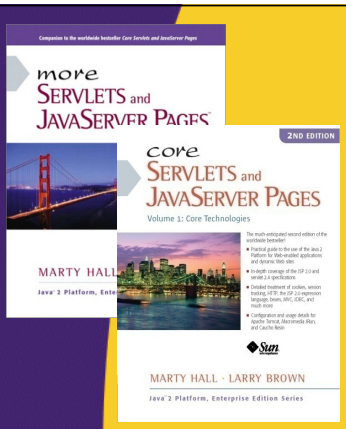
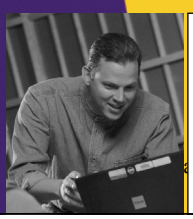
An Overview of Servlet & JSP Technology

Originals of Slides and Source Code for Examples:
http://courses.coreservlets.com/Course-Materials/csajsp2.html





Agenda

- **What servlets and JSP are all about**
  - Understanding the role of servlets
  - Building Web pages dynamically
  - Evaluating servlets vs. other technologies
  - Understanding the role of JSP
- **Testing Tomcat with Eclipse**
  - Installing Tomcat
  - Installing and starting Eclipse
  - Telling Eclipse about Tomcat
  - Deploying and running Web apps from Eclipse
  - Making new Web apps in Eclipse

What Servlets and JSP are All About

Why Web Apps?

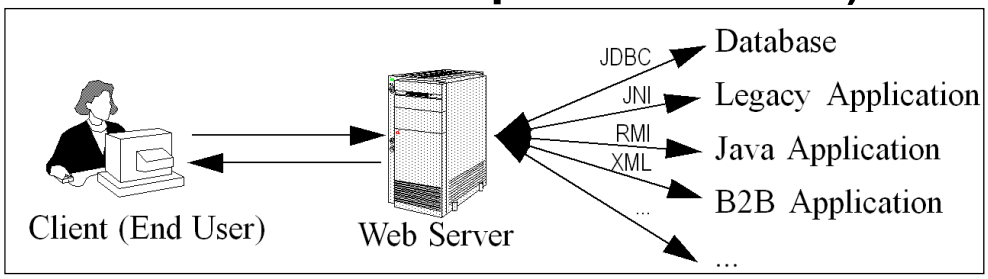
- **Downsides to browser-based apps**
  - GUI is poor
    - HTML is OK for static documents, but lousy for programs
  - Communication is inefficient
    - HTTP is poor protocol for the way we now use Web apps

Why Web Apps? (Continued)

- **So why does everyone want Web apps?**
  - Universal access
    - Everyone already has a browser installed
    - Any computer on the network can access content
  - Automatic "updates"
    - Content comes from server, so is never out of date


A Servlet's Job

- **Read explicit data sent by client (form data)**
- **Read implicit data sent by client (request headers)**
- **Generate the results**
- **Send the explicit data back to client (HTML)**
- **Send the implicit data to client (status codes and response headers)**

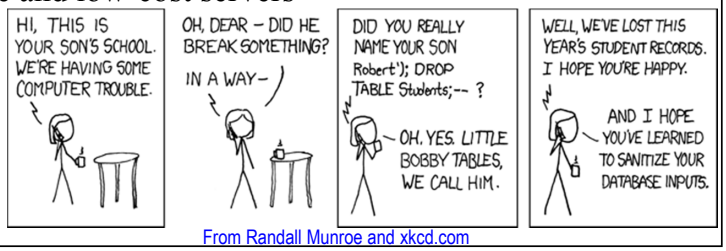


Why Build Web Pages Dynamically?

- **The Web page is based on data submitted by the user**
  - E.g., results page from search engines and order-confirmation pages at on-line stores
- **The Web page is derived from data that changes frequently**
  - E.g., a weather report or news headlines page
- **The Web page uses information from databases or other server-side sources**
  - E.g., an e-commerce site could use a servlet to build a Web page that lists the current price and availability of each item that is for sale.

The Advantages of Servlets Over "Traditional" CGI

- **Efficient**
  - Threads instead of OS processes, one servlet copy
- **Convenient**
  - Lots of high-level utilities
- **Powerful**
  - Sharing data, pooling, persistence
- **Portable**
  - Run on virtually all operating systems and servers
- **Inexpensive**
  - There are plenty of free and low-cost servers
- **Secure**
  - No shell escapes, no buffer overflows
- **Mainstream**
  - See next page



From Randall Munroe and xkcd.com

Mainstream

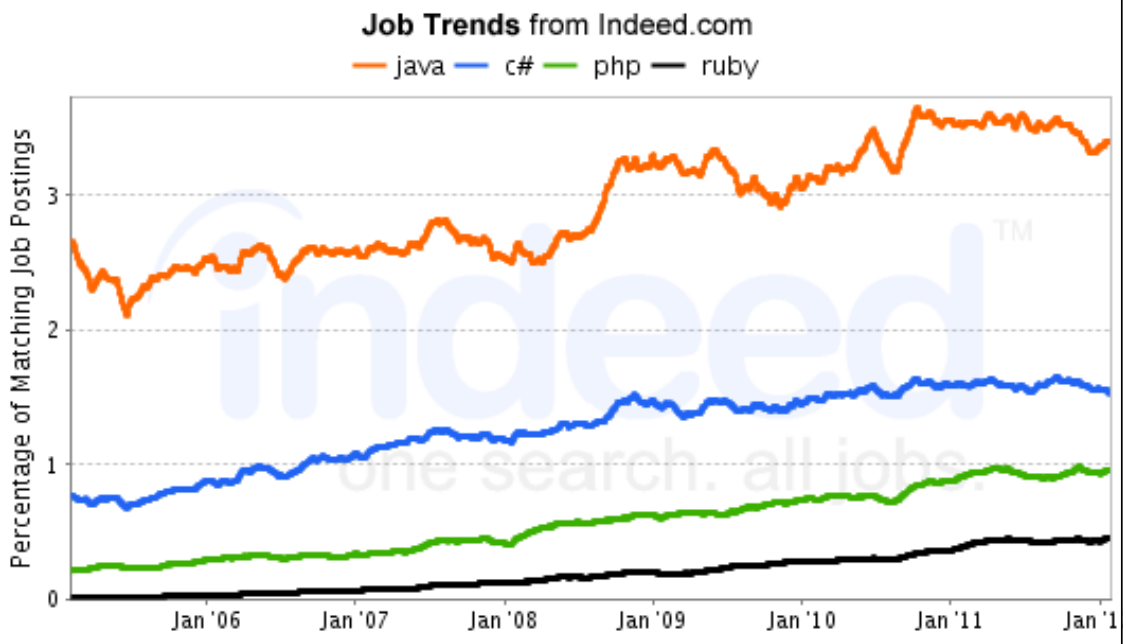
- **Popular:**
  - The single most common use of Java technology
  - The leading technology for medium/large Web applications
    - Google reports over 650 million Web pages using JSP
- **Supported by:**
  - Apache, Oracle, IBM, Sybase, BEA, Jetty, Caucho, Sun, New Atlanta, ATG, Fujitsu, Lutris, Silverstream, the World Wide Web Consortium (W3C), and many others
  - Plugins for IIS and Zeus
- **Runs on:**
  - Windows, Unix/Linux, MacOS, VMS, and IBM mainframe OSs
- **Used for:**
  - Airline companies, hotels, e-commerce sites, search engines, banks, financial sites, etc., etc., etc.

Ten Most Popular Web Sites (Alexa.com, 2010)

1. **Google**
   - Java (Web), C++ (indexing)
2. **Facebook**
   - PHP, Java (Hadoop)
3. **YouTube**
   - Flash, Python, Java
4. **Yahoo**
   - PHP and Java
5. **Microsoft Live.com**
   - .NET
6. **Baidu**
   - Unknown
7. **Wikipedia**
   - PHP
8. **Blogger**
   - Java
9. **MSN**
   - .NET
10. **Twitter**
    - Ruby on Rails, Scala, Java

Fall 2010: Google reports over two *billion* Web pages that use JSP (inurl:jsp).

Web App Language Popularity: Keywords in Job Postings



Higher-Level Alternative: JSF 2

- **Servlets and JSP**
  - Well-established standard
  - Used by google.com, ebay.com, walmart.com, and thousands of other popular sites
  - Relatively low level by today's standards
  - Covered in this tutorial
- **JSF (JavaServer Faces) Version 2**
  - Now an official part of Java EE 6
    - But runs in any recent Java-enabled server, including Tomcat 6+
  - Higher-level features: integrated Ajax support, field validation, page templating, rich third-party component libraries, etc. Designed around the MVC approach.
  - Not yet as widely used, but recommended for many or most new projects
  - Covered at http://www.coreservlets.com/JSF-Tutorial/jsf2/

Extending the Power of Servlets: JavaServer Pages (JSP)

- **Idea:**
  - Use regular HTML for most of page
  - Mark dynamic content with special tags
  - Details in second half of course

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD><TITLE>Welcome to Our Store</TITLE></HEAD>
<BODY>
<H1>Welcome to Our Store</H1>
<SMALL>Welcome,
<!-- User name is "New User" for first-time visitors -->
<%= coreservlets.Utils.getUserNameFromCookie(request) %>
To access your account settings, click
<A HREF="Account-Settings.html">here.</A></SMALL>
<P>
Regular HTML for  rest of on-line store's Web page
</BODY></HTML>
```

Accessing the Online Documentation

- **Servlets and JSP**
  - http://docs.coreservlets.com/servlet-3.0-api/
    - Servlets 3.0 and JSP 2.2 (Tomcat 7)
  - http://docs.oracle.com/cd/E17802_01/products/products/servlet/2.5/docs/servlet-2_5-mr2/
    - Servlets 2.5 (Tomcat 6)
  - http://docs.oracle.com/cd/E17802_01/products/products/jsp/2.1/docs/jsp-2_1-pfd2/
    - JSP 2.1 (Tomcat 6)
- **Java 6 and 7**
  - http://docs.oracle.com/javase/7/docs/api/
  - http://docs.oracle.com/javase/6/docs/api/
    - Class uses Java 6 and Tomcat 7
- **Advice**
  - If you have a fast and reliable internet connection, bookmark these addresses. If not, download a copy of the APIs onto your computer and use it locally.

Setting Up Tomcat on Your PC

- **Tomcat 7 with Eclipse**
  - http://www.coreservlets.com/Apache-Tomcat-Tutorial/tomcat-7-with-eclipse.html
    - Or, just follow link at top left of www.coreservlets.com
    - More details in next section of this tutorial
- **Tomcat 6 with Eclipse**
  - http://www.coreservlets.com/Apache-Tomcat-Tutorial/
- **For manual execution**
  - http://www.coreservlets.com/Apache-Tomcat-Tutorial/
    - More details in last section.
    - Eclipse or another IDE *strongly* recommended
- **Bottom line**
  - Unzip Tomcat, then point Eclipse at the install folder

Installing Java and Tomcat

For even more detailed step-by-step instructions, see tutorials on using Eclipse with Tomcat 6 or Tomcat 7 at http://www.coreservlets.com/Apache-Tomcat-Tutorial/

Installing Java SE 6

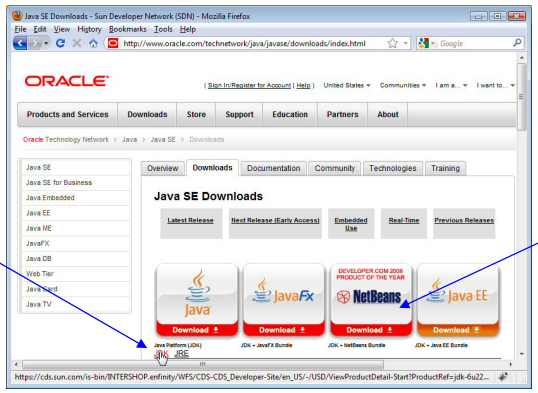
- **Minimum Java version**
  - Tomcat 7 (servlets 3.0) requires Java 6
  - Tomcat 6 and other servlet 2.5 containers require Java 5+
    - But Java 6 recommended for performance and features
- **Downloading and installation**
  - Follow directions at http://www.oracle.com/technetwork/java/javase/downloads/
    - Choose"JDK", not "JRE"
      - Not "with Java EE", "with JavaFX", or "with NetBeans"

Installing Java SE (Standard Edition)

- **Install Java 6**

  http://www.oracle.com/technetwork/java/javase/downloads/

Use this version. The "JDK – Java Development Kit" includes compiler for .java files, whereas the "JRE – Java Runtime Environment" is only for executing prebuilt .class files.



This tutorial uses Eclipse, but if you prefer the NetBeans environment, it is very easy to adapt the instructions to that development environment. So, if you prefer NetBeans or your organization has standardized on it, use this download instead of (not in addition to) the one on the left.

- **Bookmark the Java API ("JavaDocs")**
  - http://download.oracle.com/javase/6/docs/api/
    - This is the most important Java reference for developers. Eclipse integrates this API, but a separate link is still good

Download and Unzip Tomcat

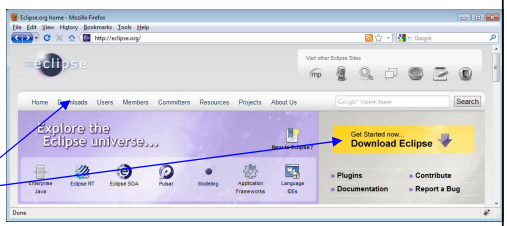
- **Start at http://tomcat.apache.org**
  - Choose download link on left, then ZIP version
    - Tomcat 7 (recommended)
    - Tomcat 6 (if you need compatibility with older servers)
- **Or, go to http://www.coreservlets.com/**
  - Choose Tomcat tutorial from top left
  - This is preconfigured version
    - Set for development, not deployment mode
      - Port changed to 80, servlet reloading enabled, directory listings turned on, etc.
    - Otherwise unchanged
- **Either way, just unzip the file**
  - E.g., resulting in C:\apache-tomcat-7.0.8

Installing Eclipse

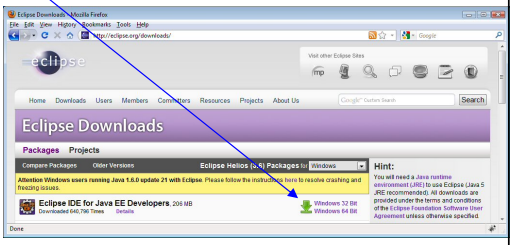
For even more detailed step-by-step instructions, see tutorials on using Eclipse with Tomcat 6 or Tomcat 7 at http://www.coreservlets.com/Apache-Tomcat-Tutorial/

Installing Eclipse

- **Overview**
  - Eclipse is a free open source IDE. Support for Java, Android, HTML, CSS, JavaScript, C++, PHP, JSF, servlets, and more.
    - http://eclipse.org/downloads/
    - Choose "Eclipse IDE for Java EE Developers"
      - Need version 3.6 or later for JSF 2.0 and Tomcat 7 support



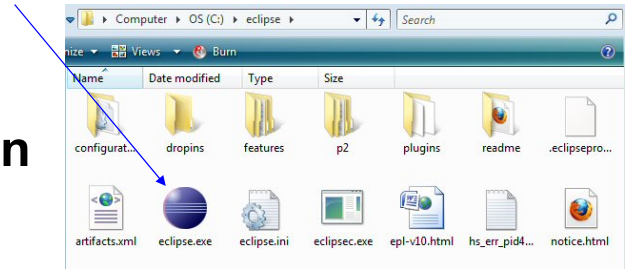
- **Features**
  - Checks your syntax as you type
  - Automatically compiles every time you save file
  - Many tools: refactoring, debugging, server integration, templates for common tasks, etc.
    - Low learning curve: beginners can use Eclipse without knowing these tools



Reminder: step-by-step guide at http://www.coreservlets.com/ (click "Apache Tomcat 7" in top left).

Running Eclipse

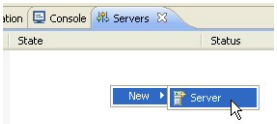
- **Unzip the downloaded file (no installer!)**
  - Call the folder you unzip into "installDir"
- **Double click eclipse.exe**
  - From *installDir*/bin
- **Click on "Workbench" icon**
  - Next time you bring up Eclipse, it will come up in workbench automatically

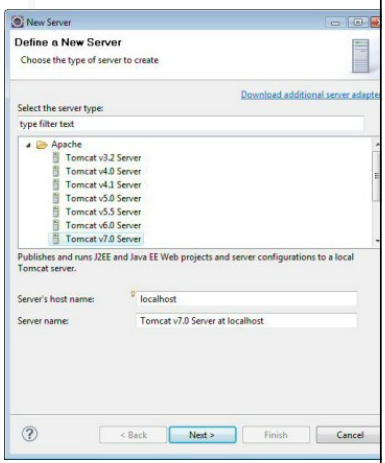


- **Shortcut**
  - Many developers put Eclipse link on their desktop
    - R-click eclipse.exe, Copy, then go to desktop, R-click, and Paste Shortcut (not just Paste!)

Configuring Eclipse

- **Tell Eclipse about Java version**
  - Window → Preferences → Java → Installed JREs → Press "Add", choose "Standard VM", navigate to JDK folder (not "bin" subdirectory)
    - E.g., C:\Program Files\Java\jdk1.6.0_21
- **Tell Eclipse about Tomcat**
  - Click on Servers tab at bottom. R-click in window.
  - New, Server, Apache, Tomcat v7.0, Next, navigate to folder, Finish.
- **JSF 2.0 support**
  - Eclipse 3.6 has support for JSF 2.
    - R-click and add Project Facet for JSF 2
    - R-click .xhtml files and Open With, Web Page Editor
    - Double-click faces-config.xml



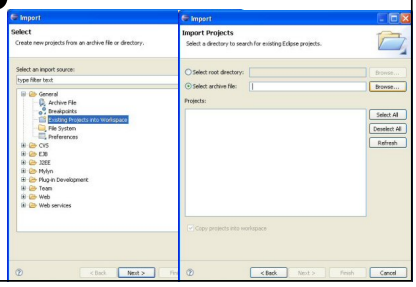


Tomcat v7.0 is choice only in Eclipse 3.6 (Helios). If you prefer Tomcat 6, choose Tomcat v6.0 above instead. If you lose the "Servers" tab at the bottom of Eclipse, use Window, Show View, and hunt for "Servers".

Deploying Apps from Eclipse

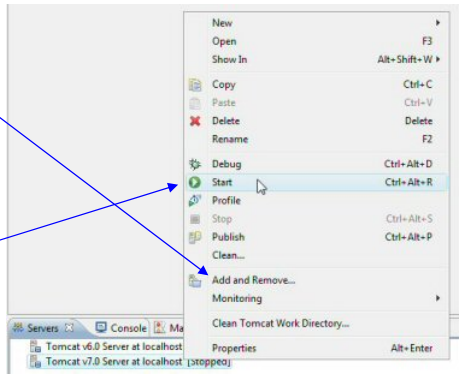
Download and Import Sample Project

- **Get test-app.zip from coreservlets.com**
  - Start at servlet and JSP tutorials
    - http://courses.coreservlets.com/Course-Materials/csajsp2.html
      - Go to first section (Overview and Setup)
  - Or, start at Apache Tomcat tutorial
    - http://www.coreservlets.com/Apache-Tomcat-Tutorial/
      - Choose Tomcat 7 (recommended) or Tomcat 6 version
- **Then, download test-app.zip**
  - Then, import into Eclipse.
    - File, Import, General, Existing Projects, Select archive file. Then click Browse and navigate to test-app.zip.



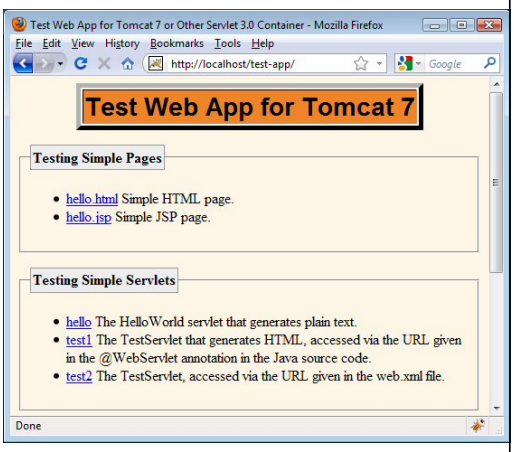
Deploying App in Eclipse

- **Deploy project**
  - Select "Servers" tab at bottom
  - R-click on Tomcat
  - Choose "Add and Remove"
  - Choose project
  - Press "Add"
  - Click "Finish"
- **Start Server**
  - R-click Tomcat at bottom
  - Start (use "Restart" if Tomcat already running)
- **Test URL**
  - http://localhost/test-app/ in any Web browser



Testing Deployed App in Eclipse

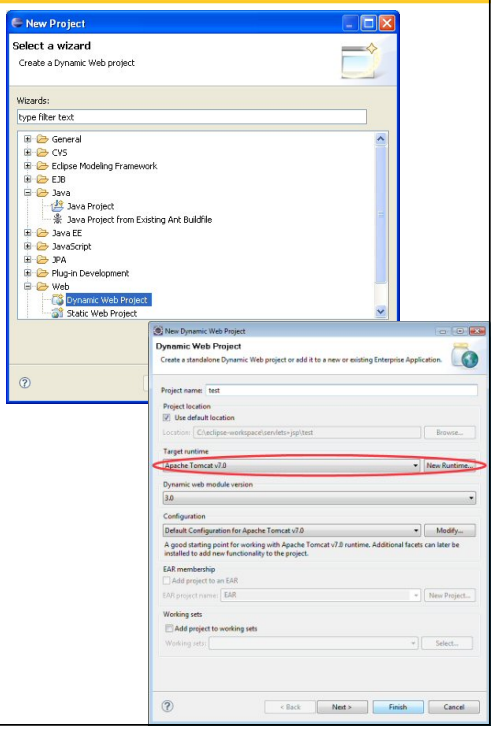
- **Start a browser**
  - Eclipse also has builtin browser, but I prefer to use Firefox, IE, or Chrome separately
- **Test base URL**
  - http://localhost/test-app/
- **Test Web content**
  - http://localhost/test-app/hello.html
  - http://localhost/test-app/hello.jsp
- **Test servlets**
  - http://localhost/test-app/hello
  - http://localhost/test-app/test1
  - http://localhost/test-app/test2



Making New Apps from Eclipse

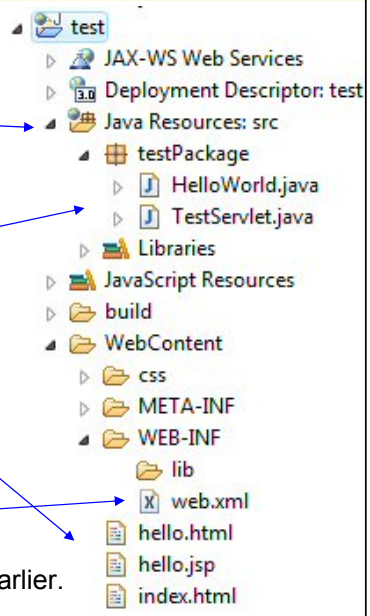
Making Web Apps in Eclipse

- **Make empty project**
  - File → New → Project → Web → Dynamic Web Project
  - For "Target runtime", choose "Apache Tomcat v7.0"
  - Give it a name (e.g., "test")
  - Accept all other defaults
- **Shortcut**
  - If you have made Dynamic Web Project recently in workspace, you can just do File → New → Dynamic Web Project



Adding Code to Eclipse Projects

- **Locations**
  - Java Resources: src
    - R-click and New → Package
    - Never use default package
  - src/testPackage
    - Java code in testPackage package
  - WebContent
    - Web files (HTML, JavaScript, CSS, JSP, images, etc.)
  - WebContent/some-subdirectory
    - Web content in subdirectory
    - R-click on WebContent, New → Folder
  - WebContent/WEB-INF
    - web.xml
      - Optional with servlets 3.0. Required in 2.5 & earlier.
      - Will be discussed later
- **Note**
  - Can cut/paste or drag/drop files into appropriate locations



Testing New App

- **Follow same procedure as "deploying app" from previous section**
  - Deploy project
    - Select "Servers" tab at bottom
    - R-click on Tomcat
    - Choose "Add and Remove"
    - Choose project
    - Press "Add"
    - Click "Finish"
  - Start Server
    - R-click Tomcat at bottom
    - Restart (use "Start" if Tomcat not already running)
  - Test URL
    - http://localhost/*appName*/ in any Web browser

Summary

- **General**
  - Servlets are efficient, portable, powerful, and widely accepted in industry
  - Regardless of deployment server, run a free server on your desktop for development
  - Using Eclipse (or another IDE like NetBeans or IntelliJ IDEA) greatly simplifies development and deployment
  - Consider JSF 2 as an alternative for new projects
    - http://www.coreservlets.com/JSF-Tutorial/jsf2/
- **Getting started**
  - Start with test-app and TestServlet from coreservlets.com
  - Click on "Intermediate Servlets and JSP" tutorial in top-left corner and you can get pre-made Eclipse projects

Questions?