JavaOne™

ORACLE®

The JMS BOF

Nigel Deakin
JMS Specification Lead

26 October 2015

nigel.deakin@oracle.com
@jms_spec

Agenda

- "... an informal discussion of the proposed changes in JMS 2.1 and an opportunity to suggest additional changes and new features as well as to discuss how JMS 2.0 went..."

Agenda

1. A look back at JMS 2.0
2. A look forward at JMS 2.1

JMS 2.0

- Released in May 2015

JMS 2.0 in summary: API simplifications

- Simplified API: JMSContext, JMSProducer, JMSConsumer
- Injection of JMSContext objects (Java EE)
- Autocloseable
- Message#getBody
- MessageConsumer#receiveBody and JMSConsumer#receiveBody
- Connection#createSession(int sessionMode)
- Resource definitions (Java EE)
- Default connection factory (Java EE)

JMS 2.0 in summary: new messaging features

- Delivery delay
- Async send
- Shared topic subscriptions
- Delivery count

JMS 2.0 review – my view

- Pooling JMSContext objects more difficult than needed:
- Need a method `createXAJMSContext()` on `XAJMSContext` to allow multiple `XAJMSContexts` to share the same connection.
  - Discovered when updating an app server to support third-party JMS 2.0 providers
- Need standard API to allow application servers to implement pooled `JMSContext` objects on top of an existing connection pool
  - Discovered when updating a generic JMS resource adapter to support JMS 2.0
- Community involvement in the JSR could have been greater

JMS 2.0 review – your view

- Who has used it?
- What do you like most?
- What do you like least?

JMS 2.1 (JSR 368)

Stage	Initial plan (Sep 2014)	Current plan (Updated Jun 2015)	Actual
JSR approval	Sep 2014		Sep 2014
Expert group formation	Q3 (Sep) 2014		Dec 2014
Early draft 1			Oct 2015
Early draft 2	Q1 (Mar) 2015	Q4 (Dec) 2015	
Public review	Q3 (Sep) 2015	Q1 (Mar) 2016	
Proposed final draft	Q4 (Dec) 2015	Q3 (Sep) 2016	
Final release	Q3 (Sep) 2016	H1 (Jun) 2017	

New!

JMS 2.1 proposed content (from the JSR proposal)

- Improved async message delivery in Java EE
  - Flexible JMS MDBs
  - CDI beans as JMS message listeners
  - Message redelivery control and dead message queue
  - Batch delivery
- Removing Java EE restrictions
  - setMessageListener
  - opting out of a JTA transaction
  - client-ack and local transactions

Flexible JMS MDBs

```
@MessageDriven
public class MyDepScopedBean {

  @JMSQueueListener(
    destinationLookup="myQueue")
  public void callMe(Message message) {
    ...
  }


}
```

- Several issues still to resolve:
- Whether to allow multiple callbacks
- How to handle checked exceptions
- How to handle type mismatches
- Show of hands: do you like this so far?

CDI beans as message listeners

```
// @Dependent
public class MyDepScopedBean {

  @JMSQueueListener(
    destinationLookup="myQueue")
  public void callMe(Message message) {
    ...
  }


}
```

- Unclear how this should work
- Use cases needed
- Is this a solution looking for a problem?
- Show of hands: who thinks we need this?

JMS 2.1 proposed content (from the JSR proposal)

- Improved async message delivery in Java EE
  - Flexible JMS MDBs
  - CDI beans as JMS message listeners
  - Message redelivery control and dead message queue
  - Batch delivery
- Removing Java EE restrictions
  - setMessageListener
  - opting out of a JTA transaction
  - client-ack and local transactions

JMS 2.1 proposed content (from the JSR proposal)

- Removing the need to use JNDI (for Java SE applications)
  - API to create connection factory
  - API to create queue and topic objects
- Making use of Java SE 8
  - Repeatable annotations
- Missing methods
  - `createXAJMSContext()` on `XAJMSContext`
  - methods to allow application servers to implement pooled `JMSContext` objects on top of an existing connection pool (e.g. create JMSContext from a Connection)

JMS 2.1 proposed content

- What else?

Getting more involved

- Download the JMS 2.1 early draft 1
- Join the JMS community mailing list – everything is open
- Visit jms-spec.java.net for links to everything
- Follow (and reply to) @jms_spec
- JSR "face to face" meeting on Thursday at 9am
- You don't have to be on the expert group to contribute (but if you're interested in joining the expert group, please get in touch)

Any other comments or questions?

Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Session Surveys

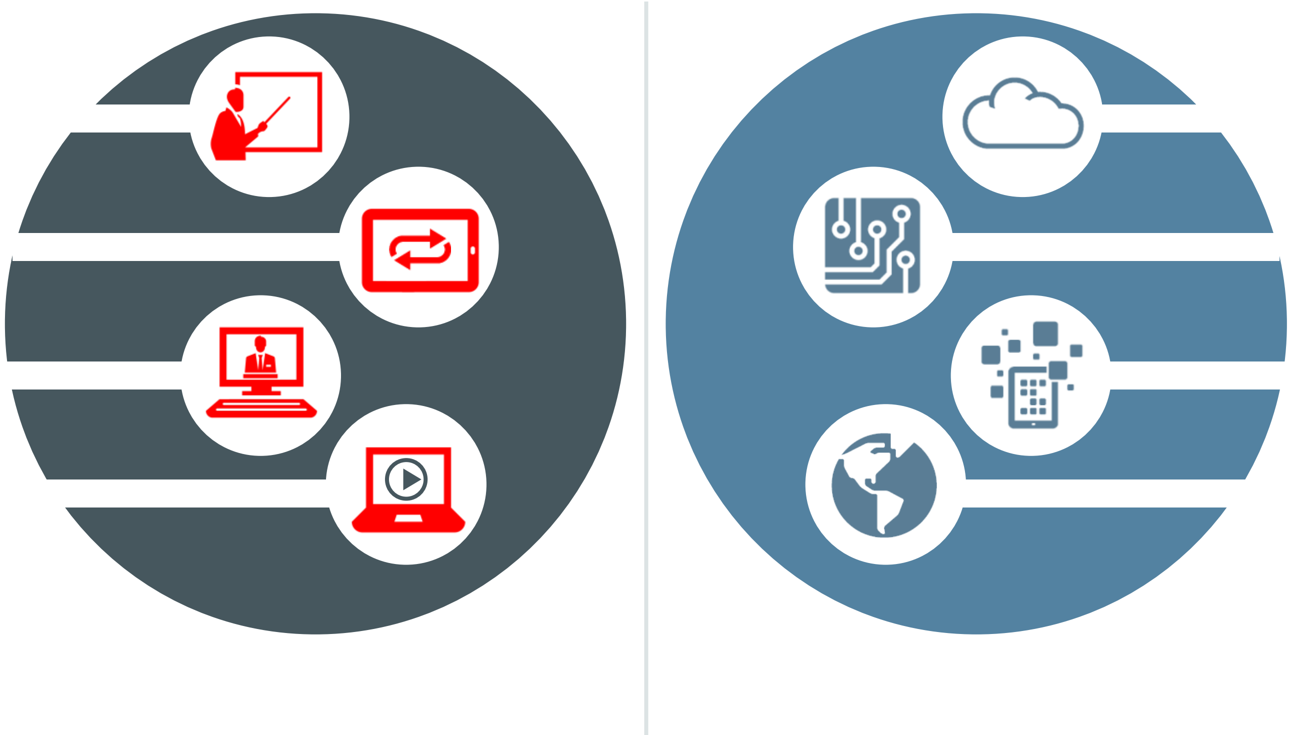
Help us help you!!

- Oracle would like to invite you to take a moment to give us your session feedback. Your feedback will help us to improve your conference.
- Please be sure to add your feedback for your attended sessions by using the Mobile Survey or in Schedule Builder.

Keep Learning with Oracle University

ORACLE

UNIVERSITY



Classroom Training

Learning Subscription

Live Virtual Class

Training On Demand

Cloud

Technology

Applications

Industries

education.oracle.com

JavaOne ORACLE

JavaOne™

ORACLE®