Fun with EJB and OpenEJB

David Blevins
@dblevins
#OpenEJB

Apache
TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

Leading the Wave of Open Source

The Basics - History

- Timeline
  - 1999 - Founded in Exoffice - EJB 1.1 level
  - 2001 - Integrated in Apple's WebObjects
  - 2002 - Moved to SourceForge
  - 2003 - Integrated in Apache Geronimo
  - 2004 - Moved to Codehaus
  - 2006 - Moved to Apache Incubator
  - 2007 - Graduated Apache OpenEJB
- Specification involvement
  - EJB 2.1 (Monson-Haefel)
  - EJB 3.0 (Blevins)
  - EJB 3.1 (Blevins)
  - EJB 3.2 (Blevins)

Apache
TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

Focuses since inception

- Always an Embeddable EJB Container
  - Good idea for Embeddable Databases, good idea for us
  - Our downfall in early 2000 -- people were not ready
  - Our success after EJB 3.0
- No love for traditional Application Servers
  - Don't give up **`main(String[] args)`**
- Always doing the Opposite
  - Instead of putting the Application in the Container, put the Container in the Application
- What do you mean hard to test??
  - Don't blame EJB because your Server is hard to test
  - In what way is mocking not writing an EJB container?

Apache

TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

Leading the Wave of Open Source

Apache

TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

Leading the Wave
of Open Source

We were only pretending to test

EJB Vision & Philosophy

- Misunderstood technology
  - Many things people attribute to “EJB” are not part of EJB
- EJB can be light
  - EJB as a concept is not heavy, implementations were heavy
- EJB can be simpler
  - Though the API was cumbersome it could be improved
- EJB can be used for plain applications
  - The portability concept can be flipped on end
- The flexability applications get also provides great flexability to the container to do things differently yet not break compliance

Apache
TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

Leading the Wave of Open Source

Apache

TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

Leading the Wave
of Open Source

There is no “heavy”
requirement

Apache

TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

Leading the Wave
of Open Source

Show me the heavy

Apache

TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

Leading the Wave
of Open Source

EJB.next and Java EE.next

- Promote @ManagedBean to a Session bean
- Break up EJB -- separate the toppings
  - @TransactionManagement
  - @ConcurrencyManagement
  - @Schedule
  - @RolesAllowed
  - @Asynchronous
- Allow all annotations to be used as meta-annotations
- Modernize the Connector/MDB relationship
- Interceptor improvements
- Balance API
  - Everything that can be turned on should be able to shut off
- Improve @ApplicationException

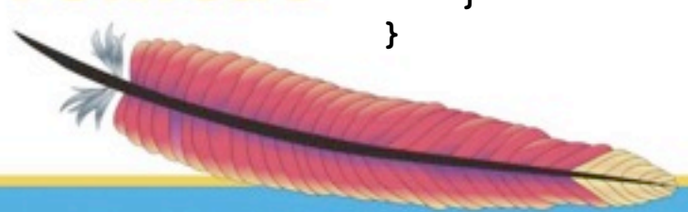
Apache TomEE OpenEJB OpenJPA OpenWB MyFaces Tomcat

Interceptor -- Today

```
    @InterceptorBinding
@Target(value = {ElementType.TYPE})
@Retention(RetentionPolicy.RUNTIME)
public @interface Log {
}

@Log
public class FooBean {

    public void somethingCommon(){
      //...

    public void somethingImportant() {
      //...

    public void somethingNoteworthy() {
      //...
}

@Log
public class LoggingInterceptor {

    private java.util.logging.Logger logger =
            java.util.logging.Logger.getLogger("theLogger");

    @AroundInvoke
    public Object intercept(InvocationContext context) throws Exception {
        logger.info("" + context.getMethod().getName());
        return context.proceed();
    }
}
```

Apache

TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat



Leading the Wave of Open Source

Interceptor Improvements

```
    @Log
public class FooBean {

    public void somethingCommon(){
        //...
    }

    @Info
    public void somethingImportant() {
        //...
    }

    @Fine
    public void somethingNoteworthy() {
        //...
    }
}
```

Apache
TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

Leading the Wave
of Open Source

Interceptor Improvements

```
    @Log
public class LoggingInterceptor {

    private java.util.logging.Logger logger =
            java.util.logging.Logger.getLogger("theLogger");

    @AroundInvoke
    public Object finest(InvocationContext context) throws Exception {
        logger.finest("" + context.getMethod().getName());
        return context.proceed();
    }

    @Info
    public Object info(InvocationContext context) throws Exception {
        logger.info("" + context.getMethod().getName());
        return context.proceed();
    }

    @Fine
    public Object fine(InvocationContext context) throws Exception {
        logger.fine("" + context.getMethod().getName());
        return context.proceed();
    }
}
```

Apache
TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

Leading the Wave of Open Source

Meta-Annotations

```
    @RolesAllowed({"SuperUser", "AccountAdmin", "SystemAdmin"})
@Stereotype
@Target(METHOD)
@Retention(RUNTIME)
public interface Admins {}
```

```
@Schedule(second="0", minute="0", hour="0", month="*", dayOfWeek="*", year="*")
@Stereotype
@Target(METHOD)
@Retention(RUNTIME)
public @interface Hourly {}
```

```
@Schedule(second="0", minute="0", hour="0", month="*", dayOfMonth="15,Last", year="*")
@Stereotype
@Target(METHOD)
@Retention(RUNTIME)
public @interface BiMonthly {}
```

```
@Singleton
@TransactionManagement(CONTAINER)
@TransactionAttribute(REQUIRED)
@ConcurrencyManagement(CONTAINER)
@Lock(READ)
@Interceptors({LoggingInterceptor.class, StatisticsInterceptor.class})
@Stereotype
@Target(TYPE)
@Retention(RUNTIME)
public @interface SuperBean {}
```

Meta-Annotations

```
    @Singleton
@TransactionManagement(CONTAINER)
@TransactionAttribute(REQUIRED)
@ConcurrencyManagement(CONTAINER)
@Lock(READ)
@Interceptors({LoggingInterceptor.class, StatisticsInterceptor.class})
public class MyBean {

    @Schedule(second="0", minute="0", hour="0", month="*", dayOfWeek="*", year="*")
    public void runBatchJob() {
        //...
    }

    @Schedule(second="0", minute="0", hour="0", month="*", dayOfMonth="15,Last", year="*")
    public void sendPaychecks() {
        //...
    }

    @RolesAllowed({"SuperUser", "AccountAdmin", "SystemAdmin"})
    public void deleteAccount(String accountId) {
        //...
    }
}
```

Apache
TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

Leading the Wave of Open Source

Meta-Annotations

```
@SuperBean
public class MyBean {

    @Hourly
    public void runBatchJob() {
        //...
    }

    @BiMonthly
    public void sendPaychecks() {
        //...
    }

    @Admin
    public void deleteAccount(String accountId) {
        //...
    }
}
```

Apache

TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

Leading the Wave
of Open Source

Apache

TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

Leading the Wave
of Open Source

Testing

Embeded / Testing Principles

- Be as invisible as possible
- No special classloaders required
- No files
  - All Configuration can be done in the test or via properties
  - No logging files
  - No database files (in memory db)
- No ports
  - Remote EJB calls done with “intra-vm” server
  - JMS done via embedded broker with local transport
  - Database connections via embedded database
- No JavaAgent
  - Avoidable if not using JPA or if using Hibernate as the provider
  - OpenJPA to a limited extent

Apache
TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

Leading the Wave
of Open Source

What can you test?

- EJBs
  - @Stateless
  - @Stateful
  - @Singleton
  - @MessageDriven
  - @ManagedBean
  - Interceptors
  - Legacy EJB 2.x and earlier
- Views
  - @Remote
  - @Local
  - @LocalBean
  - @WebService (requires a port)

Apache

TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

What can you test? (cont.)

- Container Provided resources
  - DataSources
  - EntityManagers and EntityManagerFactories
  - JMS Topics/Queues
  - WebServiceRefs
  - Any Java EE Connector provided object
- Services
  - Timers
  - Transactions
  - Security
  - Asynchronous methods

Apache
TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

What can't you test?

- ~~Servlets~~
- ~~Filters~~
- ~~Listeners~~
- ~~JSPs~~
- ~~JSF Managed Beans~~
- ~~Non-EJB WebServices~~

Hello, TomEE

Apache
TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

Leading the Wave of Open Source

Unique Testing Features

- Most spec complete embedded container
- Fast startup (1 - 2 seconds)
- Test case injection
- Overriding
  - Configuration overriding
  - Persistence Unit overriding
  - Logging overriding
- Test centric-descriptors
  - test-specific ejb-jar.xml or persistence.xml, etc.
- Validation
  - Compiler-style output of application compliance issues
  - Avoid multiple “fix, recompile, redeploy, fail, repeat" cycles
- Descriptor output -- great for xml overriding

Apache
TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

Leading the Wave of Open Source

Apache

TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

Leading the Wave
of Open Source

Questions?

Apache

TomEE
OpenEJB
OpenJPA
OpenWB
MyFaces
Tomcat

Leading the Wave
of Open Source

thank you!

openejb.apache.org