
Introduction to JSF



Contents

1. JSF overview
2. JSF example

1. JSF Overview

- What is JavaServer Faces technology?
- Architecture of a JSF application
- Benefits of JSF technology
- JSF versions and tools
- Additional JSF technologies

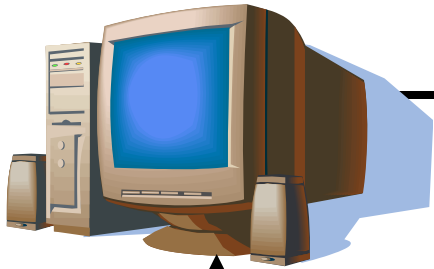
What is JavaServer Faces Technology?

- JavaServer Faces (JSF) technology is a server-side framework for Java-based Web applications
- JSF technology provides an API for:
 - Representing UI components in a Web app
 - Managing the state of UI components
 - Handling events
 - Validating user input
 - Converting data
 - Defining page navigation
 - Supporting internationalization and accessibility

Architecture of a JSF Application

Web container, e.g.
GlassFish, JBoss, Jetty, Tomcat

Browser at the
client computer



HTTP request
to JSP page

JavaServer Faces
Web Page

- Web page contains JSF tags, defining UI of Web page

HTTP response

JavaServer Faces
UI components

- Java objects that correspond to the tags in the Web page
- Event listener objects
- Validator objects
- Data converter objects
- Data encapsulation objects

Benefits of JSF Technology

- Clean separation between presentation and behaviour
 - You can map HTTP requests directly to event-handler methods on Java components (similar to event-handling in rich-client apps)
- Separation of development roles
 - Page authors
 - Application developers
 - Component writers
 - Application architects
 - Tool vendors

JSF Versions and Tools

■ Versions

- JSF 1.1, 1.2
 - See JSF Mojarra: <https://javaserverfaces.dev.java.net/>
- JSF 2
 - Standard part of Java EE 6 onwards

■ Popular JSF tools:

- Oracle JDeveloper
- BEA Workshop Studio
- Sun Java Studio Creator
- Eclipse
- NetBeans

Additional JSF Technologies

- During this course we'll also take a look at:
 - JSF and Ajax
 - Facelets

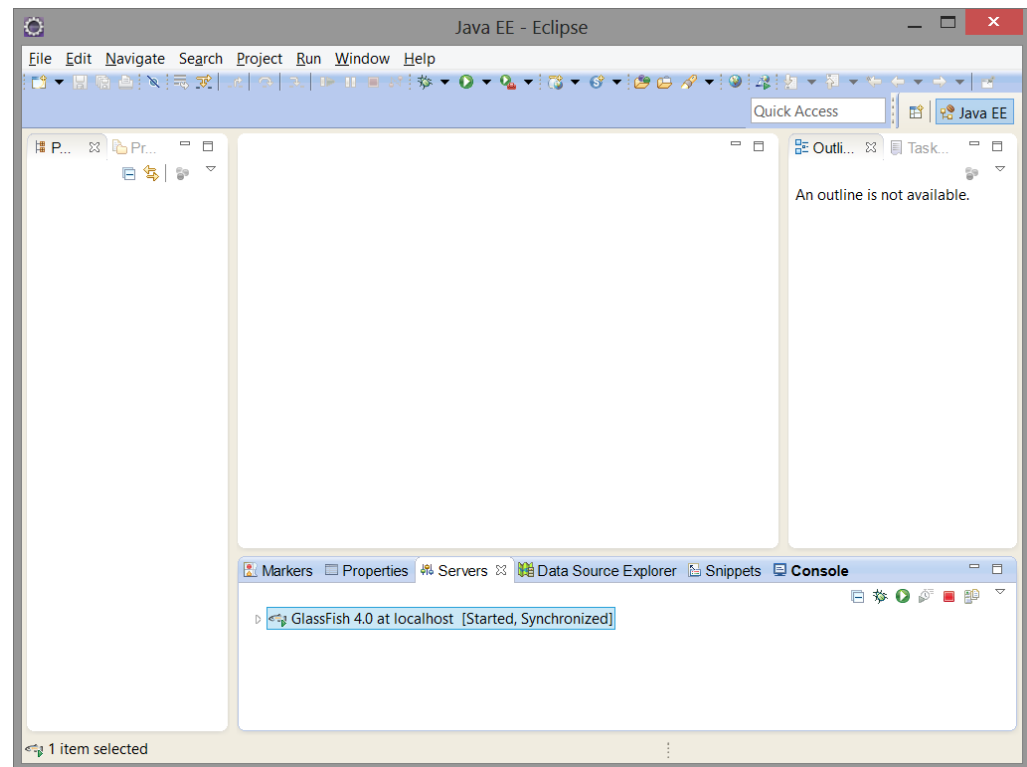
- Most of these technologies and/or tools provide:
 - Additional libraries that you must deploy to the Web container
 - Configuration info that you must add to the JSF config file
 - Tags that you can add to JSP pages

