Embedding Tomcat

Mark Thomas
SpringSource
Agenda

• Introduction
• Background
• A new way to embed
• Single servlet
• Web application
• Differences
• Other features
INTRODUCTION
Introduction

- markt@apache.org
- Tomcat committer & PMC member
- Tomcat 7 release manager
- Member of ASF security team
- Member of ASF infra team
- Staff Engineer at SpringSource
Background

- Tomcat has always been embeddable
- Hasn’t always been easy
Background

* @see org.apache.catalina.startup.Bootstrap#main For a complete example
* of how Tomcat is set up and launched as an Embedded application.

/**
 * Main method, used for testing only.
 * @param args Command line arguments to be processed
 */
public static void main(String[] args) {
    if (daemon == null) {
        // Don't set daemon until init() has completed
        Bootstrap bootstrap = new Bootstrap();
        try {
            bootstrap.init();
        } catch (Throwable t) {
            handleThrowable(t);
            t.printStackTrace();
            return;
        }
    ...
}
Background

• That ‘example’ doesn’t help very much

• Leaves a lot of questions unanswered
  – Which JARs?
  – File system layout?
  – Is all of that code necessary?
A NEW WAY TO EMBED
A new way to embed

- Costin has been working on ‘Tomcat-Lite’
- Costin wrote a new wrapper: org.apache.catalina.startup.Tomcat
- Much simpler to use
- Added this wrapper to Tomcat 7
- Use it in many of the unit tests
A new way to embed

- Added to build targets to generate JARs suitable for embedding
- Next 7.0.x release will include the embedded JARs
  - Need to modify the build script
- For now:
  
  `svn co https://svn.apache.org/repos/asf/tomcat/trunk`  
  `ant embed-release`  
  `cd output/embed`
A new way to embed

- tomcat-embed-core.jar
  - required
- tomcat-embed-logging-juli.jar
  - required
- tomcat-dbcp.jar
  - optional, provides connection pooling
A new way to embed

- **tomcat-jasper.jar**
  - optional, provides JSP support

- **ecj-3.6.jar**
  - optional, provides JSP support
SINGLE SERVLET
Single servlet

• JARs on classpath
  – tomcat-embed-core.jar
  – tomcat-embed-logging-juli.jar

• Single class
  – org.apache.markt.HelloWorld
Single servlet

// Create a Tomcat instance
Tomcat tomcat = new Tomcat();

// Set the port
tomcat.setPort(8080);
Single servlet

// Create a context
File docBase = new File(System.getProperty("java.io.tmpdir"));

Context ctxt = tomcat.addContext("", docBase.getAbsolutePath());
Single servlet

// Add a Servlet
Tomcat.addServlet(ctxt, "HelloWorld", new HelloWorldServlet());

ctxt.addServletMapping("/*", "HelloWorld");
Single servlet

// Start the instance
tomcat.start();

// Loop to serve requests
while(true) {
    Thread.sleep(5000);
}

---

ApacheCon

Leading the Wave of Open Source
WEB APPLICATION
Web application

• JARs on classpath
  – tomcat-embed-core.jar
  – tomcat-embed-logging-juli.jar
  – tomcat-embed-jasper.jar
  – ecj-3.6.jar

• Single class
  – org.apache.markt.WebApp
Web application

// Create a Tomcat instance
Tomcat tomcat = new Tomcat();

// Set the port
tomcat.setPort(8080);
Web application

// Create a context
File docBase = new File(
    "/Users/mthomas/repos/asf-public/" +
    "tomcat/trunk/output/build/" +
    "webapps/examples");

tomcat.addWebapp(null, "",
    docBase.getAbsolutePath());
Web application

// Start the instance
tomcat.start();

// Loop to serve requests
while (true) {
    Thread.sleep(5000);
}

Leading the Wave of Open Source
WEB APPLICATION DEMO
Differences
Differences

- Tomcat.addContext(...)  
  - Completely programmatic
  - You must do everything
  - No default servlet
  - No JSP servlet
  - No welcome files
  - No web.xml parsing

- If you don’t code it, it doesn’t happen
Differences

- Tomcat.addWebapp(...)  
  - Emulates $CATALINA_BASE/conf/web.xml  
  - Default servlet present  
  - JSP servlet present  
  - Welcome files configured  
  - MIME mappings configured  
  - WEB-INF/web.xml parsed
Realms

MapRealm realm = new MapRealm();

realm.addUser("tomcat", "tomcat");
realm.addUserRole("tomcat", "tomcat");
ctx.setRealm(realm);
Login configuration

LoginConfig config = new LoginConfig();
config.setAuthMethod("BASIC");
ctx.setLoginConfig(config);
ctx.getPipeline().addValve(new BasicAuthenticator());
Connector properties

```java
Tomcat tomcat = getTomcatInstance();

tomcat.getConnector().setProperty("maxKeepAliveRequests", "5");
tomcat.getConnector().setProperty("soTimeout", "20000");
tomcat.getConnector().setProperty("keepAliveTimeout", "50000");
```
Servlet configuration

```java
Bug49567Servlet servlet = new Bug49567Servlet();

Wrapper wrapper = Tomcat.addServlet(ctx, "servlet", servlet);
wrapper.setAsyncSupported(true);
ctx.addServletMapping("/", "servlet");
```
Further examples

- Tomcat source code
- Unit tests
- All previous examples taken from here
- Lots of other examples
- CI means they will always be working examples
QUESTIONS