

10 Ways to Start a Java Web App

Tomcat / Jetty / ...
x Shell / Console / Maven / Eclipse

Arnaud Nauwynck

This document:

<http://arnaud.nauwynck.chez-alice.fr/CoursIUT/10-ways-start-JavaWebApp.pdf>

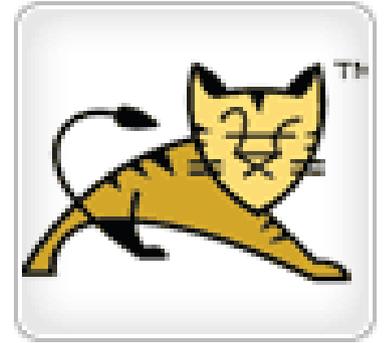
Table Of Content

- Intro : Java Web App, Java Web Servers
- Tomcat
 - Shell
 - Console
 - Maven
 - Eclipse
- Jetty
 - ...

Intro : WAR and Web Servers

- WAR = Web Archive = zip containing ...
 - WEB-INF/web.xml the entry-point
 - *.html, *.png, *.js ... static web files
 - WEB-INF/lib/, classes ... dynamic
- Web Server = Servlet Container
 - NOT apache ! (no java servlet mod_)
 - Tomcat, Jetty, Weblogic, Websphere, Glassfish, Jboss, Felix, ...
 - http://en.wikipedia.org/wiki/Comparison_of_application_servers

Who is “Tom le Chat” ?



- his little name “Catalina”
- Most famous Jakarta Apache project
- Install it : download + unzip
- Run it:
 - `<<tomcat>>/bin/statup.sh`
- Deploy on it
 - `cp myapp.jar <<tomcat>>/webapps/`

Easy!

Notes on Tomcat Scripts

- “bin/startup” = shortcut for “catalina.sh start”
Starts in ~700ms ... !!!
- “bin/shutdown.sh” = ... catalina stop
(using socket control or PID file + kill)
- “bin/catalina.sh” ... shortcut for
java -D... -classpath=...
org.apache.catalina.startup.Bootstrap start

fast

standard

Tomcat Script Checks

- `ps -edf | grep java | grep catalina`

```
[apache-tomcat-7.0.23] ps -edf |grep java | grep catalina
soft      5168      1  0 21:49 pts/0    00:00:05 /usr/bin/java -Djava.util.logging.config.file=/n
:-7.0.23 -Dcatalina.home=/mnt/a_sdd-2/shared/devtools/tomcat/apache-tomcat-7.0.23 -Djava.io.tmpd
/shared/devtools/tomcat/apache-tomcat-7.0.23/temp org.apache.catalina.startup.Bootstrap start
```

- `tail -100f logs/catalina.out`

```
[apache-tomcat-7.0.23] tail -5f logs/catalina.out
INFO: Starting ProtocolHandler ["http-bio-8080"]
Jan 23, 2012 9:49:50 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-bio-8009"]
Jan 23, 2012 9:49:50 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 787 ms
```

- `Jconsole, jps`

Network command check...

- netstat -ln

protocol : "http-alt"
... with -n : "8080"
(--numeric-port)
memo: -nlapute

```
$ netstat -l | grep 8080
$
$ netstat -l | grep http
tcp6      0      0 [::]:http-alt  [::]:*  LISTEN
$
$ netstat -ln | grep 8080
tcp6      0      0 :::8080      :::*    LISTEN
```

- lsof -P

```
$ lsof -P | grep 8080
java      6969      soft  42u   IPv6 127354    0t0  TCP *:8080 (LISTEN)
$
$ lsof | grep http
java      6969      soft  42u   IPv6 127354    0t0  TCP *:http-alt (LISTEN)
```

- TELNET, WGET ..

```
[apache-tomcat-7.0.23] telnet localhost 8080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET /

<!DOCTYPE html>
```

Start in Debug Mode

- Catalina JPDA “Debug” facilities

```
bin/catalina.sh jpda start
```

- Internally..

- `-agentlib:jdwp=transport=dt_socket,address=8000,server=y,suspend=n`

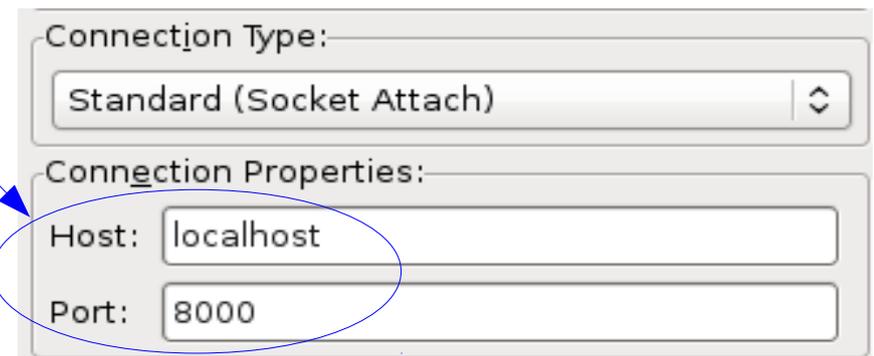
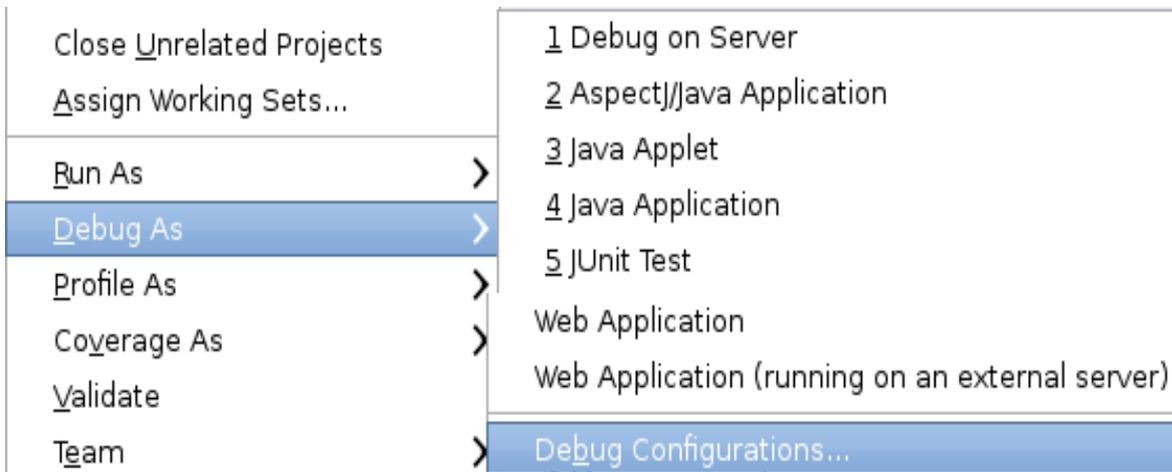
- (equivalent in `mvnDebug...`)

```
-Xdebug -Xnoagent -Djava.compiler=NONE
```

```
-Xrunjdwp:transport=dt_socket,server=y,suspend=y,address=8000
```

- To debug the beginning, change `"suspend=y"`

(Remote) Attach Eclipse Debug

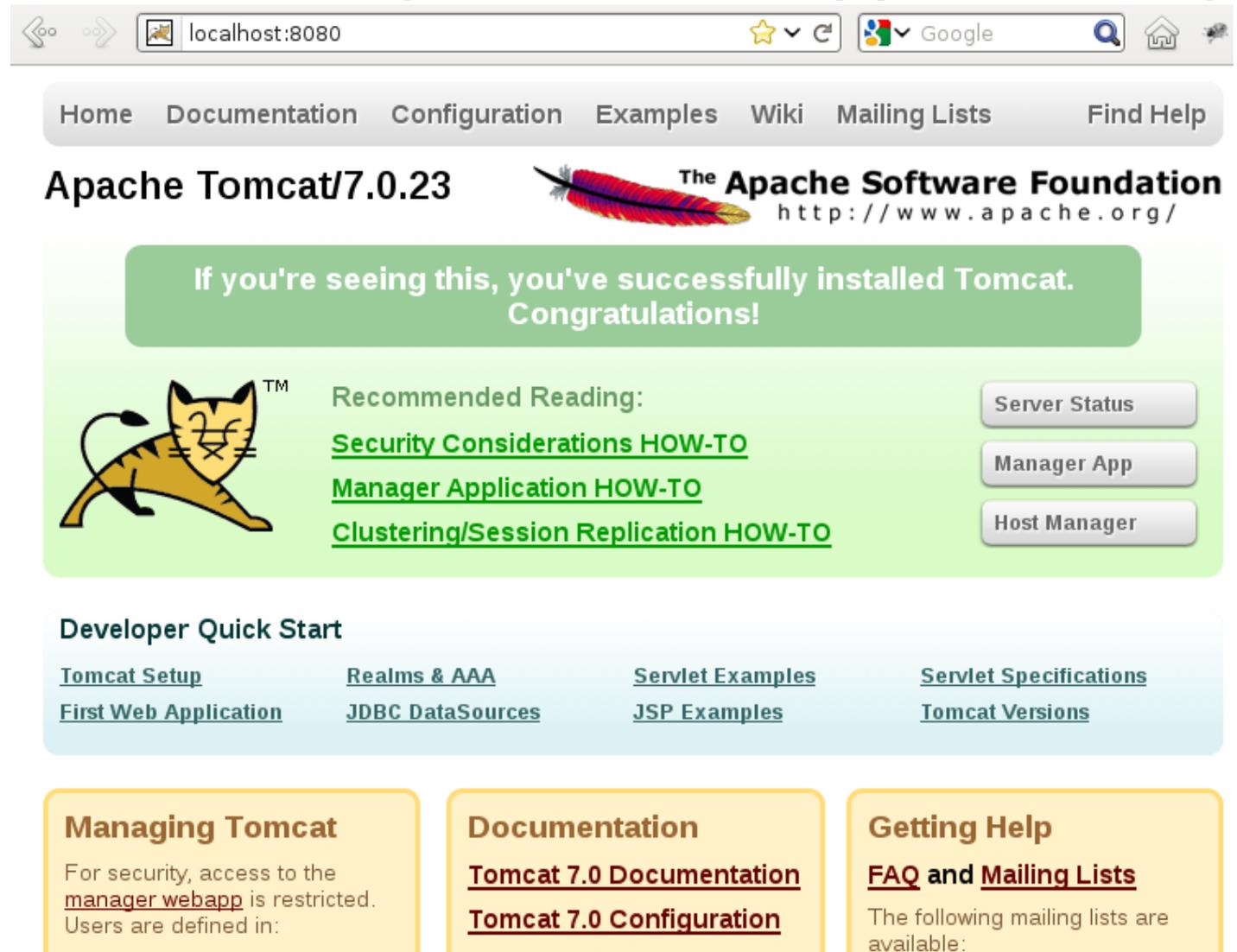


Using Tomcat Web Console

(/manager built-in webapp)

Tomcat Welcome Page

- <http://localhost:8080> (default webapp installed)



The screenshot shows a web browser window with the address bar set to `localhost:8080`. The page title is "Apache Tomcat/7.0.23". The navigation menu includes "Home", "Documentation", "Configuration", "Examples", "Wiki", "Mailing Lists", and "Find Help". The Apache Software Foundation logo and URL (<http://www.apache.org/>) are visible. A green banner reads: "If you're seeing this, you've successfully installed Tomcat. Congratulations!". Below this, there is a cartoon tiger logo and a "Recommended Reading" section with links to "Security Considerations HOW-TO", "Manager Application HOW-TO", and "Clustering/Session Replication HOW-TO". On the right, there are buttons for "Server Status", "Manager App", and "Host Manager". A "Developer Quick Start" section contains links for "Tomcat Setup", "Realms & AAA", "Servlet Examples", "Servlet Specifications", "First Web Application", "JDBC DataSources", "JSP Examples", and "Tomcat Versions". At the bottom, there are three yellow boxes: "Managing Tomcat" (with a note about security and restricted access), "Documentation" (with links to "Tomcat 7.0 Documentation" and "Tomcat 7.0 Configuration"), and "Getting Help" (with links to "FAQ and Mailing Lists" and a note about mailing lists).

Catalina Server Status

- Edit file `conf/tomcat-users.xml`

```
<tomcat-users>
<!-- tomcat 7 -->
<user username="user" password="secr3t"
      roles="manager-gui,manager-script,manager-jmx,manager-status" />
<!-- tomcat6: ... roles="manager" -->
</tomcat-users>
```

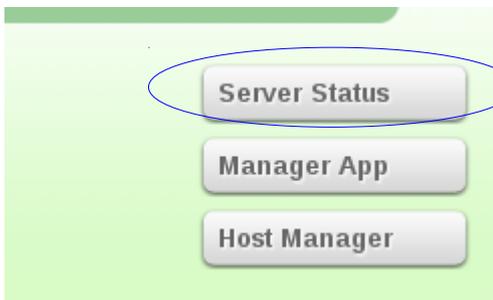
- + restart + click

Server Status

Server Information							
Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture	Hostname	IP Address
Apache Tomcat/7.0.23	1.6.0_18-b18	Sun Microsystems Inc.	Linux	2.6.32-5-amd64	amd64	arn1	127.0.1.1

JVM

Free memory: 93.34 MB Total memory: 116.18 MB Max memory: 1724.50 MB



Tomcat App Manager

Server Status

Manager App

Host Manager



Tomcat Web Application Manager

Message: OK

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	2	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy	
Deploy directory or WAR file located on server	
Context Path (required):	<input type="text"/>
XML Configuration file URL:	<input type="text"/>
WAR or Directory URL:	<input type="text"/>
<input type="button" value="Deploy"/>	
WAR file to deploy	
Select WAR file to upload	<input type="button" value="Choose File"/> No file chosen
<input type="button" value="Deploy"/>	

Http Deploy / Redeploy a WAR

- Click for deploy/undeploy / stop/start / ...

<u>/test-springweb</u>	<i>None specified</i>		true	0	Start Stop Reload Undeploy
					Expire sessions with idle ≥ 30 minutes

3) check new deployed

Deploy

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

Deploy

WAR file to deploy

Select WAR file to upload: **1) select WAR file**

2) deploy

Http Deploy Tools

- Actions handled by “built-in” /manager webapp
- Many tools can do Http POST...
- Sample advanced scripting (using /text, not /html)

```
wget "http://user:pwd@localhost:8080/manager/text/list" -q -O -
```
- Example: Maven-tomcat-plugin
 - Need to configure URL, Username, Password
(see next)

Java coded Http deploy

```
@Test
public void deployWarWithHttpManager() throws Exception {
    String deployUrl = "http://localhost:8080/manager/html/deploy?path=%2Ftest-springweb&update=true";
    File warFile = new File("target/test-springweb.war");

    DefaultHttpClient httpClient = new DefaultHttpClient();
    Credentials creds = new UsernamePasswordCredentials(username, password);
    httpClient.getCredentialsProvider().setCredentials(new AuthScope("localhost", 8080), creds);
    HttpPut httpPut = new HttpPut(deployUrl);
    httpPut.setEntity(new FileEntity(warFile, null));

    HttpResponse response = httpClient.execute(httpPut);

    assertEquals(200, response.getStatusLine().getStatusCode());
    HttpEntity entity = response.getEntity();
    String content = IOUtils.toString(entity.getContent());
    if (content.startsWith("OK - Undeployed")) {
        assertTrue(content.startsWith("OK - Undeployed application at context path /test-springweb"));
    }
    assertTrue(content.endsWith("OK - Deployed application at context path /test-springweb\n"));
}
```

Using Maven Plugin(s)

```
<plugin>  
  <groupId>org.apache.tomcat.maven</groupId>  
  <artifactId>tomcat7-maven-plugin</artifactId>  
</plugin>
```

Maven Http Tomcat Deploy

in pom.xml
Profile for url ...

```
<plugin>
  <groupId>org.apache.tomcat.maven</groupId>
  <artifactId>tomcat7-maven-plugin</artifactId>
  <configuration>
    <url>http://localhost:8080/manager/html</url>
    <server>my-local-tomcat-serverid</server>
    <!-- for user/password see server in settings.xml -->
    <update>true</update>
  </configuration>
</plugin>
```

in ~/.m2/settings.xml
serverId = user-password
match conf/tomcat-users.xml

```
<server>
  <id>my-local-tomcat-serverid</id>
  <username>manager-user</username>
  <password>manager-password</password>
</server>
```

```
$ mvn org.apache.tomcat.maven:tomcat7-maven-plugin:deploy
```

```
[INFO] --- tomcat7-maven-plugin:2.0-SNAPSHOT:deploy (default-cli) @ test-springweb ---
[INFO] Deploying war to http://localhost:8080/test-springweb
Uploading: http://localhost:8080/manager/html/deploy?path=%2Ftest-springweb&update=true
Uploaded: http://localhost:8080/manager/html/deploy?path=%2Ftest-springweb&update=true (4245 KB at 4050.5 KB/sec)
[INFO] tomcatManager status code:200, ReasonPhrase:OK
[INFO] OK - Undeployed application at context path /test-springweb
[INFO] OK - Deployed application at context path /test-springweb
```

Choose your plugins ... and versions

- Historically, Jetty better supported than Tomcat
 - Could run “embedded” in maven, zero-conf
 - Now Possible with tomcat7
- Jetty 6/7 → migration to jetty8
 - Code at mortbay moved to eclipse.org
- Tomcat-plugin → migration to tomcat7-plugin
 - Code at codehaus moved to apache.org

Choose Maven plugins...

```
<plugin>
  <groupId>org.mortbay.jetty</groupId>
  <artifactId>jetty-maven-plugin</artifactId>
</plugin>

<plugin>
  <groupId>org.apache.tomcat.maven</groupId>
  <artifactId>tomcat7-maven-plugin</artifactId>
</plugin>

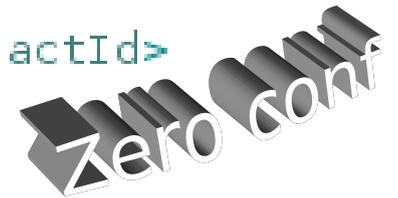
<groupId>org.codehaus.cargo</groupId>
<artifactId>cargo-maven2-plugin</artifactId>
<configuration>
  <container>
    <containerId>tomcat7x</containerId>
  </container>
</configuration>

<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-failsafe-plugin</artifactId>
</plugin>
```

Jetty-Maven-Plugin

- The simplest (and fastest) !

```
<plugin>
  <groupId>org.mortbay.jetty</groupId>
  <artifactId>jetty-maven-plugin</artifactId>
</plugin>
```



```
$ mvn jetty:run
```

```
08:16:03 [main] INFO o.s.w.servlet.DispatcherServlet - FrameworkServlet 'springDispatc
initialization completed in 781 ms
08:16:03 [main] DEBUG o.s.w.servlet.DispatcherServlet - Servlet 'springDispatcherServle
successfully
2012-01-26 08:16:03.025::INFO: Started SelectChannelConnector @ 0.0.0.0:8080
[INFO] Started Jetty Server
[INFO] Starting scanner at interval of 10 seconds.
```

Maven-tomcat-plugin

- <http://tomcat.apache.org/maven-plugin-2.0-SNAPSHOT/index.html>

```
<plugin>
  <groupId>org.apache.tomcat.maven</groupId>
  <artifactId>tomcat7-maven-plugin</artifactId>
  <!-- <version>2.0-SNAPSHOT</version> -->
</plugin>
```



```
$ mvn org.apache.tomcat.maven:tomcat7-maven-plugin:run
...
[INFO] --- tomcat7-maven-plugin:2.0-SNAPSHOT:run (default-cli) @ test-springweb -
[INFO] Running war on http://localhost:8080/test-springweb
[INFO] Using existing Tomcat server configuration at /mnt/a_ltera-1/homeData/arn:
...
INFO: Starting ProtocolHandler ["http-bio-8080"]
```

Notes on maven plugins commands

- `mvn "phase"` ==> perform lifecycle phase
- `!= mvn "plugin:goal"` ==> perform 1 plugin goal
alias for `mvn "groupId:artifactId:version:goal"`
- **Implicit groupId** ==> cf settings.xml `<pluginGroups>`

```
<pluginGroups>
  <!-- implicit
  <pluginGroup>org.apache.maven.plugins</pluginGroup>
  <pluginGroup>org.codehaus.mojo</pluginGroup>
  -->
  <pluginGroup>org.apache.tomcat.maven</pluginGroup>
</pluginGroups>
```

- **Implicit artifactId** ==> accept
"`{prefix}-maven-plugin`" and "`maven-{prefix}-plugin`"

Notes on maven Profiles activation

- To switch between tomcat6 / tomcat7 / jetty ...
- To activate / disable integration-test ...
- => Use profiles in pom.xml (+settings.xml)

```
<profiles>
  <profile>
    <id>integration-test</id>
    <build>
      <plugins>
        <plugin>
          <groupId>org.apache.maven.plugins</groupId>
          <artifactId>maven-failsafe-plugin</artifactId>
```

- Activate with "mvn -Pprofile" ...

Cargo Maven plugins

```
<groupId>org.codehaus.cargo</groupId>
<artifactId>cargo-maven2-plugin</artifactId>
<configuration>
  <container>
    <containerId>tomcat7x</containerId>
  </container>
</configuration>
```

Containers

- Geronimo
 - Geronimo 1.x
 - Geronimo 2.x
- Glassfish
 - Glassfish 2.x
 - Glassfish 3.x
- JBoss
 - JBoss 3.x
 - JBoss 4.x
 - JBoss 4.2.x
 - JBoss 5.x
 - JBoss 5.1.x
 - JBoss 6.x
 - JBoss 6.1.x
 - JBoss 7.x
 - JBoss 7.1.x
- Jetty
 - Jetty 4.x
 - Jetty 5.x
 - Jetty 6.x
 - Jetty 7.x
 - Jetty 8.x
- Jol
 - jo 1.x
- JOnAS
 - JOnAS 4.x
 - JOnAS 5.x
- JRun
 - JRun 4.x
- Orion/OC4J
 - Oc4J 9.x
 - Oc4j 10.x
- Resin
 - Resin 2.x
 - Resin 3.x
 - Resin 3.1.x
- Tomcat
 - Tomcat 4.x
 - Tomcat 5.x
 - Tomcat 6.x
 - Tomcat 7.x
- WebLogic
 - WebLogic 8.x
 - WebLogic 9.x
 - WebLogic 10.x
 - WebLogic 10.3.x

Cargo

- Game of the name **“Cargo”** : handle **“Container”**
- Many supported Containers
 - Tomcat, Jetty, Weblogic, Glassfish, Jboss, ...
- Cargo can
 - Download tomcat.zip
 - + configure
 - + start tomcat
 - + deploy (1 or several) WAR on tomcat
 - + stop tomcat

Cargo-maven2-plugin

```
<plugin>  
  <groupId>org.codehaus.cargo</groupId>  
  <artifactId>cargo-maven2-plugin</artifactId>  
  <version>1.2.0</version>
```

More config

```
$ mvn -Ptomcat-7 org.codehaus.cargo:cargo-maven2-plugin:run
```

```
[INFO] --- cargo-maven2-plugin:1.2.0:run (default-cli) @ test-springweb ---  
[INFO] Press Ctrl-C to stop the container...
```

Maven Cargo settings

```
<configuration>
  <!-- Container configuration -->
  <container>
    <containerId>tomcat7x</containerId>
    <zipUrlInstaller><url>http://archive.apache.org/dist/tomcat/tomcat-7/v7
    <home>/opt/devtools/tomcat/apache-tomcat-7.0.23</home>
    <output>${project.build.directory}/tomcat7x-container.log</output>
    <append>>false</append>
    <log>${project.build.directory}/tomcat7x-cargo.log</log>
  </container>
  <!-- Configuration to use with the container or the deployer -->
  <configuration>
    <home>${project.build.directory}/tomcat7x</home>
    <properties>
      <cargo.servlet.port>8080</cargo.servlet.port>
    </properties>
    <deployables>
      <deployable>
        <groupId>${project.groupId}</groupId>
        <artifactId>${project.artifactId}</artifactId>
        <type>war</type>
        <properties>
          <context>/test-springweb</context>
        </properties>
      </deployable>
    </deployables>
  </configuration>
</configuration>
```

Maven-Failsafe-Plugin

Automated start / stop ...
before/after Integration-tests

```
<plugin>  
  <groupId>org.apache.maven.plugins</groupId>  
  <artifactId>maven-failsafe-plugin</artifactId>  
</plugin>
```

Maven FailSafe plugin

- Failsafe = Junit runner for “Integration Test“

<http://maven.apache.org/plugins/maven-failsafe-plugin/index.html>

- Integration Tests
 - != Unit Tests (surefire)
 - They depends on servers (not mock / jvm)
 - They CAN FAIL without breaking build
- Test are Suffixed “**IT.java**” instead of “Test.java”

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-failsafe-plugin</artifactId>
  <version>2.11</version>
  <executions>
    <execution>
      <id>integration-test</id>
      <goals>
        <goal>integration-test</goal>
      </goals>
    </execution>
    <execution>
      <id>verify</id>
      <goals>
        <goal>verify</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

Sample Http Client Integration Test

```
public class HelloWorldHttpGetIT extends Assert {

    @Test
    public void testHttpGet() throws Exception {
        HttpClient httpClient = new DefaultHttpClient();
        HttpGet httpGet = new HttpGet("http://localhost:8080/test-springweb");
        HttpResponse response = httpClient.execute(httpGet);
        StatusLine statusLine = response.getStatusLine();
        assertEquals(200, statusLine.getStatusCode());
        HttpEntity entity = response.getEntity();
        assertNotNull(entity);
        if (entity != null) {
            InputStream instream = entity.getContent();
            assertNotNull(instream);
            String content = IOUtils.toString(instream);
            assertTrue(content.startsWith("<html>\n"
                + "<body>\n"
                + "<h2>Hello World Index JSP!</h2>\n"));
        }
    }
}
```

Server RUNNING



Server OFF



executions

with pre/post-integration-test

maven-failsafe-plugin

Cargo/jetty/...

Tomcat/jetty

```
<groupId>org.codehaus.cargo</groupId>
<artifactId>cargo-maven2-plugin</artifactId>
<version>1.2.0</version>
<executions>
  <execution>
    <id>start-container</id>
    <phase>pre-integration-test</phase>
    <goals>
      <goal>start</goal>
    </goals>
  </execution>
  <execution>
    <id>stop-container</id>
    <phase>post-integration-test</phase>
    <goals>
      <goal>stop</goal>
    </goals>
  </execution>
</executions>
```

pre-integration-test

integration-test

post-integration-test

start

RUNNING
WebApp

stop

Running Maven Failsafe + Cargo

```
$ mvn -Pintegration-test -Ptomcat-7 install  
$ mvn -Pintegration-test -Ptomcat-6 install
```

pre-integration-test

```
[INFO] --- cargo-maven2-plugin:1.2.0:start (start-container) @ test-springweb ---  
[INFO]  
[INFO] --- maven-failsafe-plugin:2.11:integration-test (integration-test) @ test-springweb  
[INFO] Failsafe report directory: /mnt/a_ltera-1/homeData/arnaud/perso/devPerso/tests/test-
```

start

integration-test

```
.....  
T E S T S  
.....  
Running fr.an.test.HelloWorldHttpGetIT  
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.558 sec  
  
Results :
```

RUNNING
WebApp

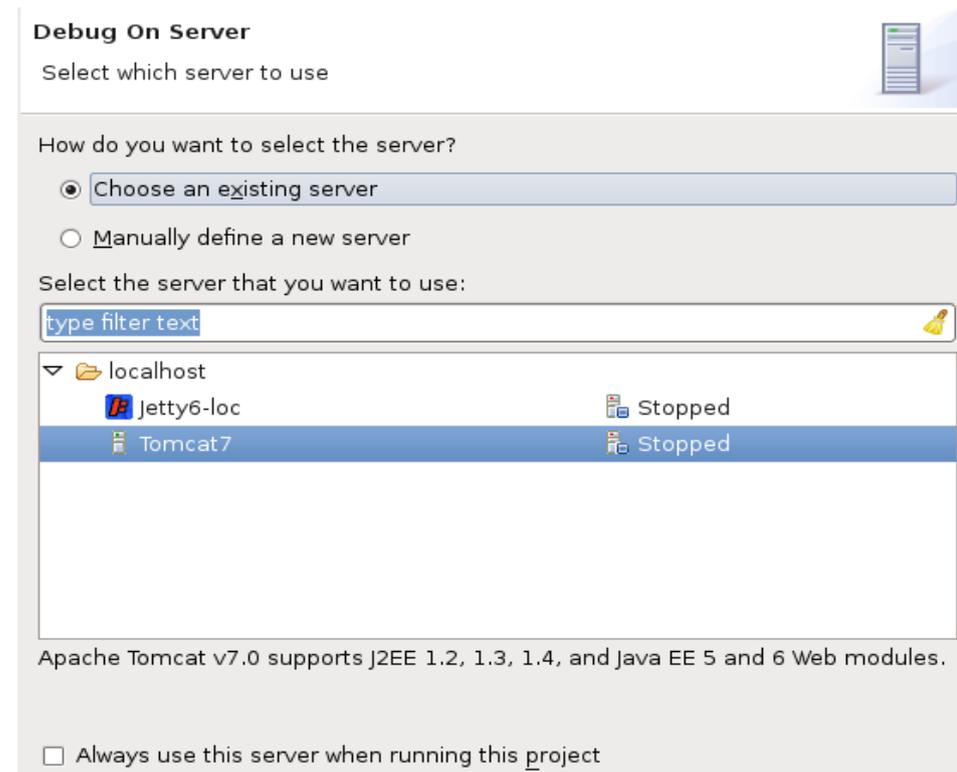
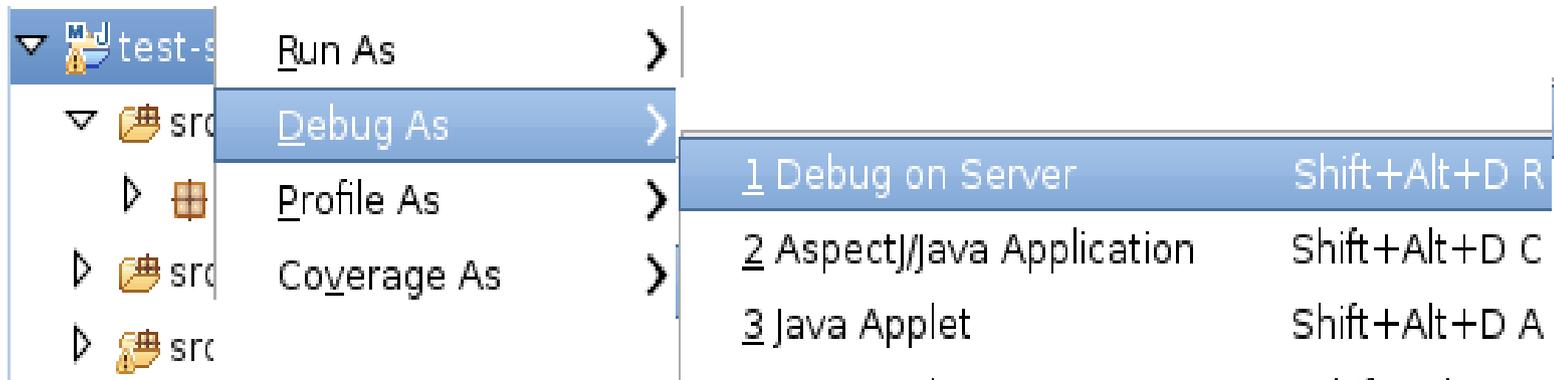
post-integration-test

```
: 1, Failures: 0, Errors: 0, Skipped: 0  
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is plat  
[INFO]  
[INFO] --- cargo-maven2-plugin:1.2.0:stop (stop-container) @ test-springweb ---  
[INFO]  
[INFO] --- maven-failsafe-plugin:2.11:verify (verify) @ test-springweb ---  
[INFO] Failsafe report directory: /mnt/a_ltera-1/homeData/arnaud/perso/devPerso/tests/test-
```

stop

Using Eclipse WTP

Debug As > Debug on Server



New Servers

New server (from existing adapter)

New Server Runtime Environment
Define a new server runtime environment

[Download additional server adapters](#)

Select the type of runtime environment:

type filter text

- Apache
 - Apache Tomcat v3.2
 - Apache Tomcat v4.0
 - Apache Tomcat v4.1
 - Apache Tomcat v5.0
 - Apache Tomcat v5.5
 - Apache Tomcat v6.0
 - Apache Tomcat v7.0**
- BEA Systems, Inc.
- Basic
- IBM
- JBoss
- Jetty

Apache Tomcat v7.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5 and 6 Web modules.

Create a new local server

< Back Next > Cancel Finish

Download additional server adapter

Install New Extension
Download and install support for a new server

The following server adapters have been found on remote sites. Select a server adapter to download and install.

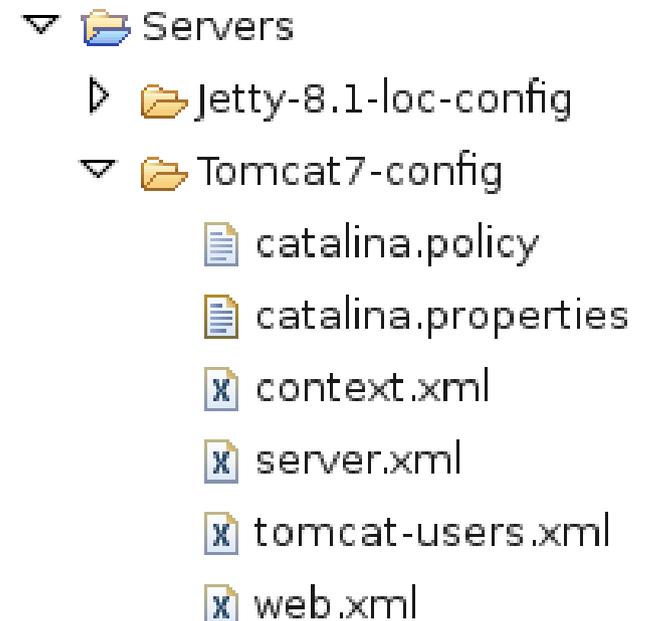
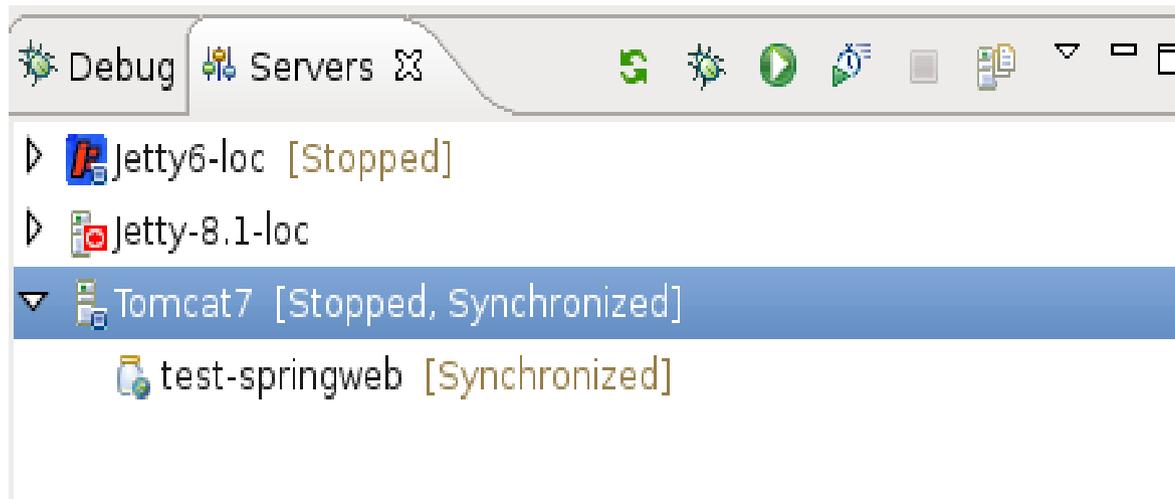
Geronimo v2.2 Server Adapter	Apache Software Foundation	2.1.8
Geronimo v2.2 Server Adapter	Apache Software Foundation	2.2.1
Geronimo v3.0 Server Adapter	Apache Software Foundation	3.0.0
JBossAS Tools	JBoss by Red Hat	2.3.0.v20111213-1423-H61-M5
Jonas WTP Adapter	OW2	5.1.2
Oracle GlassFish Server Tools	Oracle	2.0.0.201111040904
Oracle Public Cloud Tools	Oracle	2.0.0.201112072225
Resin (Java EE Web Profile) Server	Caucho Technology, Inc.	4.0.23

Provides tools for developing Java EE Applications for GlassFish Server Open Source Edition 3, 3.1 and 3.1.1 and Oracle GlassFish Server 3, 3.1 and 3.1.1 (Java EE 6) and GlassFish 2 (Java EE 5).

? < Back Next > Cancel Finish

Server in Eclipse

- Configuration stored in project “Servers”
- Launcher + data hidden in “workspace/.metadata/.plugins/org.eclipse.wst.server.core”



Server Debugging

Debug Servers

Tomcat7 [Apache Tomcat]

org.apache.catalina.startup.Bootstrap at localhost:34374

- Thread [main] (Running)
- Daemon Thread [ContainerBackgroundProcessor[StandardEngine[Catalina]]]
- Daemon Thread [http-bio-8080-AsyncTimeout] (Running)
- Daemon Thread [http-bio-8080-Acceptor-0] (Running)
- Daemon Thread [ajp-bio-8009-Acceptor-0] (Running)

Tomcat7 <http://localhost:8080/test-springweb/> HomeController.java

Overview

General Information

Specify the host name and other common settings.

Server name: Tomcat7

Host name: localhost

Runtime Environment: Tomcat7

Configuration path: /Servers/Tomcat7-config

[Open launch configuration](#)

Server Locations

Specify the server path (i.e. catalina.base) and deploy path. Server must be published with no modules present to make changes.

Use workspace metadata (does not modify Tomcat installation)

- New
- Open F3
- Show In Shift+Alt+W
- Copy Ctrl+C
- Paste Ctrl+V
- Delete Delete
- Rename F2
- Restart in Debug Ctrl+Alt+D
- Restart Ctrl+Alt+R
- Restart in Profile
- Stop Ctrl+Alt+S
- Publish Ctrl+Alt+P
- Clean...
- Add and Remove...
- Monitoring
- Clean Tomcat Work Directory...
- Properties Alt+Enter

Publishing

Timeouts

Ports

Modify the server ports.

Port Name	Port Number
Tomcat admin port	8005
HTTP/1.1	8080
AJP/1.3	8009

MIME Mappings

Eclipse JSP Debugger / Web viewer



The screenshot shows the Eclipse IDE interface with the following components:

- Editor:** `helloWorldView.jsp` with the following code:

```
<%@ page session="false" %>
<%@ taglib prefix="c" uri="http://java.sun.com/" %>
<%@ taglib prefix="fmt" uri="http://java.sun.com/" %>
<html>
<head>
<title>HelloWorldView jsp</title>
</head>
<body>
date: <%= new java.util.Date() %>
<h1>
Hello world View JSP!
</h1>
message: ${serverMessage}
<BR/>
out message: <c:out value="${serverMessage}" />
<BR/>
```
- Variables View:** A table showing the current state of variables in the JSP page.

Name	Value
▶ this	org.apache.jsp.WEB_002dINF.jsp.helloWorldView_jsp (
▶ request	org.apache.catalina.core.ApplicationHttpRequest (id=
▶ response	org.apache.catalina.connector.ResponseFacade (id=3
▶ pageContext	org.apache.jasper.runtime.PageContextImpl (id=395
▶ out	org.apache.jasper.runtime.JspWriterImpl (id=3958)
▶ _jspx_out	org.apache.jasper.runtime.JspWriterImpl (id=3958)
▶ _jspx_page_context	org.apache.jasper.runtime.PageContextImpl (id=395

Eclipse Project Facets



Project Facets

Configuration: <custom> [Save As...] [Delete]

Project Facet	Version
<input type="checkbox"/> Apache XMLBeans	
<input type="checkbox"/> Application Client module	6.0
<input type="checkbox"/> Axis2 Web Services	
<input type="checkbox"/> CXF 2.x Web Services	1.0
<input checked="" type="checkbox"/> Dynamic Web Module	2.5
<input type="checkbox"/> EAR	6.0
<input type="checkbox"/> EJB Module	3.1
<input type="checkbox"/> EJBDoclet (XDoclet)	1.2.3
<input type="checkbox"/> Grails Application	1.0
<input checked="" type="checkbox"/> Java	1.6
<input type="checkbox"/> Java Annotation Processing Support	5.0
<input type="checkbox"/> JavaScript	1.0
<input type="checkbox"/> JavaServer Faces	2.0
<input type="checkbox"/> JAX-RS (REST Web Services)	1.1

Dynamic Web Module 2.5

Adds support for the Java Servlet API, for generation of dynamic Web page content.

Requires the following facet:

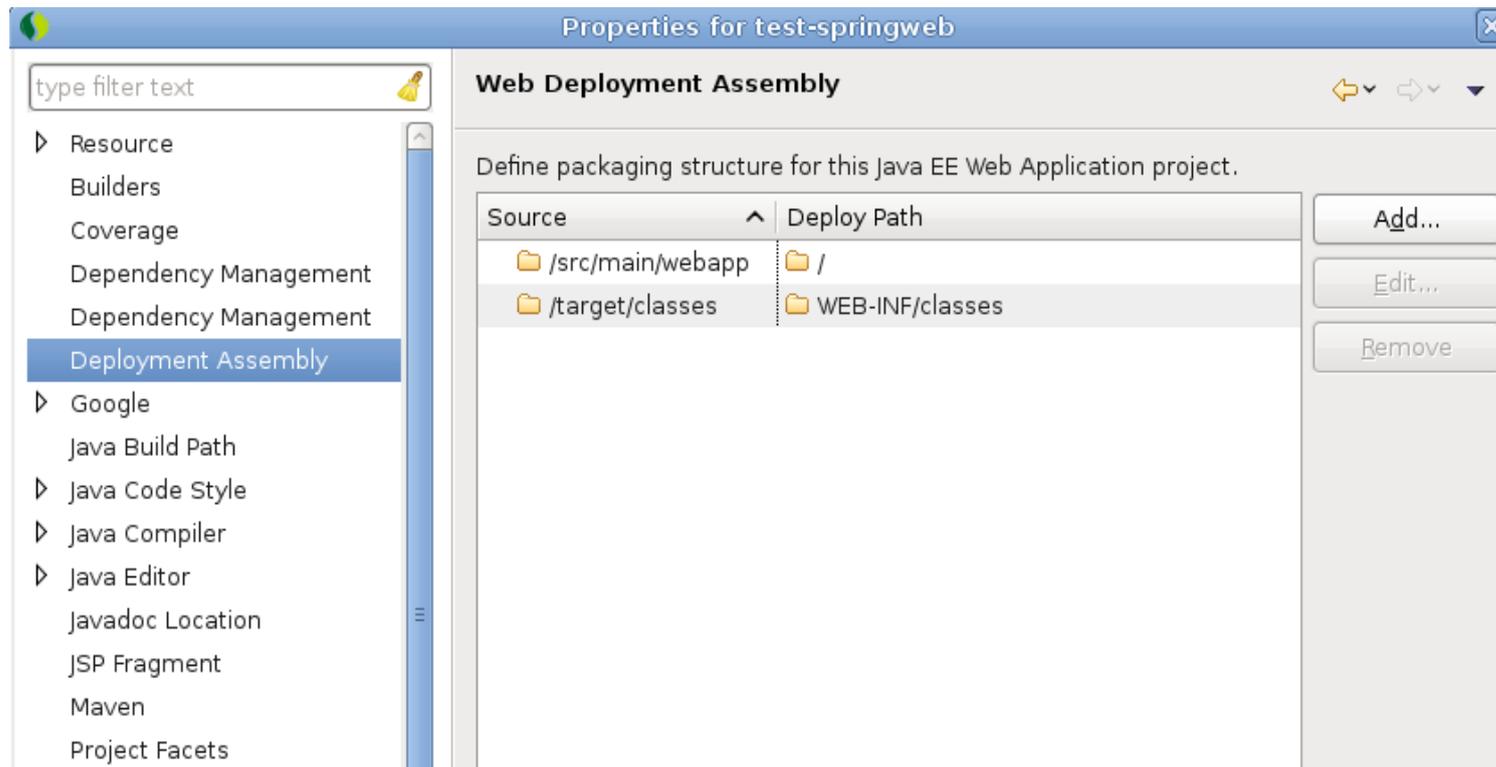
- Java 1.5 or newer

Conflicts with the following facets:

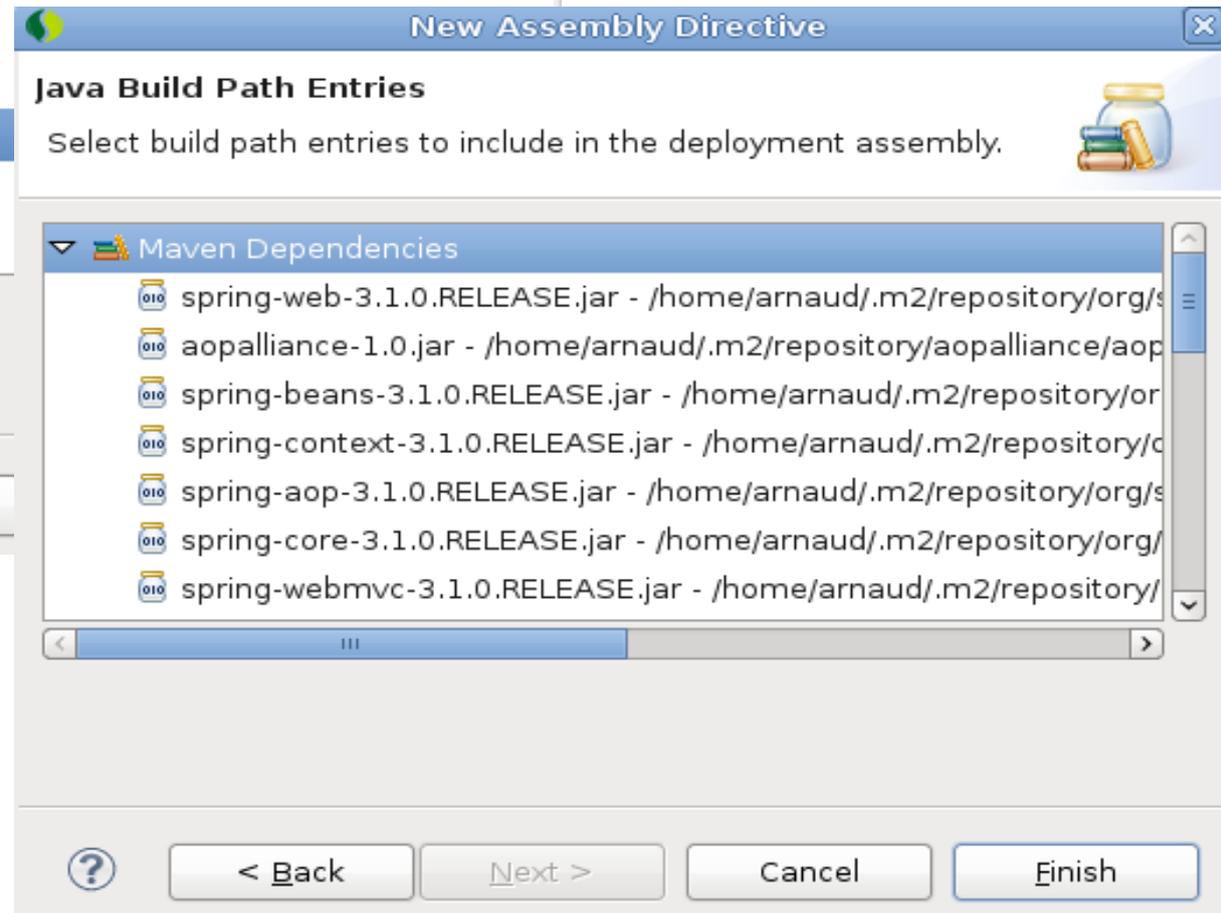
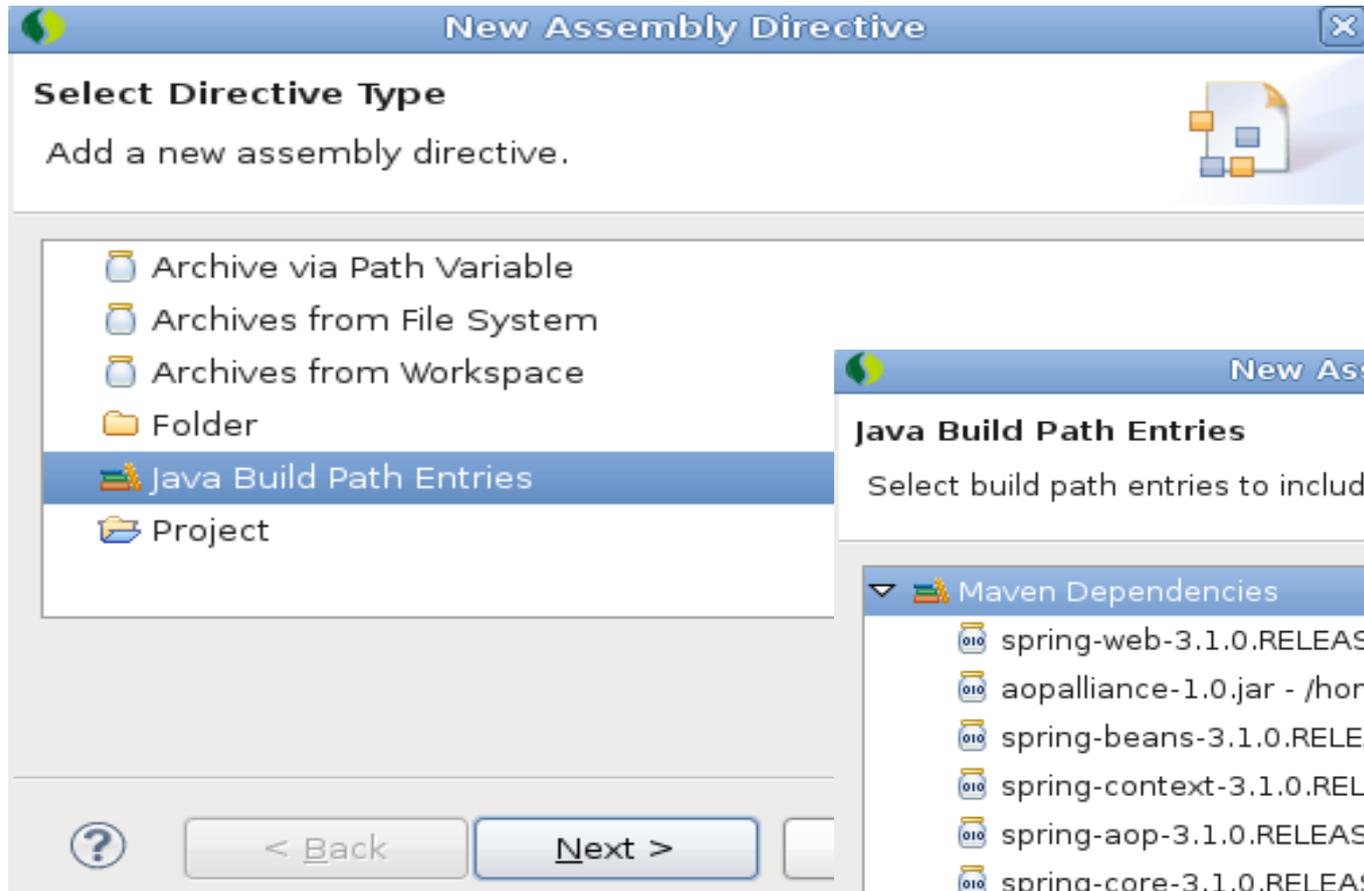
- Application Client module
- Dynamic Web Module
- EAR
- EJB Module
- JCA Module
- Static Web Module
- Utility Module

@^Grbm!#!

Jan 29, 2012 11:42:24 PM org.apache.catalina.core.StandardContext listenerStart
SEVERE: Error configuring application listener of class org.springframework.web.context.ContextLoaderListener
java.lang.ClassNotFoundException: org.springframework.web.context.ContextLoaderListener
at org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1688)
at org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1533)



Fix Deployment



Now OK

- ▶ Resource
- Builders
- Coverage
- Dependency Management
- Dependency Management
- Deployment Assembly
- ▶ Google

Define packaging structure for this Java EE Web Application project.

Source	Deploy Path
/src/main/webapp	/
/target/classes	WEB-INF/classes
/target/test-classes	WEB-INF/classes
Maven Dependencies	WEB-INF/lib

```
23:54:28 [pool-2-thread-1] INFO o.s.w.s.m.m.a.RequestMappingHandlerMapping - Mapped "[/hello],methods=[GET],params=[],headers=[],consumes=[],produces=[application/json]" onto handler 'org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter'
23:54:29 [pool-2-thread-1] INFO o.s.w.s.h.SimpleUrlHandlerMapping - Mapped URL path ["/resources/**"] onto handler 'org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter'
23:54:29 [pool-2-thread-1] INFO o.s.w.servlet.DispatcherServlet - FrameworkServlet 'springDispatcherServlet': initialization completed in 1186 ms
Jan 29, 2012 11:54:29 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-bio-8080"]
Jan 29, 2012 11:54:29 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-bio-8009"]
Jan 29, 2012 11:54:29 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 2891 ms
```

Deployment Context Root

The image shows a screenshot of an IDE's settings dialog for a web project. On the left is a sidebar with a search bar containing 'type filter text' and a list of categories. 'Web Project Settings' is selected and highlighted in blue. The main area of the dialog is titled 'Web Project Settings' and contains a single text input field labeled 'Context root:' with the value 'test-springweb'. At the bottom right of the dialog are two buttons: 'Restore Defaults' and 'Apply'.

type filter text

- Project Facets
- Project References
- Refactoring History
- Run/Debug Settings
- Server
- Service Policies
- Targeted Runtimes
- ▶ Task Repository
- Task Tags
- ▶ Validation
- Web Content Settings
- Web Page Editor
- Web Project Settings**
- WikiText
- ▶ XDoclet

Web Project Settings

Context root: test-springweb

Restore Defaults Apply

Questions ?

arnaud.nauwynck@gmail.com

This document:

<http://arnaud.nauwynck.chez-alice.fr/CoursIUT/10-ways-start-JavaWebApp.pdf>