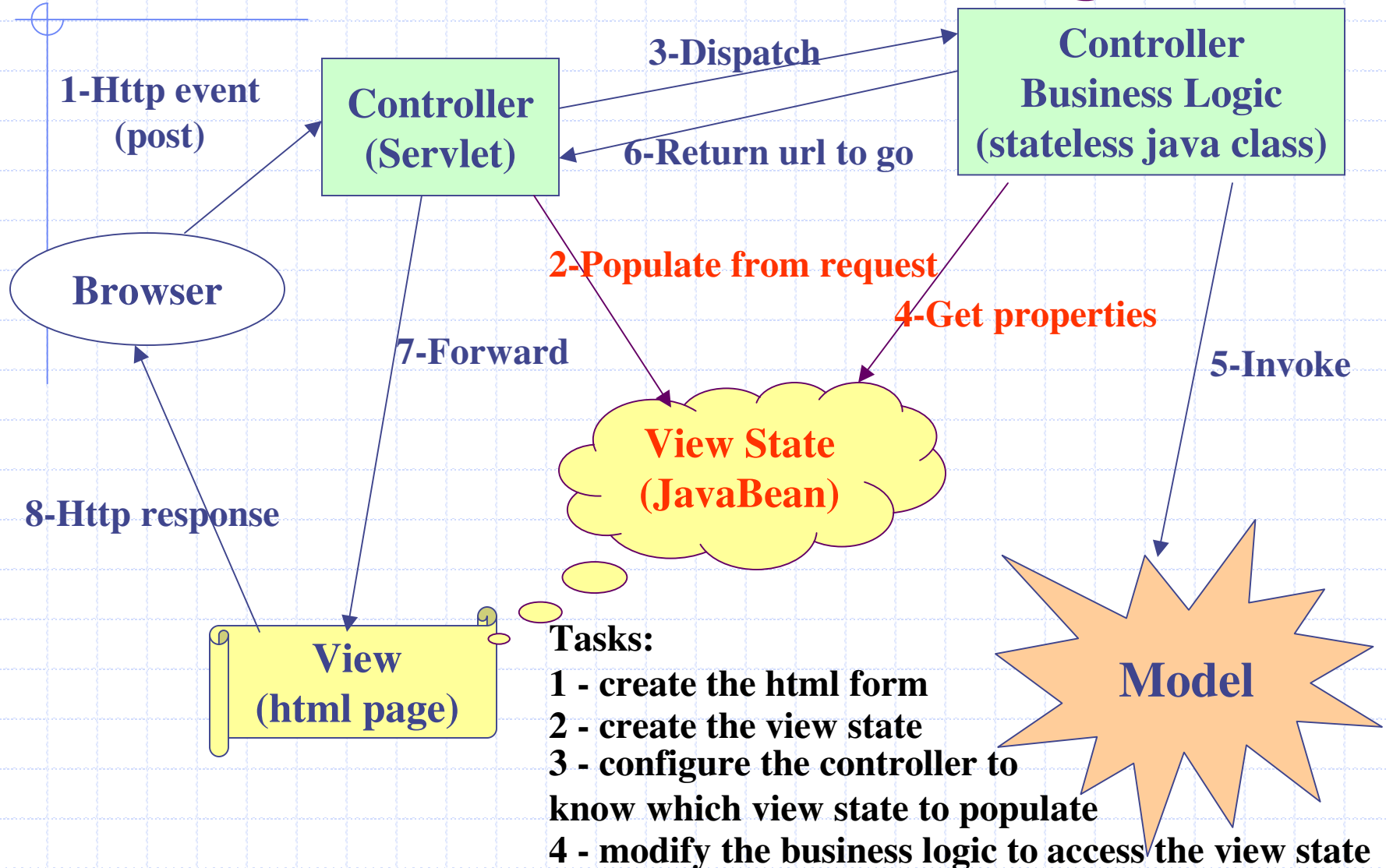
2



3



Helloworld 2 interaction diagram



Step 1: creating the html form

sendMessage.html:

```
<html>
<head>
<title>sendMessage page</title>
</head>
<body>
<form method="post" action="/controller/helloworld2">
  Message:<input type="text" name="message"/><BR><BR>
  <input type="submit"/>
</form>
</body>
</html>
```

Http event that will be sent to the controller when the submit button is pressed

Introduction to the struts view state

- ◆ Is a javabean, consequences:
 - Zero argument constructor
 - Public getter and setter
- ◆ Must extend `org.apache.struts.action.ActionForm`
- ◆ Properties should be String
- ◆ Is **NOT** an entity EJB, Data Transfer Object, or Value Object

Step 2: creating the view state

javabean

```
public class HelloWorld2Form extends
    ActionForm {
    private String message;

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }
}
```

Step 3: associate the form with its business logic

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<!DOCTYPE struts-config PUBLIC "-//Apache Software Foundation//DTD
Struts Configuration 1.1//EN"
"http://jakarta.apache.org/struts/dtds/struts-config_1_1.dtd">

<struts-config>

  <form-beans>
    <form-bean name="helloWorld2Form"
type="presentation.HelloWorld2Form"/>
  </form-beans>

  <action-mappings>
    <action path="/helloworld2" scope="request"
name="helloWorld2Form" type="presentation.HelloWorld2Action"/>
  </action-mappings>

</struts-config>
```



A red arrow points from the `name="helloWorld2Form"` attribute in the `<action>` tag to the `name="helloWorld2Form"` attribute in the `<form-bean>` tag, indicating the association between the form and its business logic.

Step 4: modifying the business logic to get the message from the view state

```
public ActionForward execute(ActionMapping arg0,  
    ActionForm actionForm, HttpServletRequest arg2,  
    HttpServletResponse arg3)  
throws Exception {  
    // get the property out of the view state  
    HelloWorld2Form helloWorld2Form =  
        (HelloWorld2Form) actionForm;  
    String message = helloWorld2Form.getMessage();  
  
    System.out.println("Message: " + message);  
  
