Introduction to Struts2

- What is Struts2?
- Basic Request Flow
- Configuration

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Introducing Struts2

- Struts2 is a modern, model-view-controller (MVC) Web framework for Java
- Struts2 is based on the well regarded OpenSymphony WebWork framework and borrows ideas from the Struts1 project



Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Struts2 Highlights

- Mapping HTTP requests to Java methods (GoF **Command** pattern)
- Navigation or flow between pages
- JSP components that make it easy to create HTML forms that participate in Struts2
- Input validation, including JavaScript
- The **Object Graph Notation Language** (OGNL), which is an expression language more powerful than standard JSP EL

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

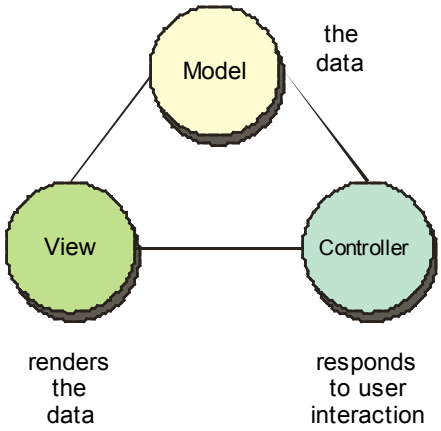
Struts2 vs Struts1

- Both Struts1 and Struts2 are action-based frameworks, but there are several differences:
    - Struts2 is still under active development, while Struts1 is in "maintenance mode"
    - Struts2 requires less configuration in XML, especially if you use **annotations**
    - Struts1 uses a servlet for the MVC controller, while Struts2 uses a servlet **filter**
    - Struts2 allows more use of POJOs, which simplifies re-use and testing

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

What is MVC?

- MVC originated in the 1980s in Smalltalk as a technique to help build graphical user interfaces (GUIs) – Java's Swing GUI toolkit uses a version of MVC
- MVC is effective since it decomposes applications into components that you can develop, debug and modify separately

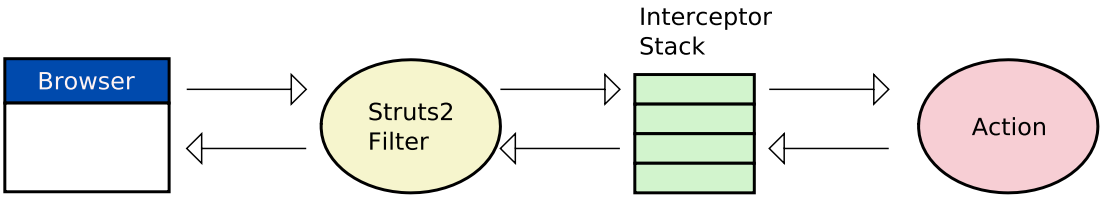


The Model-View-Controller architecture (MVC) splits an application into three objects. The model component stores the application's data -- the model may connect to a database. The view component reads the data from the model and presents it to the user. There may be multiple views on the same model. The controller responds to requests from the user and updates the model as required. If the model has changed, the view needs to be regenerated.

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Basic Request Flow

- Struts2 JSPs can contain **custom actions** that generate requests that pass through the Struts2 filter
- The filter passes requests through the **interceptor stack**, which can pre-and-post process requests
- Developers write **actions** that respond to the request and return a **result** that tells the filter where to navigate



In Struts2, JSPs act as the MVC "view", while the Struts2 filter plays the role of "controller". The "model" role is played by actions and their associated domain objects.

Interceptors are objects that pre-or-post process the request. Struts2 uses interceptors heavily itself to implement many features, including form processing. You can also write your own custom interceptors.

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

What is an Action?

- A Struts2 **action** is a Java object that responds to HTTP requests, usually invoking business logic and then returning a **result** that tells Struts2 what to do next
- An action class may, but is not required to, extend a Struts2 class or implement a Struts2 interface, but must provide an **execute** method

```
1    public class HelloAction
2    {
3        public String execute()
4        {
5            return "myresult";
6        }
7
8        public String getMyProp() { return "Hello"; }
9    }
```

There are some advantages to implementing Struts2 interfaces, as we will cover later, but note that Struts2 supports pure POJO actions that are easy to test.

Note that this action exposes a property named "myprop" with the value "Hello". Actions are not required to define properties, but it's quite common.

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Action Results

- Actions can return **result strings** that tell the Struts2 runtime what should happen next
- By default, Struts2 uses result strings to navigate to a view, typically a JSP
- You can map navigation result strings to URLs, either in the struts.xml file or via annotations

```
1    struts.xml
2    ----------
3
4    <action name="Hello" class="hello.HelloAction">
5      <result name="myresult">/mygreat.jsp</result>
6    </action>
```

On the last page, we showed an action that returned a String "myresult". Here we show mapping that string to "/mygreat.jsp" - Struts2 will forward to this JSP when the action returns.

Note that if an action returns "success", the configuration is a bit easier, since Struts2 recognizes it as a default:

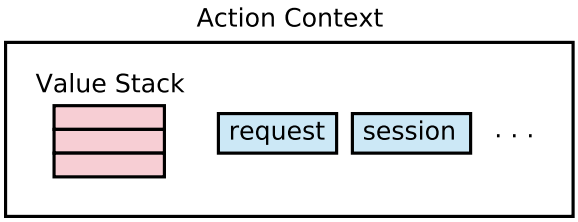
```
<result>/someother.jsp</>
```

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Struts2 Tag Libraries

- Struts2 provides **tag libraries** so that JSPs can act as views in a Struts2 application
- Tags can use OGNL to reference properties in objects within the **action context**

```
1    <%@ taglib prefix="s" uri="/struts-tags" %>
2
3    <html>
4      <body>
5        <p>
6          Hello there, <s:property value="myprop"/>!
7        </p>
8      </body>
9    </html>
```



Struts2 automatically places the action object on the "value stack", so that OGNL expressions in JSPs can reference properties exposed by the action.

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Configuring the Struts2 Filter

- The Struts2 filter acts as a **front controller** and routes requests to actions
- You must configure the Struts2 filter in the web.xml deployment descriptor

```
1    web.xml
2    -------
3    . . .
4
5      <filter>
6        <filter-name>struts2</filter-name>
7        <filter-class>
8           org.apache.struts2.dispatcher.FilterDispatcher
9        </filter-class>
10     </filter>
11     <filter-mapping>
12       <filter-name>struts2</filter-name>
13       <url-pattern>/*</url-pattern>
14     </filter-mapping>
15   . . .
```

It's customary to write a URL pattern that configures the filter to "see" all requests to the Web application so that the filter can do its work.

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Configuration: Struts.xml

- One technique to configure actions and results is to write entries in the struts.xml file
- This file, if present, must be in the application's **CLASSPATH**

```
<?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>

. . .

</struts>
```


Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Configuration: Annotations

- Another technique to configure actions and navigation strings is to use Java5 **annotations**
- Struts2 uses annotations and **configuration by convention** to determine how to map actions and navigation strings

```
1    @Results(
2    {
3      @Result(name="myresult",
4        location="/mygreat.jsp")
5    })
6    public class HelloAction
7    {
8      . . .
9    }
```

Using annotations reduces, or even eliminates the need to write a Struts2 configuration XML file.

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Chapter Summary

In this chapter, you learned:

- The fundamentals of the Struts2 framework
- How to write and configure a simple Struts2 application

Copyright © Descriptor Systems, 2001-2014. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum