JSF Custom Renderers

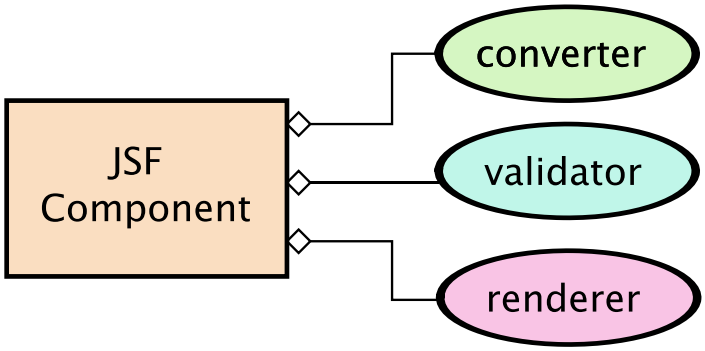
- What is a Renderer?
- Writing a Custom Renderer

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

JSF Components are Modular

- Most JSF components use the **delegating implementation** programming model in which the component delegates functionality like conversion, validation and rendering to separate objects
- This approach allows developers to substitute renderers for the standard components without needing to completely rewrite the component



Since renderers are pluggable, it's easy to change the HTML that standard components generate. For example, you could ensure that the generated HTML is compliant with Section 508 of the US Rehabilitation Act for accessibility for impaired vision users.

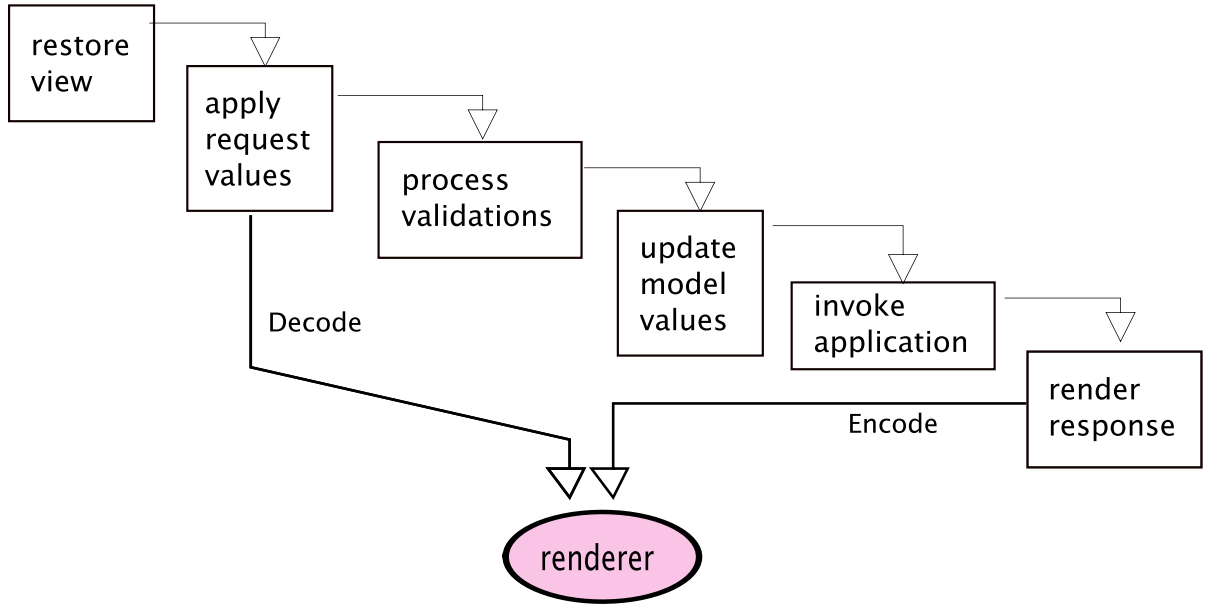
Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

What is a Renderer?

- Renderers are responsible for two jobs:
    - **Decoding**: Retrieving user input from an HTTP request for a component
    - **Encoding**: Generating a component's output
- A custom converter typically extends an existing converter and overrides methods to modify standard function

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

When Does JSF Call Renderers?



Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

What is a Render Kit?

- A **render kit** is a library of renderers
- All JSF implementations must provide a standard render kit
- You can create complete custom render kits or individually replace renders from the standard render kit

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Renderers vs Converters

- A converter's job is convert textual data to/from Java types - to Java types during **Apply Request Values** and to text during **Render Response**
- A renderer's job is to extract request parameters and to render to the actual output stream

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

The Renderer Interface

javax.faces.render.Renderer
convertClientId(ctx:FacesContext, id:String):String decode(ctx:FacesContext, comp:UIComponent):void encodeBegin(ctx:FacesContext, comp:UIComponent):void encodeChildren(ctx:FacesContext, comp:UIComponent):void encodeEnd(ctx:FacesContext, com:UIComponent):void getConvertedValue(ctx:FacesContext, comp:UIComponent, submittedVal:Object):Object getRendersChildren():boolean

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Steps for Writing a Renderer

- Create the renderer subclass, overriding methods as necessary
- Register the renderer in faces-config.xml

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

A Simple Renderer Class

```
1    package renderers;
2
3    import java.io.IOException;
4    import javax.faces.component.UIComponent;
5    import javax.faces.context.*;
6    import com.sun.faces.renderkit.html_basic.LabelRenderer;
7
8    public class MyRenderer extends LabelRenderer
9    {
10       public void encodeBegin(FacesContext ctx,
11                UIComponent comp) throws IOException
12       {
13           ResponseWriter writer = ctx.getResponseWriter();
14           writer.write("* - ");
15           super.encodeBegin(ctx, comp);
16       }
17   }
```


Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Configuring a Renderer

```
1    <faces-config . . .>
2
3      <render-kit>
4        <renderer>
5          <component-family>
6            javax.faces.Output
7          </component-family>
8          <renderer-type>
9            javax.faces.Text
10         </renderer-type>
11         <renderer-class>
12           renderers.MyRenderer
13         </renderer-class>
14       </renderer>
15     </render-kit>
16
17   </faces-config>
```


Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

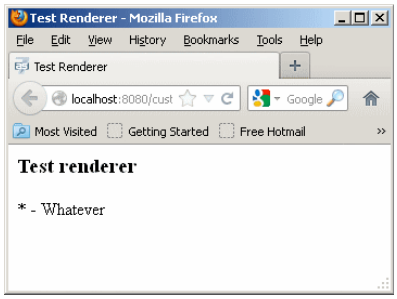
More on Configuring

- To replace a renderer in the standard render kit, you need to specify the **component-family** and **renderer-type** in faces-config.xml
- You can find this information in the renderkit JavaDocs available on the Oracle Web site (Google for "HTML_BASIC RenderKit")

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

A Page That Uses the Custom Renderer

```
1    <html xmlns="http://www.w3.org/1999/xhtml"
2       xmlns:h="http://java.sun.com/jsf/html">
3
4      <h:head>
5        <title>Test Renderer</title>
6      </h:head>
7      <h:body>
8        <h3>Test renderer</h3>
9        <p>
10         <h:outputText value="Whatever"/>
11       </p>
12     </h:body>
13   </html>
```



Copyright © Descriptor Systems, 2001-2016. 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 architecture of JSF renderers
- How to write and configure a simple renderer

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum