Clouds, Tenants, Simplicity or just ...Java EE 7

[kill the bloat]





- Expert Group Member (jcp.org) of Java EE 6, Java EE 7, JPA 2.1, EJB 3.2, CDI 1.1, JMS 2.0 (...)
- Java Champion, (JavaONE) speaker + rockstar, freelancer, consultant and author: >100 articles, 7 German books,
- Author: "Real World Java EE Patterns— Rethinking Best Practices" and "Real World Java EE Night Hacks" http://press.adam-bien.com
- NEW: workshops.adam-bien.com
- http://kenai.com/projects/javaee-patterns/

http://java.net/projects/x-ray



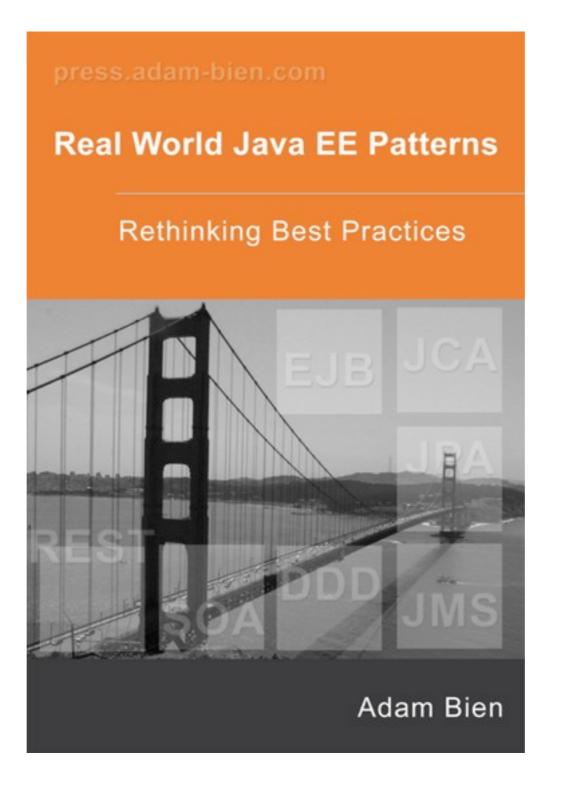


"Perfection (in enterprise development) is achieved not when there is nothing more to add, but rather when there is nothing more to take away."

-- Antoine de Saint-Exupéry



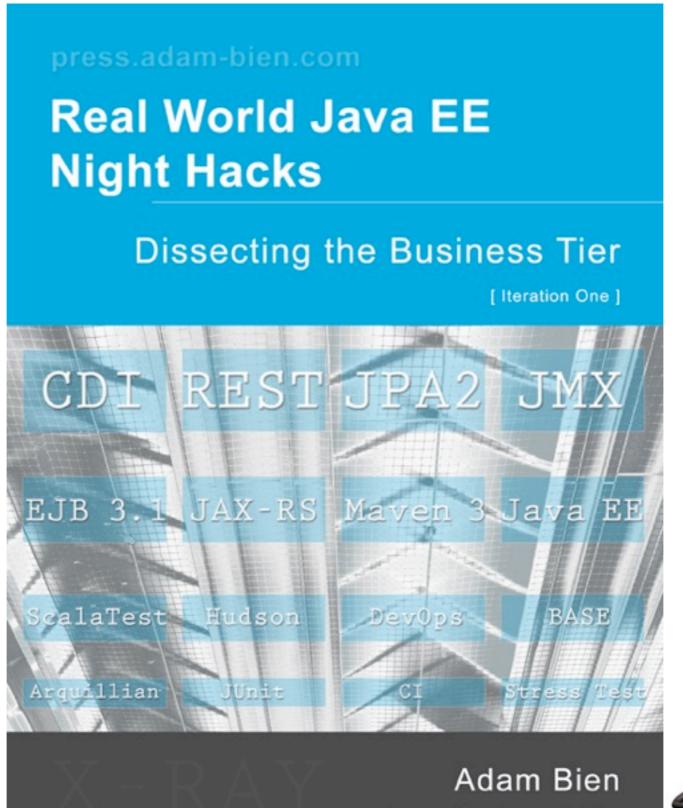




Adam Bien, press.adam-bien.com







Adam Bien, press.adam-bien.com





Foreword by James Gosling

Everything here is subject to change - and probably already outdated :-)

How JSRs / Java EE 7 operates

...The Java EE Crowd...

Java EE 7

Resource Configuration

```
<data-source>
 <name>java:app/jdbc/myOracleDB</name>
 <class-name>....SomeDataSource</class-name>
 <initial-pool-size>5</initial-pool-size>
 <max-pool-size>25</max-pool-size>
 <min-pool-size>10</min-pool-size>
</data-source>
```

JavaEE 7

- Platform As A Service
- Pruning: JSR-77?, CMP, JAX-RPC etc.
- Clouds / Multi-Tenancy
- Auto provisioning of resources
- Staging

New JSRs

- Caching (JSR-107 aka JCache)
- Data Grids (JSR-347)
- Java Identity API (JSR-351)
- Batch (JSR-352)
- JSON
- Concurrency Utilities

EJB 3.2

```
@MessageDriven
public class SimpleMDB {

@SomeAnnotation(tbdAttr="jms/myQueue")
void processMessages(Message message) {
...
}
```

EJB 3.2

- @Asynchronous, Timers in EJB lite
- Reuse of cross cutting aspects
- Further simplifications (MDB)

EJB 3.2

- @MaxConcurrency / @PoolSize
- Transactions in Lifecycle Callbacks
- Asynchronous processing (decided by the client)

- Simplification
- Better / mandatory integration with JCA 1.6
- Seamless integration with other Java EE APIs
- Clarification in the spec / removal of inconsistencies

- Batch sending / receiving
- Last Value Cache
- JMXDeliveryCounter / JMX instrumentation
- Topic Hierarchies

Asynchronous Acknowledge

```
producer.send(message, new
AcknowledgeListener(){
public void onAcknowledge(Message message) {
    // process ack
    }
});
```

- Timed Messages (setDeliveryDelay)
- Direct instantiation of "Factories"
- Direct JNDI-usage

CDI integration

```
@Inject
@JMSConnection(lookup="jms/connFactory")
@JMSDestination(lookup="jms/inboundQueue")
MessageProducer producer;
@Inject
TextMessage textMessage;
public void sendMessageNew(String payload){
   textMessage.setText(payload);
   producer.send(textMessage);
```

• Pruning of:

XAQueueConnection

XAQueueConnectionFactory

XAQueueSession

XATopicConnection

XATopicConnectionFactory

XATopicSession

QueueConnection

QueueConnectionFactory

QueueReceiver

QueueSender

QueueSession

TopicConnection

TopicConnectionFactory

TopicPublisher

TopicSession

TopicSubscriber

- Multi-tenancy
 - Separate Database
 - Shared Database, Separate Schema
 - Sharaed Database, Share Schema
- Downcasting TREAT()

- Support for functions and stored procedures (IN / OUT parameter)
- Improved result mapping for native queries
- Dynamically defined named queries

- CDI injection into event listeners
- Persistence context synchronization
 - @PersistenceContext(type=EXTENDED, synchronization=UNSYNCHRONIZED)
- Dirty detection

JAX-RS 2

JAX-RS 2

- Client API
- Hypermedia, LinkBuilder
- Asynchronous processing (Execution Context)
- Bean Validation Integration

References

- jms-spec.java.net
- java.net/projects/jax-rs-spec
- jpa-spec.java.net
- java.net/projects/ejb-spec
- java.net/projects/javaee-spec/

Thank You!

blog.adam-bien.com twitter.com/AdamBien