2. JSF Example

- Overview
- Creating a new project
- JSF configuration
- Using JSF components in a Web page
- Requesting the Web page

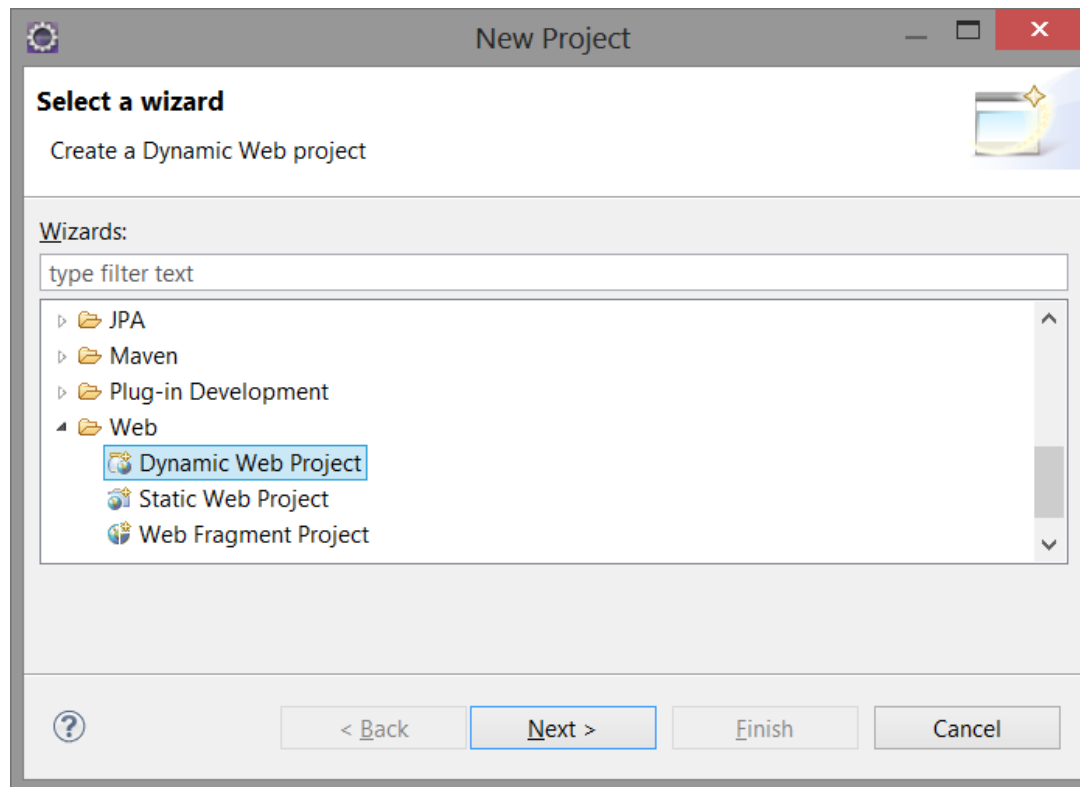
Overview

- This section shows how to use Eclipse to:
 - Create a JSF-enabled Web application
 - Add JSF components to a Web page
 - View the configuration info for the Web application
 - Run the application



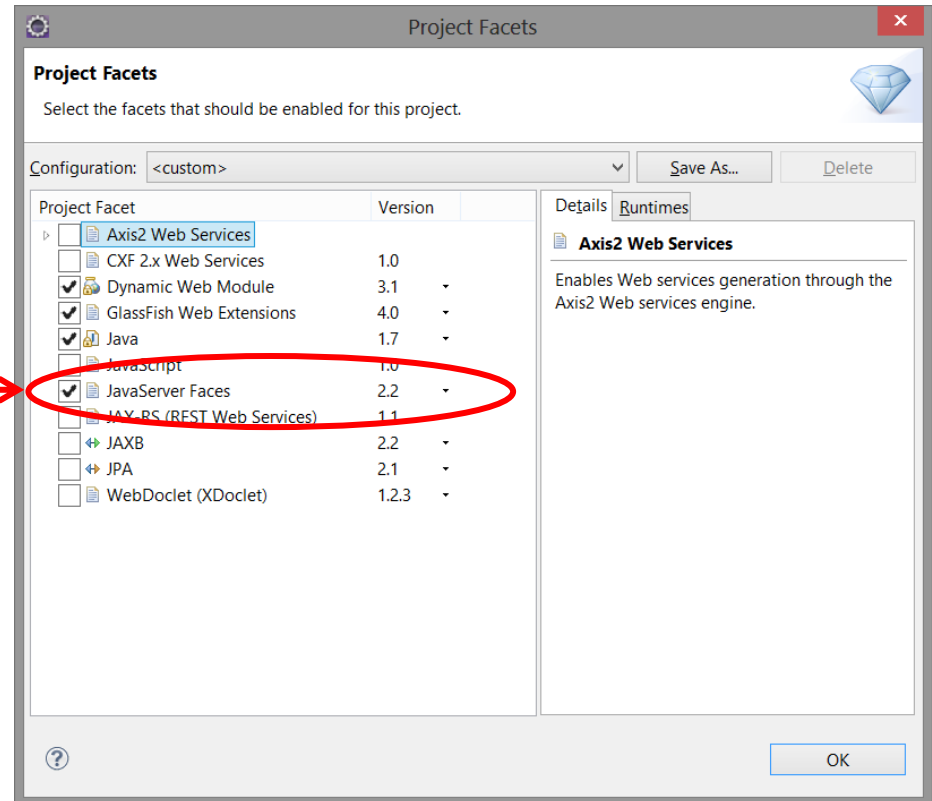
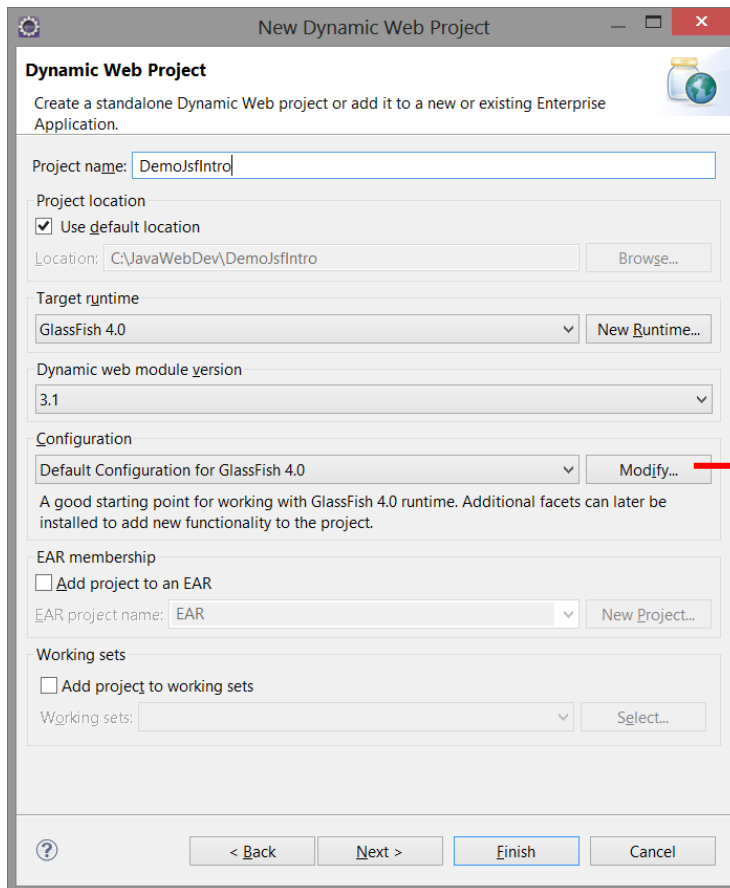
Creating a New Project (1 of 4)

- Every IDE has its own foibles... we'll show how to proceed with Eclipse as a concrete example
- The first step is to create a new Dynamic Web Project



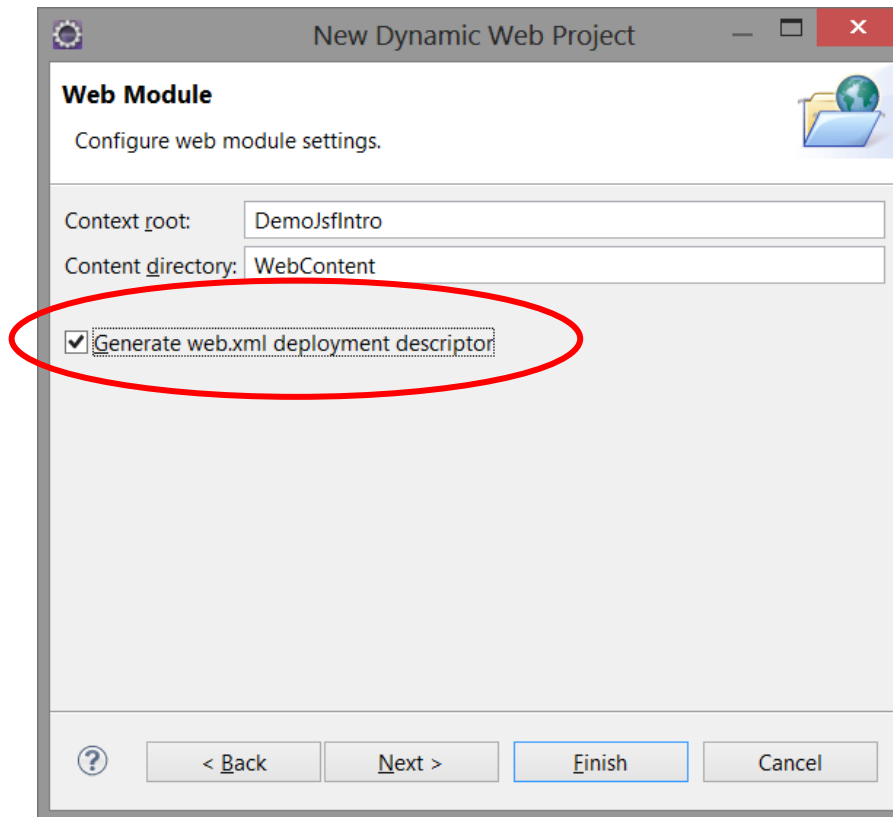
Creating a New Project (2 of 4)

- Specify the name and target runtime of the project
 - Also, modify the configuration to include the JSF project facet



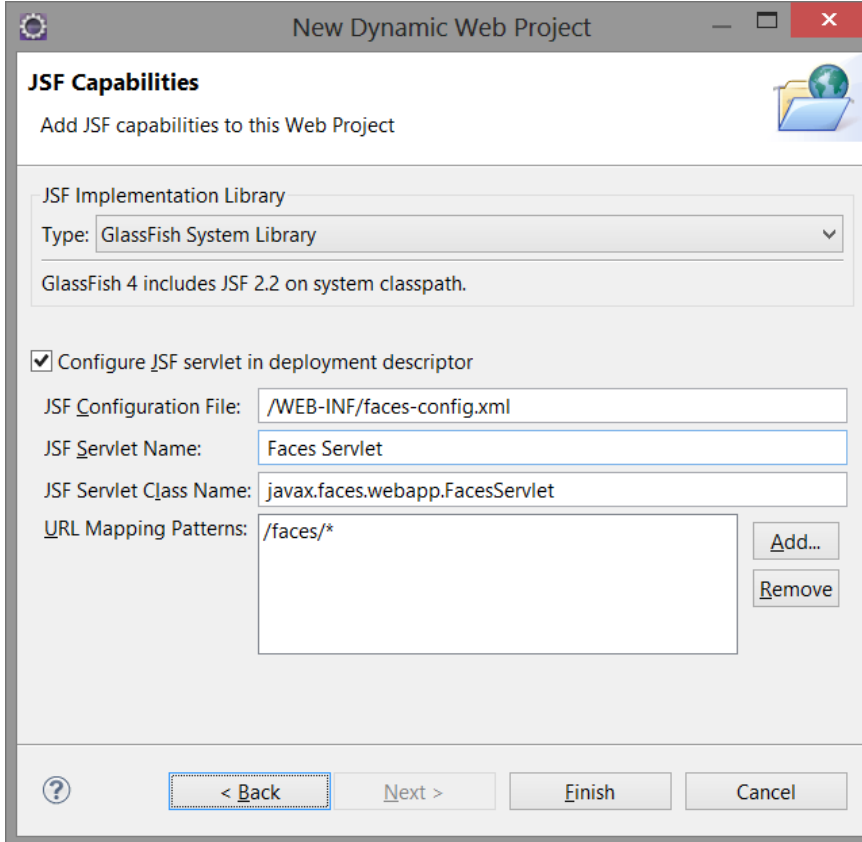
Creating a New Project (3 of 4)

- In the Web Module page, select the option to generate a web.xml file
 - Even though web.xml isn't strictly needed in Java Web apps these days, it's almost always useful to have it



Creating a New Project (4 of 4)

- In the JSF Capabilities page, select the following options to generate a `faces-config.xml` file
 - Even though `faces-config.xml` isn't strictly needed in JSF apps these days, it's almost always useful to have it



The screenshot shows the 'New Dynamic Web Project' dialog box, specifically the 'JSF Capabilities' page. The title bar reads 'New Dynamic Web Project'. The page is titled 'JSF Capabilities' and has a subtitle 'Add JSF capabilities to this Web Project'. Below the subtitle, there is a section for 'JSF Implementation Library' with a dropdown menu set to 'GlassFish System Library'. A note below this section states 'GlassFish 4 includes JSF 2.2 on system classpath.'. There is a checked checkbox for 'Configure JSF servlet in deployment descriptor'. Below this, there are four text input fields: 'JSF Configuration File' (containing '/WEB-INF/faces-config.xml'), 'JSF Servlet Name' (containing 'Faces Servlet'), 'JSF Servlet Class Name' (containing 'javax.faces.webapp.FacesServlet'), and 'URL Mapping Patterns' (containing '/faces/*'). To the right of the 'URL Mapping Patterns' field are 'Add...' and 'Remove' buttons. At the bottom of the dialog, there are four buttons: a help button (question mark), '< Back', 'Next >', 'Finish', and 'Cancel'.

JSF Configuration (3 of 3)

- Here's the initial `faces-config.xml` file:

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<faces-config version="2.2"  
  xmlns="http://xmlns.jcp.org/xml/ns/javaee"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee  
    http://xmlns.jcp.org/xml/ns/javaee/web-facesconfig_2_2.xsd">  
  
</faces-config>
```

faces-config.xml

Web Application Configuration

- JSF-enabled web applications also have an entry (or multiple entries) in `web.xml`
- Here's `web.xml` for our simple Web application:

