Apache TomEE

Tomcat with a kick

David Blevins
dblevins@apache.org
@dblevins

Jonathan Gallimore
jgallimore@apache.org
@jongallimore

Apache TomEE: Overview

- Java EE 6 Web Profile certification in progress
- Apache TomEE Includes support for:
    - Servlet 3.0 (Apache Tomcat)
    - JPA 2.0 (Apache OpenJPA)
    - JSF 2.0 (Apache MyFaces)
    - CDI 1.0 (Apache OpenWebBeans)
    - EJB 3.1 (Apache OpenEJB)
    - JMS (Apache ActiveMQ)
    - Webservices (Apache CXF)

Apache TomEE: Philosophy

- Preserve Tomcat
  - Leverage Tomcat JNDI, Security, everything
- Get more, don't give up anything
  - Add the extras without removing anything
- Lightweight
  - Just 38MB zip
  - Runs with no extra memory requirements
- No need to learn new server environment
- Existing IDE tools for Tomcat should also work with TomEE

Apache TomEE: History

- Predates Java EE 6 Web Profile
- Integration approach inspired by OpenEJB's embeddable nature
- Previously known as OpenEJB-Tomcat integration
- Also available as a drop-in-war for Tomcat 5.5 - 7.0
- Origin of EE 6 "EJBs in .wars" feature

Web Profile Certification Status

- We can't say (thats the law)
- Work being done on Amazon EC2, using
- T1.Micro Linux images, lots of them
- 100 going at once!
- Each has 613MB memory max
- Though we still run with default memory options (64MB)
- It's quick!
- Will be Cloud certified!
- Wish we could show you the setup (sorry, also the rules)

Apache TomEE: Getting started

- Very simple installation:
  - Obtain bundle and unzip OR
  - Deploy openejb.war into an existing Tomcat installation and run through a simple wizard
- Sample application included in bundle
- Additional lib folder in $CATALINA_BASE/webapps/openejb/lib
- Dashboard available at http://localhost:8080/openejb

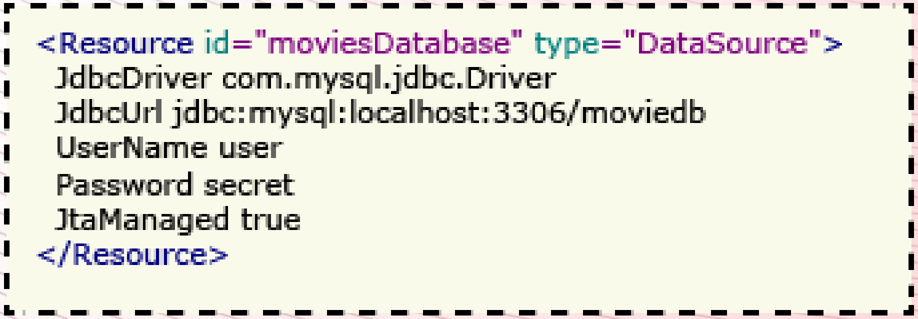
Apache TomEE: Moviefun example

- Movie fun sample – available from OpenEJB SVN repository:

  http://svn.apache.org/repos/asf/openejb/trunk/openejb3/examples/webapps/moviefun/
- Simple example demonstrating:
  - Servlet @EJB injection
  - Bean lookup from a JSP
  - Dependency injection with JSF ManagedBean
  - Persistence with JPA 2
  - Notifications via JMS
  - Webservice support
  - Stateless EJB with no-interface view

Apache TomEE: Configuration

- Resources configured in $CATALINA_BASE/conf/openejb.xml
- Simple XML + properties format
- Tags match Annotation names
- «Resource id="moviesDatabase" type="DataSource"»
- @Resource DataSource moviesDatabase
- Drop drivers in tomcat.home/lib



```
<Resource id="moviesDatabase" type="DataSource">
  JdbcDriver com.mysql.jdbc.Driver
  JdbcUrl jdbc:mysql:localhost:3306/moviedb
  UserName user
  Password secret
  JtaManaged true
</Resource>
```

ApacheTomEE: Security

- Uses Tomcat's Security Realm
- Extra TomEE layer adds support for
  - JAAS
  - JACC
  - WS Security
- Supports any org.apache.catalina.Realm implementation
- E.g. add users to $CATALINA_BASE/conf/tomcat-users.xml
- Alternatively use login.config to provide your own security module

Apache TomEE: JPA

- All persistence.xml files found and deployed
- Connection persistence.xml values filled in automatically
  - jta-data-source
  - non-jta-data-source
- Servlets, ManagedBeans, Session Beans, etc now use
  - @PersistenceUnit EntityManagerFactory emf;
  - @PersistenceContext EntityManager em;
- OpenJPA included, easy to remove
- Hibernate used in many Apache TomEE installs
- EclipseLink also works

Apache TomEE: Transactions

- Connection pooling is Transaction aware
  - Everyone in same transaction, shares same connection
- Servlets, ManagedBeans, etc. can start transactions
  - @Resource UserTransaction transaction;
- No need for EJBs to use transactions

Apache TomEE: JMS

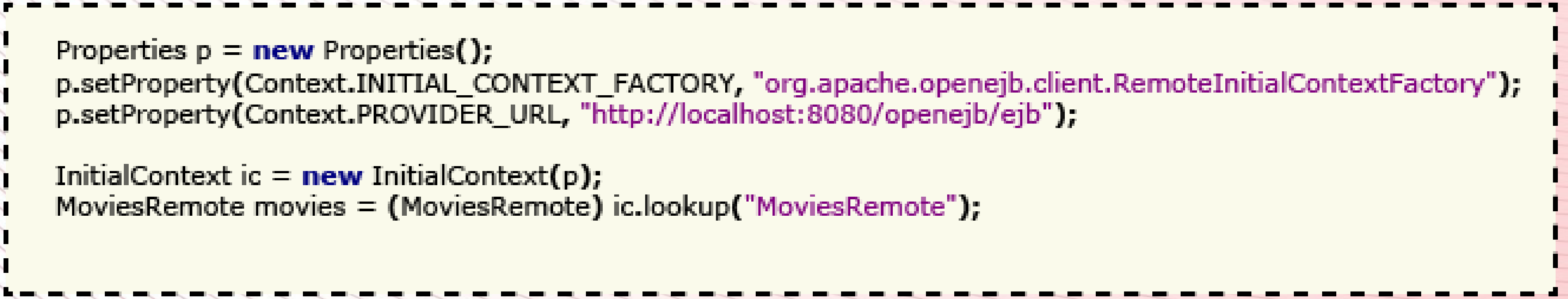
- Resources configured in $CATALINA_BASE/conf/openejb.xml (Connection Factory, Topics, Queues)
- By default ActiveMQ listens for connections on port 61616
- Resources can be injected into Servlets etc via
  - @Resource ConnectionFactory connectionFactory;
  - @Resource Queue queue;
  - @Resource Topic topic;
- Example functionality in the Moviefun / monitor example

Apache TomEE: Web Services

- Use @Webservice on either
  - POJO
  - Session Bean
- CXF & Web Service plumbing happens automatically
- Accessible via: http://localhost:8080/Movies?wsdl
- Web Service Security supported
- Web Service Clients also work
  - @WebServiceRef CalculatorWs service;
- Sample webservice client: get-movies.pl and delete-movie.pl

Apache TomEE: EJB Client

- HTTP can be used for EJB remote communication (http://localhost:8080/openejb/ejb)
- Great for firewalled systems
- User/Pass supplied with InitialContext params
- HTTP or HTTPS



```
Properties p = new Properties();
p.setProperty(Context.INITIAL_CONTEXT_FACTORY, "org.apache.openejb.client.RemoteInitialContextFactory");
p.setProperty(Context.PROVIDER_URL, "http://localhost:8080/openejb/ejb");

InitialContext ic = new InitialContext(p);
MoviesRemote movies = (MoviesRemote) ic.lookup("MoviesRemote");
```

Apache TomEE: Functional Testing

- T7MP Maven plugin can use TomEE as a server
  - Run your application embedded in your Maven build
  - Provides an isolated environment
  - Very useful for functional tests, combined with HtmlUnit or Selenium
  - Example integration test in the moviefun project

Thanks!

http://openejb.apache.org

users@openejb.apache.org
dev@openejb.apache.org