

## About Eclipse Platform Features

Provider	Feature Name	Version	Feature Id
Oracle	Oracle Coherence Tools	1.6.0.2010070621	oracle.eclipse.tools.helios.coheren
Oracle	Oracle Database Tools	1.6.0.2010070621	oracle.eclipse.tools.helios.databa
Oracle	Oracle GlassFish Server Tools	1.6.0.2010070116	oracle.eclipse.tools.helios.glassfis
Oracle	Oracle Spring Tools	1.6.0.2010070621	oracle.eclipse.tools.helios.spring
Oracle	Oracle Web Tier Tools	1.6.0.2010070621	oracle.eclipse.tools.helios.webtier
Oracle	Oracle WebLogic Scripting Tools	1.6.0.2010070621	oracle.eclipse.tools.helios.weblog
Oracle	Oracle WebLogic Server Tools	1.6.0.2010070621	oracle.eclipse.tools.helios.weblog

# Java EE 6 & GlassFish 3 using Eclipse



Oracle GlassFish Server Tools  
Part of Oracle Enterprise Pack for Eclipse 11.1.1.6.0

Provides tools for developing Java EE applications for GlassFish Server Open Source Edition 3 and Oracle GlassFish Server 3 (Java EE 6) and GlassFish 2 (Java EE 5).

Copyright (c) 2010 Oracle and/or its affiliates.  
All rights reserved. Oracle is a registered trademark of  
Oracle Corporation and/or its affiliates. Other names may  
be trademarks of their respective owners.



License

Plug-in Details

Columns...

Close

## Define a New Server

Choose the type of server to create



[Download additional server adapters](#)

Select the server type:

type filter text

- ▶ Apache
- ▶ BEA Systems, Inc.
- ▶ Basic
- ▼ GlassFish
  - ⌕ GlassFish 2.1 Java EE 5
  - ⌕ GlassFish Server Open Source Edition 3 (Java EE 6)**
  - ⌕ Sailfin v2

GlassFish Server Open Source Edition 3 Java EE 6: Open Source Reference implementation and Production ready.

Server's host name: localhost

Server name: GlassFish Server Open Source Edition 3 (Java EE 6) at lo

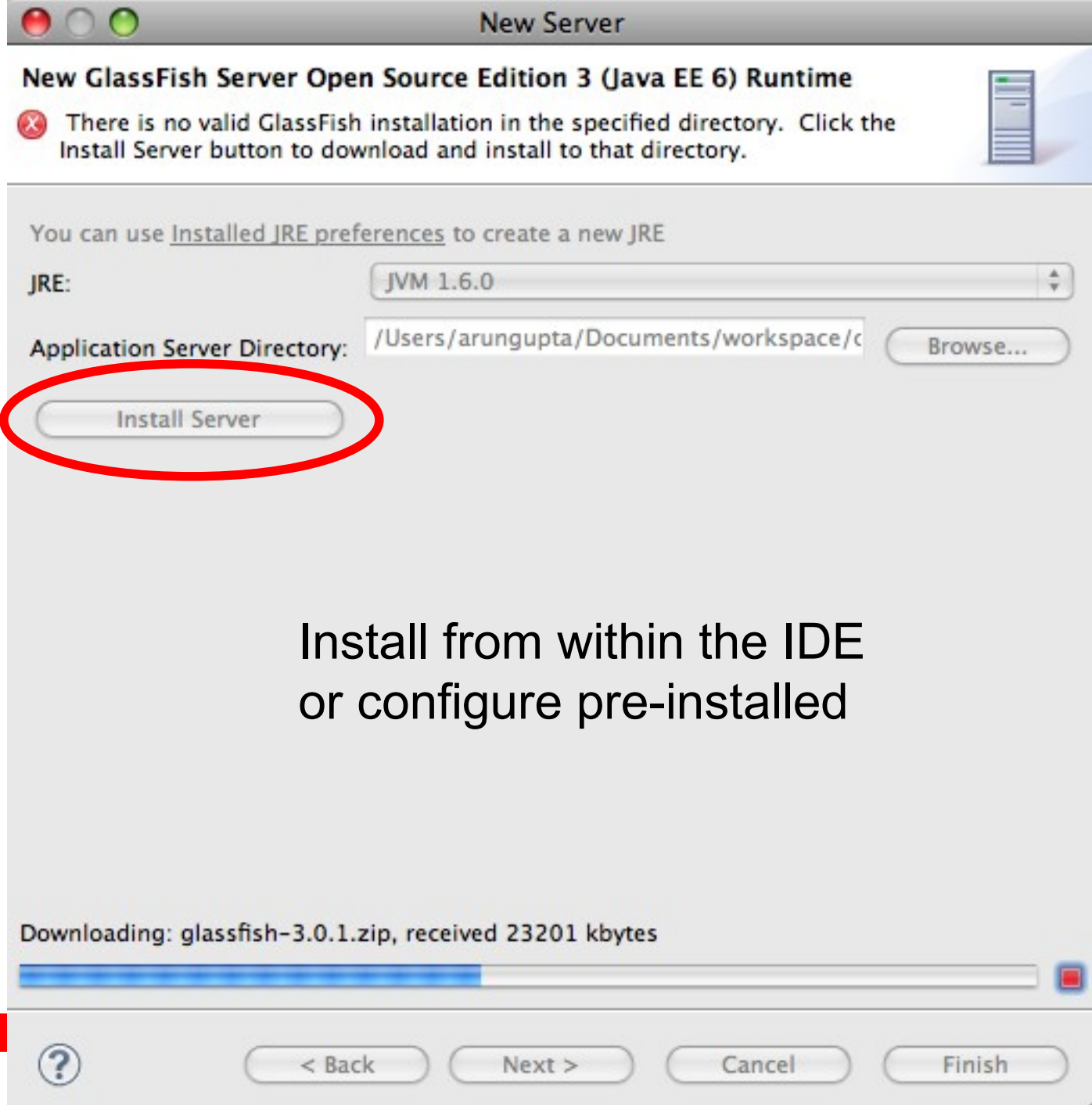


< Back

Next >

Cancel

Finish



Install from within the IDE  
or configure pre-installed

## Dynamic Web Project

Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.



Project name: HelloGlassFish

## Project location

Use default location

Location: /Users/arungupta/Documents/workspace/oepe/HelloGlassFish

Browse...

## Target runtime

GlassFish Server Open Source Edition 3 (Java EE 6)

New Runtime...

## Dynamic web module version

3.0

## Configuration

Default Configuration for GlassFish Server Open Source Edition 3 (Java EE 6)

Modify...

A good starting point for working with GlassFish Server Open Source Edition 3 (Java EE 6) runtime. Additional facets can later be installed to add new functionality to the project.

## EAR membership

Add project to an EAR

EAR project name: EAR

New Project...

## Working sets

Add project to working sets

Working sets:

Select...



### Overview

#### General Information

Specify the host name and other common settings.

Server name:

Host name:

Runtime Environment:

[Open launch configuration](#)

#### Application Server

You can edit some runtime properties for the Application Server (GlassFish or Sun)

Domain Name:

Domain Directory:

Admin Name:

Admin Password:

Server Port Number:

Admin Server Port Number:

**Publishing**

Modify settings for publishing.

Never publish automatically

Automatically publish when resources change

Automatically publish after a build event

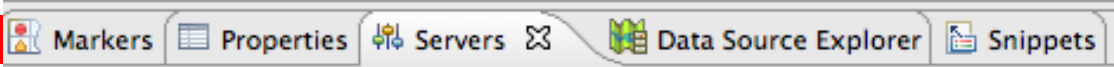
Publishing interval (in seconds):

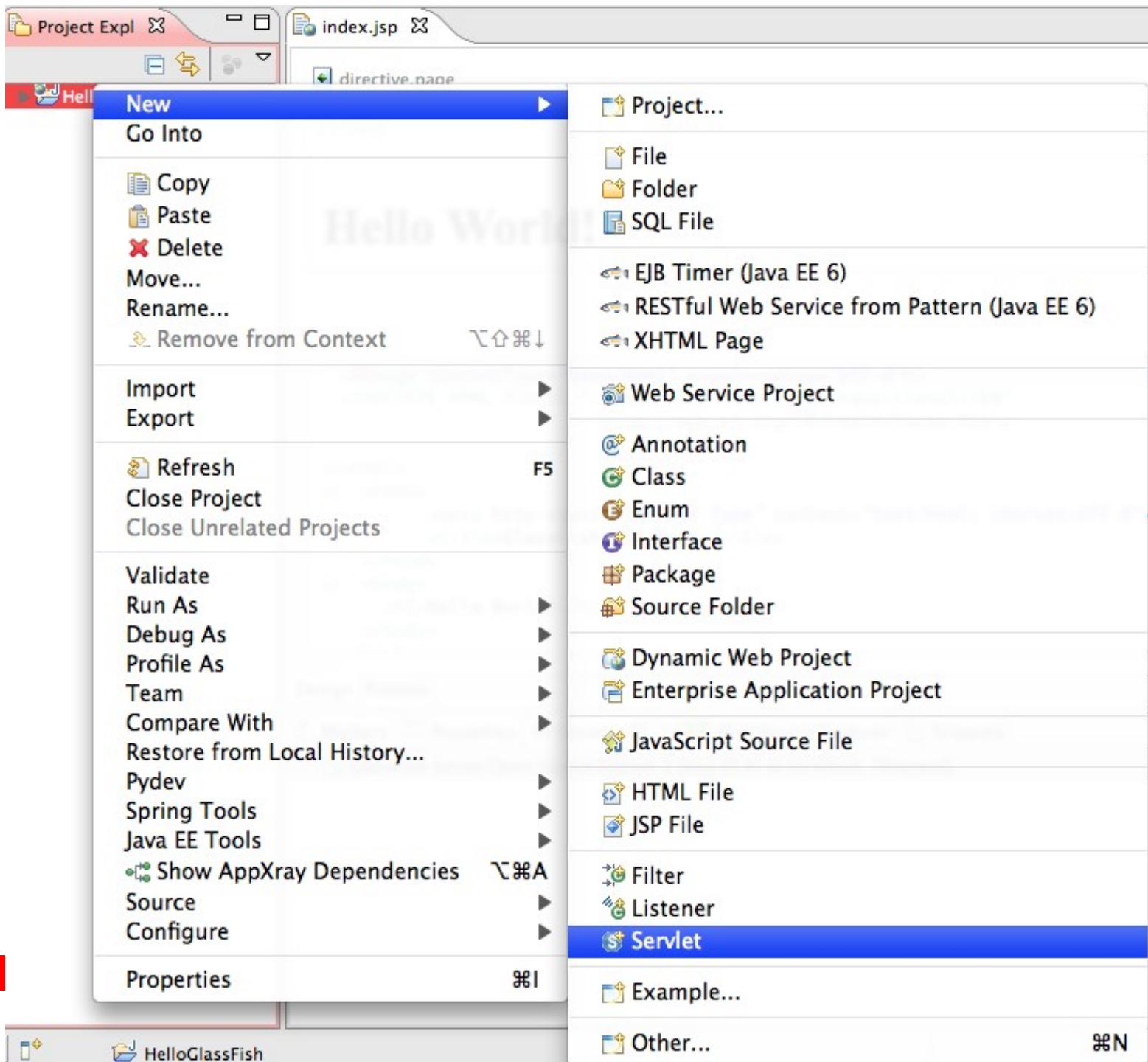
#### Timeouts

# Higher Productivity



### Overview





New

Go Into

Copy

Paste

Delete

Move...

Rename...

Remove from Context

Import

Export

Refresh F5

Close Project

Close Unrelated Projects

Validate

Run As

Debug As

Profile As

Team

Compare With

Restore from Local History...

Pydev

Spring Tools

Java EE Tools

Show AppXray Dependencies

Source

Configure

Properties

Project...

File

Folder

SQL File

EJB Timer (Java EE 6)

RESTful Web Service from Pattern (Java EE 6)

XHTML Page

Web Service Project

Annotation

Class

Enum

Interface

Package

Source Folder

Dynamic Web Project

Enterprise Application Project

JavaScript Source File

HTML File

JSP File

Filter

Listener

Servlet

Example...

Other...



**Create Servlet**  
Specify class file destination.

Project: HelloGlassFish

Source folder: /HelloGlassFish/src Browse...

Java package: server Browse...

Class name: HelloServlet

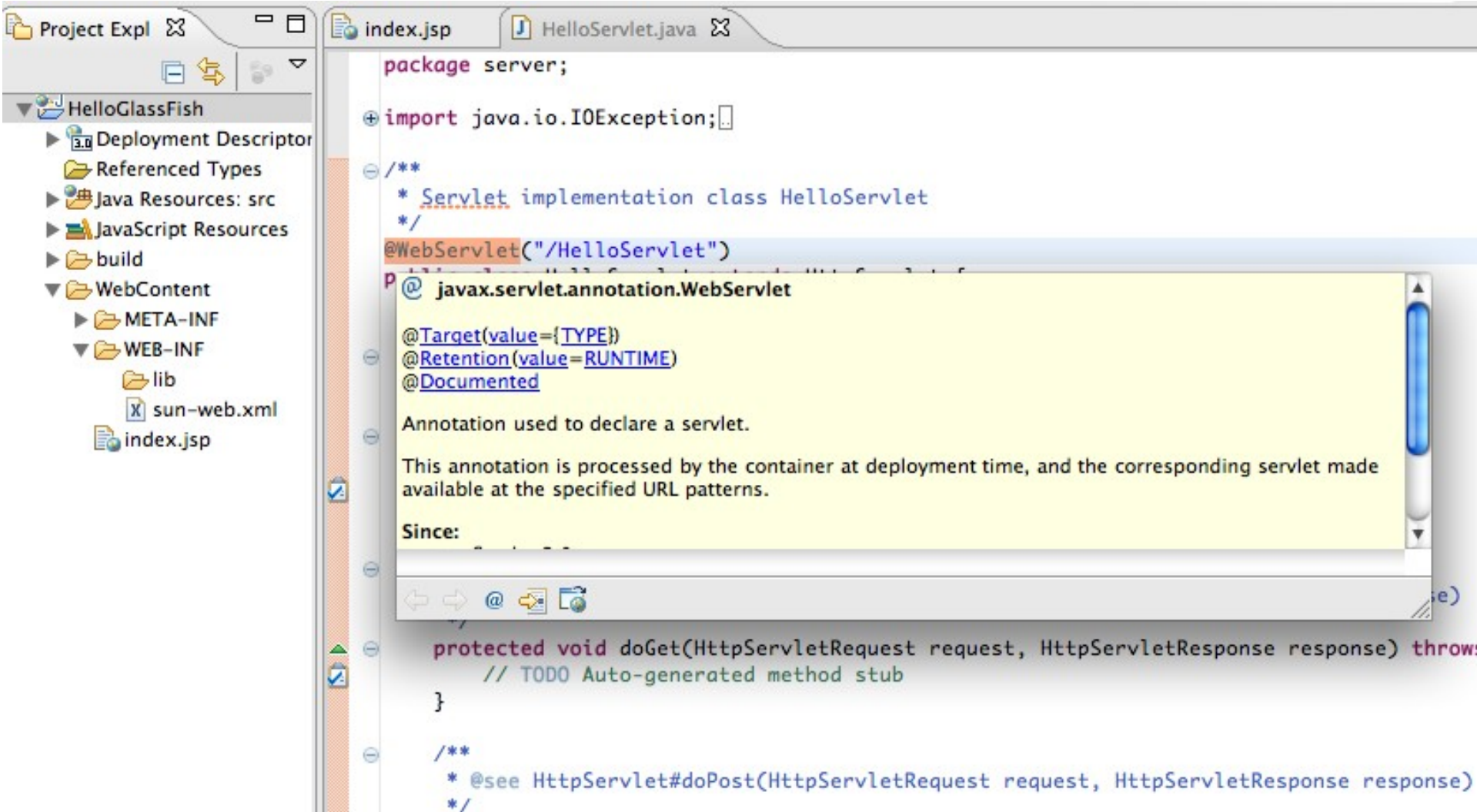
Superclass: javax.servlet.http.HttpServlet Browse...

Use an existing Servlet class or JSP

Class name: HelloServlet Browse...

? < Back Next > Cancel Finish

# Integrated & Off-line Javadocs



The screenshot shows an IDE window with two tabs: `index.jsp` and `HelloServlet.java`. The `HelloServlet.java` tab is active, displaying the following code:

```
package server;

import java.io.IOException;

/**
 * Servlet implementation class HelloServlet
 */
@WebServlet("/HelloServlet")
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
    // TODO Auto-generated method stub
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
```

A tooltip is displayed over the `@WebServlet` annotation, showing the following information:

- @ javax.servlet.annotation.WebServlet**
- @Target(value={ TYPE})**
- @Retention(value=RUNTIME)**
- @Documented**
- Annotation used to declare a servlet.
- This annotation is processed by the container at deployment time, and the corresponding servlet made available at the specified URL patterns.
- Since:

The IDE's Project Explorer on the left shows the project structure for `HelloGlassFish`, including `Deployment Descriptor`, `Referenced Types`, `Java Resources: src`, `JavaScript Resources`, `build`, `WebContent`, `META-INF`, `WEB-INF`, `lib`, `sun-web.xml`, and `index.jsp`.



New

### Select a wizard

Create a new EJB 3.x Session Bean

Wizards:

ejb

- ▼ EJB
  - EJB Project
  - Entity Bean (WebLogic EJBGen)
  - Message-Driven Bean (EJB 3.x)
  - Message-Driven Bean (WebLogic EJBGen)
  - Session Bean (EJB 3.x)**
  - Session Bean (WebLogic EJBGen)
- ▼ GlassFish
  - EJB Timer (Java EE 6)
- ▼ WebLogic Configuration Files
  - WebLogic EJB Module Descriptor

? < Back Next > Cancel Finish

### Create EJB 3.x Session Bean

Specify class file destination.

Project: HelloGlassFish

Source folder: /HelloGlassFish/src

Java package: server

Class name: HelloBean

Superclass:

State type:

- Stateless
- Stateful
- Singleton

Create business interface:

- Remote: server.HelloBeanRemote
- Local: server.HelloBeanLocal
- No-interface

# EJB-in-a-WAR

```
*/
@WebServlet("/HelloServlet")
public class HelloServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @EJB HelloBean bean;
    // ...
}

// TODO Auto-generated method stub
}
```

...  
import java.io.IOException;  
  
import javax.ejb.EJB;  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
...

Press 'Tab' from proposal table or click for focus

# Cron-like Syntax for EJB Timer

Create EJB Timer Callback

Create Java class with an added EJB Timer Callback method.

Java EE 6 Project: HelloGlassFish

Source folder: /HelloGlassFish/src

Java package: server

Class name: MyTimer

Superclass:

Schedule: `second="*/10", minute="*", hour="8-23", dayOfWeek="Mon-Fri", dayOfMonth="*", month="*", year="*", info="MyTimer"`

**Pre-configured JDBC Resource**


The screenshot displays the Oracle SQL Developer interface. On the left, the 'Database Connections' tree shows a hierarchy: 'Sample JavaDB Database (Apache De)' > 'sun-appserv-samples' > 'Schemas' > 'APP' > 'Tables' > 'MYUSER'. The 'MYUSER' table is highlighted with a red box. The main editor shows a SQL query: `create table myuser(name varchar(30) not null primary key, password varchar(30) not null);`. The 'Connection profile' at the top shows 'Type: Derby\_10.x', 'Name: Sample JavaDB', and 'Database: sun-appserv' (circled in red). The 'SQL Results' pane at the bottom shows the query was executed successfully on August 26, 2010, at 0 hr, 0 min, 0 sec, 4 ms.

Status	Operation	Date	Connection Pri
✓	Succeed create table m	Aug 26, 2010	Sample JavaDB



## JPA Facet



 Library configuration is disabled. The user may need to configure further classpath changes later.

## Platform

EclipseLink 2.0.x

## JPA implementation

Type: Disable Library Configuration

The JPA facet requires a JPA implementation library to be present on the project classpath. By disabling library configuration, the user takes on the responsibility of ensuring that the classpath is configured appropriately via alternate means.

## Connection

Sample JavaDB Database

[Add connection ...](#)[Connect](#) Add driver library to build path

Driver: DerbyForSampleDB

 Override default catalog from connection

Catalog:

 Override default schema from connection

Schema: APP

## Persistent class management

 Discover annotated classes automatically Annotated classes must be listed in persistence.xml Create mapping file (orm.xml)

## Select Tables

Select tables to generate entities from.



Connection: Sample JavaDB Database



(Note: You must have an active connection to select schema.)

Schema: APP

Tables:  MYUSER

Java EE Tools

Show AppXray Dependencies ⌘A

Source

Configure

JPA Tools

Properties ⌘I

tipart.c.impl.MultiPartReader  
tipart.impl.MultiPartWriter

HelloGlassFish at /HelloGlassFish

HelloGlassFish at /HelloGlassFish

HelloGlassFish at /HelloGlassFish

Generate Tables from Entities...

Generate Entities from Tables...

Generate Entity from Java Class...

Show in Entity Editor

Update class list in persistence.xml

Restore Defaults

# Facelets using JSF 2.0

## Select XHTML Template

Select a template as initial content in the XHTML page.



Use HTML Template

Templates:

Name	Description
New Facelet Composition Page	Creates a new Facelet page for use with a temp
New Facelet Footer	Creates a footer for use with the Facelet templ
New Facelet Header	Creates a header for use with the Facelet temp
<b>New Facelet Template</b>	<b>Creates a basic header/content/footer Facelet</b>
New XHTML File (1.0 frameset)	xhtml 1.0 frameset
New XHTML File (1.0 strict)	xhtml 1.0 strict
New XHTML File (1.0 transitional)	xhtml 1.0 transitional

Preview:

```
<body>

<div id="header">
  <ui:insert name="header">
    Header area. See comments below this line in the source.
    <!-- include your header file or uncomment the include below
    <!-- <ui:include src="header.xhtml"/> -->
  </ui:insert>
</div>
```

Templates are 'New HTML' templates found in the [HTML Templates](#) preference page.

Name:

Password:

```
<ui:composition template="./template.xhtml">
  <ui:define name="content">
    <h:form>
      <h:panelGrid columns="2">
        <h:outputText value="Name:"/>
        <h:inputText value="#{userBean.name}" title="name"
          id="name" required="true"/>
        <h:outputText value="Password:"/>
        <h:inputText value="#{userBean.password}" title="password"
          id="password" required="true"/>
      </h:panelGrid>
      <h:commandButton action="show" value="submit"/>
    </h:form>
  </ui:define>
</ui:composition>
```

Design Preview



Markers



Properties



Servers



Data Source Explorer



Snippets



Console



/Users/arungupta/Documents/workspace/oepe/glassfishv3/glassfish/domains/domain1/logs/server.log

INFO: Portable JNDI names for EJB HelloBean : [java:global/HelloGlassFish/HelloBean, java:gl

INFO: Initializing Mojarra 2.0.2 (FCS b10) for context '/HelloGlassFish'

INFO: Loading application HelloGlassFish at /HelloGlassFish

INFO: Loading application HelloGlassFish at /HelloGlassFish

INFO: Loading application HelloGlassFish at /HelloGlassFish

INFO: HelloGlassFish was successfully deployed in 2,367 milliseconds.

# RESTful Web services using JAX-RS

The screenshot shows the configuration of a web project in an IDE. On the left, the 'Project Facets' panel lists various technologies, with 'JAX-RS (REST Web Services)' selected. On the right, the 'JAX-RS Capabilities' panel is active, showing the 'JAX-RS Implementation Library' set to 'JAX-RS GlassFish Library'. Below this, the 'Update Deployment Descriptor' checkbox is checked, and the 'Servlet Information' section is populated with the following details:

- JAX-RS servlet name: JAX-RS Servlet
- JAX-RS servlet class name: com.sun.jersey.spi.container.servlet.ServletContainer
- URL mapping patterns: /jaxrs/\*

Buttons for 'Add...' and 'Remove' are visible next to the URL mapping patterns.



## RESTful Web Service from Pattern

Create RESTful Web Service from Pattern.

Project:	<input type="text" value="HelloGlassFish"/>	<input type="button" value="Browse..."/>
Source folder:	<input type="text" value="/HelloGlassFish/src"/>	<input type="button" value="Browse..."/>
Java package:	<input type="text" value="server"/>	<input type="button" value="Browse..."/>
Class name:	<input type="text" value="SimpleResource"/>	
Superclass:	<input type="text"/>	<input type="button" value="Browse..."/>
Pattern type:	<input type="text" value="Simple Root Resource"/>	
Path:	<input type="text" value="generic"/>	
MIME type:	<ul style="list-style-type: none"><li>✓ application/xml</li><li>application/json</li><li>text/plain</li><li>text/html</li></ul>	
Representation class:	<input type="text"/>	<input type="button" value="Browse..."/>
Container representation class:	<input type="text" value="java.lang.String"/>	<input type="button" value="Browse..."/>
Container path:	<input type="text" value="/simpleResources"/>	

# References

- [glassfish.org](http://glassfish.org)
- [oracle.com/goto/glassfish](http://oracle.com/goto/glassfish)
- [blogs.sun.com/theaquarium](http://blogs.sun.com/theaquarium)
- [glassfish.org/roadmap](http://glassfish.org/roadmap)
- [youtube.com/user/GlassFishVideos](http://youtube.com/user/GlassFishVideos)
- [blogs.sun.com/arungupta/entry/screencast\\_31\\_java\\_ee\\_6](http://blogs.sun.com/arungupta/entry/screencast_31_java_ee_6)
- Follow [@glassfish](https://twitter.com/glassfish)