```
<web-app ... >
```

```
...
```

```
<servlet>
```

```
  <servlet-name>Faces Servlet</servlet-name>
```

```
  <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
```

```
  <load-on-startup>1</load-on-startup>
```

```
</servlet>
```

```
<servlet-mapping>
```

```
  <servlet-name>Faces Servlet</servlet-name>
```

```
  <url-pattern>/faces/*</url-pattern>
```

```
</servlet-mapping>
```

```
</web-app>
```

`web.xml`

Using JSF Components in a Web Page

```
<%@ page ... %>

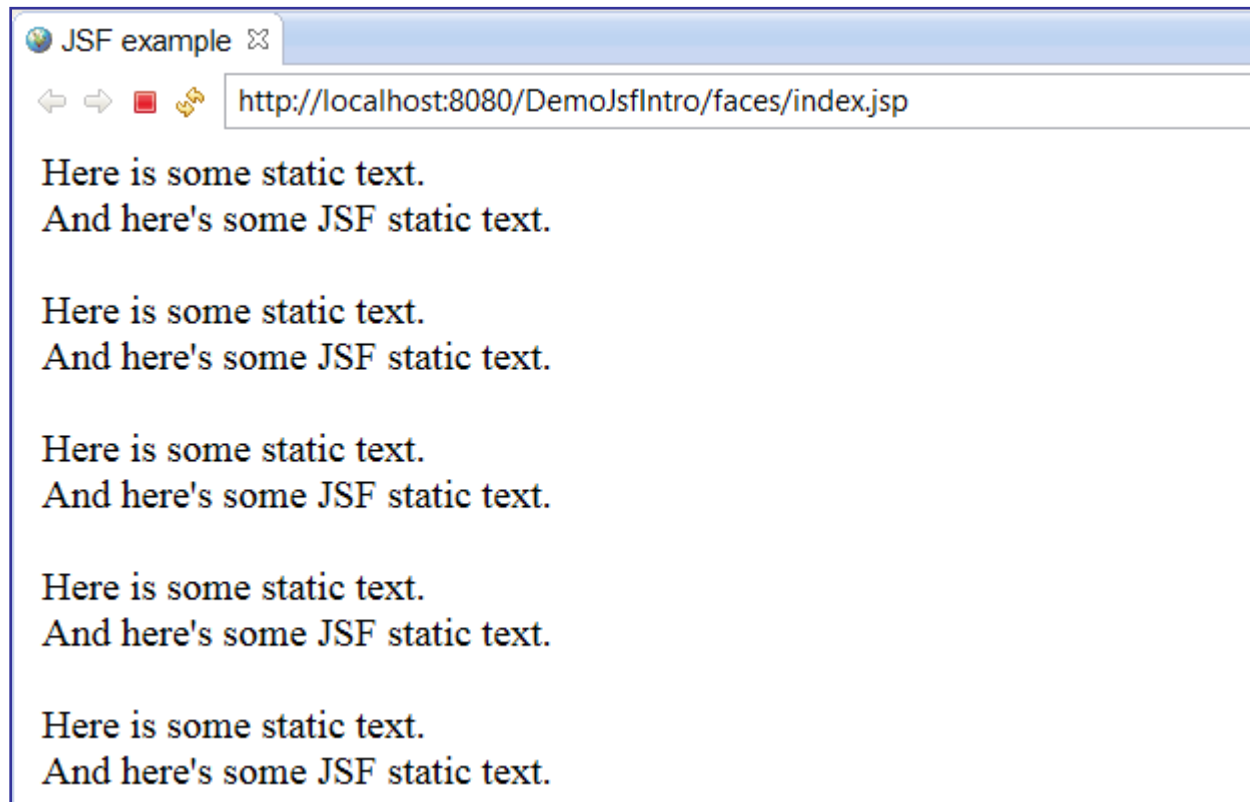
<%@taglib prefix="f" uri="http://java.sun.com/jsf/core"%>
<%@taglib prefix="h" uri="http://java.sun.com/jsf/html"%>

<html>
<body>
  <f:view>
    <% for (int i=0; i < 5; i++) { %>
      Here is some static text. <br>
      <h:outputText value="And here's some JSF static text." /> <br><br>
    <% } %>
  </f:view>
</body>
</html>
```

index.jsp

Requesting the Web Page

- Here's how the JSP page appears when you request it
 - Note the URL!



Summary

- JSF is a Java Web technology that decouples the UI from the model
 - Use JSF components to represent the UI
 - Define backing beans to implement the model
- Configuration files:
 - `faces-config.xml` (optional) - backing beans, navigation, etc.
 - `web.xml` - installs the Faces servlet
- JSF tag library URLs:
 - `<%@taglib prefix="f" uri="http://java.sun.com/jsf/core"%>`
 - `<%@taglib prefix="h" uri="http://java.sun.com/jsf/html"%>`