jax 2011 LONDON Autumn Edition

Jonathan Gallimore @jongallimore #TomEE

Apache TomEE // JavaEE Web Profile on Tomcat

What is Apache TomEE

- Tomcat + Java EE = TomEE
- Complete Tomcat with nothing taken away
- Java EE 6 Web profile certified stack
- All Apache components
  - MyFaces
  - OpenEJB
  - OpenWebBeans
  - OpenJPA

Philosophy

- Be small, be certified, be Tomcat
- Integrate OpenEJB into Tomcat, not the other way around
- Get more, don’t give up anything
- Leverage existing Tomcat features (JNDI, Security)
- Lightweight
  - just 24MB
  - no extra memory requirement
- No need to learn new server environment
- Use existing IDE tools

Flavours of TomEE

- Apache TomEE Web Profile (Java EE 6 certified)
  - OpenEJB
  - OpenWebBeans
  - OpenJPA
  - Bean validation
  - MyFaces
- Apache TomEE Plus (not Java EE 6 certified)
  - CXF
  - ActiveMQ
  - Geronimo connector
- Embedded Apache TomEE

Stats

- Size: the entire Web Profile is only 24Mb
- Memory usage: very little required, passed the TCK without changing any default JVM memory settings
- Agility: the server can go through a start/deploy/test/undeploy/stop cycle in 2-3 seconds, when run in embedded mode

Certification

- Certified on Amazon EC2
- t1.micro linux images, lot’s of them
- 100 spot instances going at once (sometimes more)!
- Each has 613MB memory max
- Use default JVM memory settings
- Current certified OSs
  - Amazon Linux AMI 2011.09, EBS boot, 32-bit EC2 t1.micro
  - Amazon Linux AMI 2011.09, EBS boot, 32-bit EC2 m1.small
  - Amazon Linux AMI 2011.09, EBS boot, 32-bit EC2 c1.medium

Getting started

- Unzip
- Start with usual Catalina scripts or services
- Deploy application .war or .ear in webapps
- Console application at http://localhost:8080/openejb

- Or deploy openejb.war in any standard Tomcat 7 installation

Demo

- Moviefun sample - demonstrates:
  - Servlets
  - CDI
  - JSF
  - EJBs
  - JPA
- Available in OpenEJB source code repository
- Over 50 different examples available for various Java EE features

Why not just use Tomcat?

- Plain Tomcat doesn’t have support for:
  - Transactions
  - Transaction aware connection pooling
  - @Resource
  - @PersistenceUnit
  - @Inject
  - @EJB
  - Global JNDI (java:module, java:app, java:global)
- Are you adding libraries to provide EE features (e.g. persistence or CDI)?

Configuration

- Resources configured in conf/openejb.xml
- Simple XML + properties format
- Tags match Annotation names
  - <Resource id="moviesDatabase" type="DataSource">
- Can be injected using this code
  - @Resource DataSource moviesDatabase

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

Persistence

- All persistence.xml files found and deployed
- Connection persistence.xml values filled in automatically
- 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

EJBs

- EJBs can be added directly to webapp
- Servlets, Managed beans can use @EJB
- EJB 3.1 provides no-interface view
- HTTP can be used for EJB remote communication (http://localhost:8080/openejb/ejb)
- User/Pass supplied with InitialContext params
- HTTP or HTTPS

What next?

- How small can we get?
    - Below 20MB download?
    - 6000 classes loaded, can we get to 2000?
- Optimisations
- Certify more parts
    - Connector
    - JAX-WS
    - JAX-RS

Testing - Embedded TomEE

- Very straightforward to use, very little config needed
- Maven users can add tomee-embedded dependency, or add all-in-one jar to classpath
- Could be used with frameworks like HtmlUnit or Selenium to test your application

```
Properties p = new Properties();
p.setProperty(EJBContainer.APP_NAME, "moviefun");
p.setProperty(EJBContainer.PROVIDER, "tomee-embedded"); // need web feature
p.setProperty(EJBContainer.MODULES, webApp.getAbsolutePath());
p.setProperty(EmbeddedTomEEContainer.TOMEE_EJBCONTAINER_HTTP_PORT, "9999");
container = EJBContainer.createEJBContainer(p);
```

Arquillian tests

- Test harness from JBoss (http://www.jboss.org/arquillian)
- Allows you to run your test in any supported container
- Adapters available for a number of servers
- Skip the build
- Run tests straight from the IDE

TomEE Arquillian Adapters

- Embedded
  - Boots TomEE embedded directly in the test
- Remote
  - Can connect to a running instance of TomEE
  - OR download and start TomEE bundle
  - OR any version of Tomcat/OpenEJB
  - Configured in arquillian.xml

Anatomy of an Arquillian Test

- Deployment
  - Build an archive using the ShrinkWrap API
  - Choose parts of your application to test
  - Build archive with test settings
- Test logic
  - Exercise your application
  - HtmlUnit
  - Selenium

Questions?

Thanks!

http://openejb.apache.org
users@openejb.apache.org
dev@openejb.apache.org

@ApacheTomEE
#TomEE