    return new ActionForward("/helloworld2.html");  
}
```

Agenda

- ◆ What is struts?
- ◆ How to install struts in your webapp?
- ◆ Controller, Model
- ◆ View State
- ◆ View

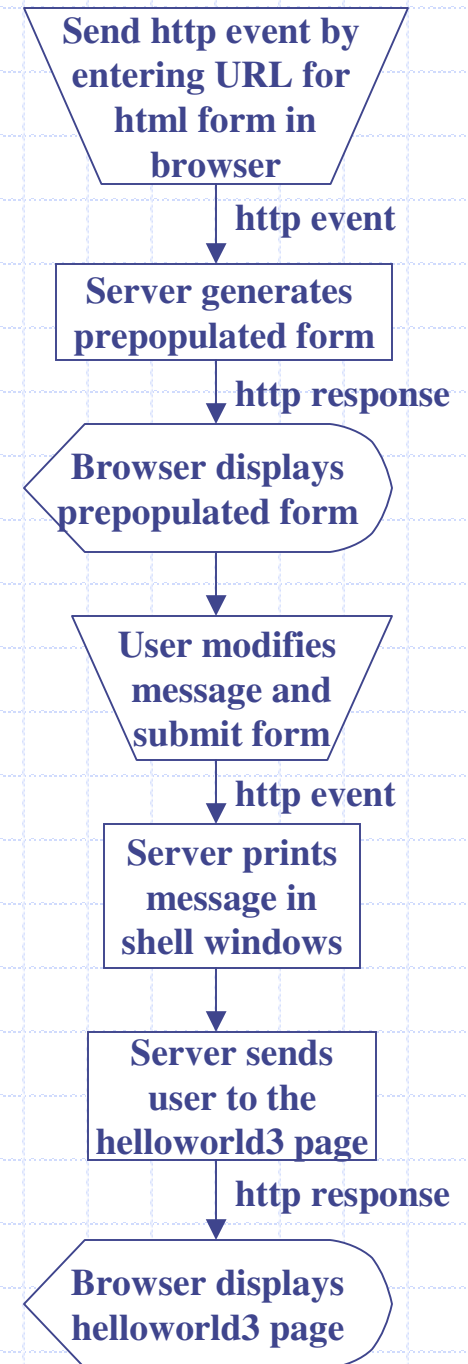
"Hello World!" 3

◆ Addition to the previous requirement:

- The message field needs to be prepopulated with the current time in milliseconds

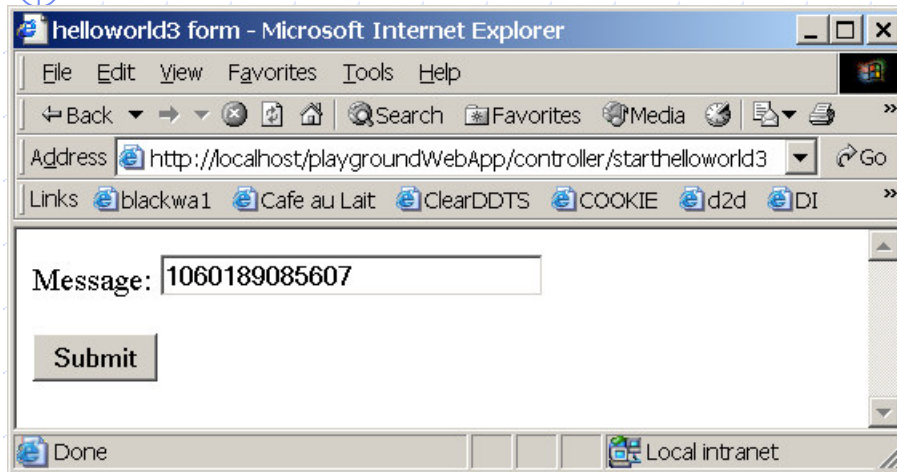
Tasks:

- 1 – Modify the html form to be able to be prepopulated
- 2 – Write the business logic to prepopulate the form
- 3 – Modify controller configuration to handle form prepopulation

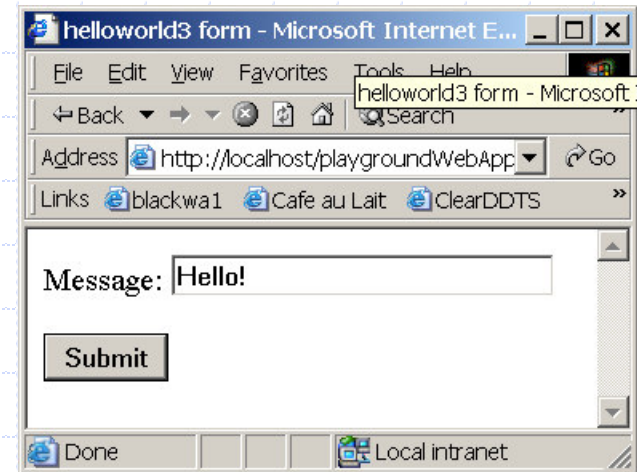


Demo

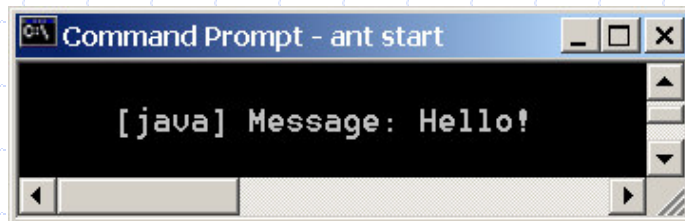
1



2



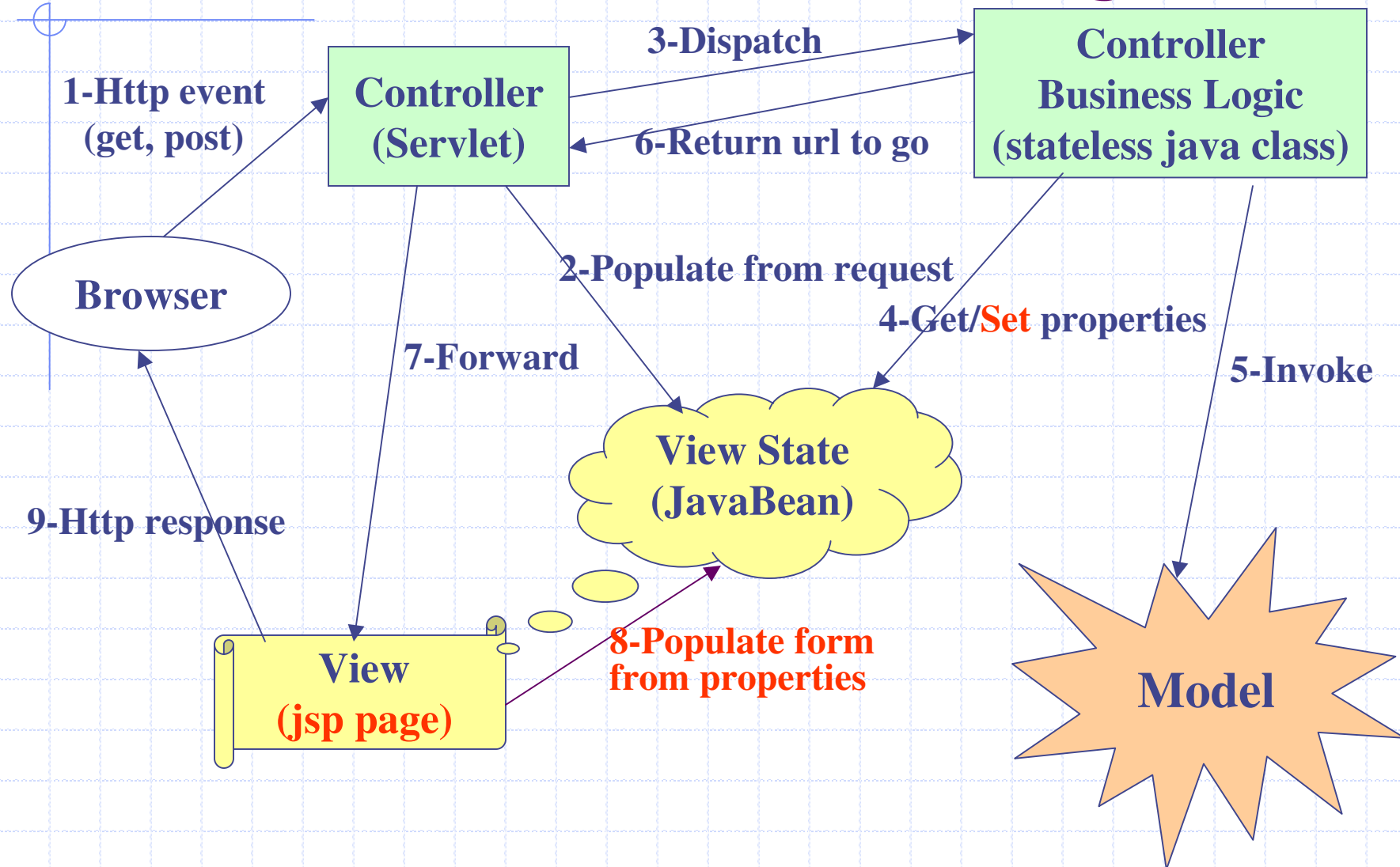
3



4



Helloworld 3 interaction diagram



Introduction to the struts view

- ◆ JSP pages

- ◆ Uses Struts or JSTL tags to populate the form. They get the properties from the view state javabean that has been put in the request by the controller

Step 1: creating the view

```
<%@ taglib uri='/WEB-INF/struts-html.tld' prefix='html' %>
```

```
<html>
```

```
<head>
```

```
<title>helloworld3 form</title>
```

```
</head>
```

```
<body>
```

```
<html:form action="/endhelloworld3">
```

```
  Message: <html:text property="message" size="30"/>
```

```
  <BR><BR>
```

```
  <html:submit/>
```

```
</html:form>
```

```
</body>
```

```
</html>
```

Step 2: Business logic to prepopulate the form

```
public ActionForward execute(ActionMapping  
    actionMapping, ActionForm  
    actionForm, HttpServletRequest  
    arg2, HttpServletResponse arg3)  
throws Exception {  
    HelloWorld3Form helloWorld3Form = (HelloWorld3Form)  
        actionForm;  
    helloWorld3Form.setMessage(String.valueOf(  
        System.currentTimeMillis()));  
return actionMapping.findForward("continue");  
}
```

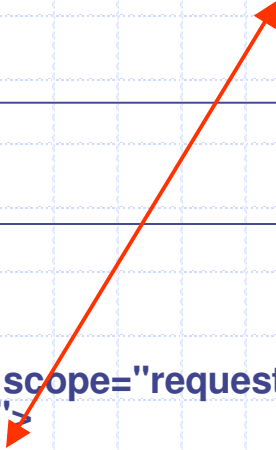
Step 3: Controller configuration

Extract from the HelloWorld3Action.java (business logic):

```
public ActionForward execute(ActionMapping  
    actionMapping, ActionForm actionForm, HttpServletRequest  
    arg2, HttpServletResponse arg3)  
    throws Exception {  
    [...]  
    return actionMapping.findForward("continue");  
}
```

Extract from struts-config.xml:

```
<action-mappings>  
    [...]  
    <action path="/starthelloworld3" scope="request" name="helloWorld3Form"  
type="presentation.StartHelloWorld3Action">  
        <forward name="continue" path="/helloworldForm3.jsp"/>  
    </action>  
</action-mappings>
```



Issues

- ◆ The user can directly go to the jsp page
 - Solution: put jsp under /WEB-INF/ (feature not supported by every container)
- ◆ The user can directly go to the endhelloworld3 action
 - Solution: use the saveToken/isTokenValid feature

Conclusion

- ◆ Very nice framework:
 - Can be used very simply
 - But can also be very powerful (lots of advanced features)
- ◆ Very robust, and real world tested
- ◆ Development bottleneck: struts-config.xml
 - Solution: multiple struts-config.xml configuration

Books

