



Declarative Programming: Tighten EJB 3.0 and JSR 303 Beans Validation

Emmanuel Bernard
Core Developer
JBoss, A Division of Red Hat

<http://www.jboss.com>
<http://www.hibernate.org>

TS-4112

Rethink Programming

When pieces **finally** fit together

Metadata + Java™ Platform, Enterprise Edition
(Java EE)

+

POJO validation

=

Don't Repeat Yourself made easy

Agenda

- Java Specification Request (JSR) 303 Beans Validation
 - What's the problem
 - Standardizing the solutions
 - Status
- **Demo**
- EJB™ 3.0 Architecture Interceptors
 - What is it useful for?
 - How to use them
- **Even Better Demo**
 - **Now you've got the picture**

Ensuring Constraints

- Database Schema constraints

```
create table Document (  
    id bigint not null primary key,  
    title varchar(30) not null,  
    summary varchar(255),  
    content varchar(255))
```

- Business/Service level constraints

```
if ( document.getTitle() == null  
    || document.getTitle().length() > 30 ) {  
    throw new BusinessException("Document title"  
        + " invalid");}
```

Ensuring Constraints

- Presentation level constraints

```
if ( documentForm.getTitle() == null
    || documentFor;.getTitle().length() > 30 ) {
    throw new BusinessException("Document title"
        + " invalid");}
```

- Client-side constraints

```
//Your favorite DHTML + javascript
//or
//Your favorite JSF component library
//or
//Your favorite other web framework
```

Domain Model Centric Validation

- Duplication across all the layers
 - Definition
 - Validation
- Centralize definition
- Execute validation automatically
 - Abstract the application logic from it

JSR 303

- Standardize the constraints Metadata Model
 - Primary through annotations
 - XML for complex cases
 - API to query it
- Standardize the validation API
 - Layer agnostic
- Ability to write custom constraints
 - Annotation
 - Validation logic
- i18n

With These API and Metadata...

- DDL update when generated by ORMs
 - e.g., Hibernate Validator and Hibernate Core integration
- Entity validation on insertion/update by Java Persistence API
- WebBeans (Seam) validation
 - Business level
 - Presentation level
- JavaServer™ Faces technology validation at the client side from Metadata
- Your code triggering validation



DEMO

Validation of a Domain Model



Possible Validations

- Logic
 - @NotNull
 - @NotEmpty
 - @AssertFalse/True
- Business
 - @Email
 - @EAN
 - @CreditCard
- Range
 - @Min/@Max
 - @Digits
 - @Length
 - @Size
- Other
 - @Valid
 - @Pattern

...and your own!

Expected Use of JSR 303

- Java Persistence API 2.0
 - Automatic listener
 - DDL update
- WebBeans
- JavaServer faces technology and AJAX libraries
 - Ajax4jsf/RichFaces
- Any validation metadata consumer library
- Embedded in Java Platform, Standard Edition 7 (Java SE 7)?

Agenda

- JSR 303 Beans Validation
 - What's the problem
 - Standardizing the solutions
- Demo
- EJB 3.0 Architecture Interceptors
 - What is it useful for?
 - How to use them?
- **Even Better Demo**
 - **Now you've got the picture**

Intercepting Business Components

- Do stuff before and/or after a business method
 - “Cross cutting” concerns
- Available in most application servers
 - Proprietary extension
- AOP
 - Lots of usecase end up being simple interception
- Interception + metadata
 - Actually useful!
 - “contextual” concern

EJB 3.0 Architecture

- Inherits the good of EJB 2 Architecture
- POJO Style
- Annotations instead of XML
- Lightweight ORM
- Ease of use

EJB 3.0 Architecture Interceptor ID Card

- Finally standardized!
- On any Session- or Message-Driven Bean
 - @Interceptors
 - @AroundInvoke
 - Can also be a lifecycle callback
- Support Dependency Injection
- Default interceptors
- Inherit the invocation context
 - Transaction...



Even Better DEMO

Validating on a Business Method Call



Mixing All Together

- JSR 303: Bean validation
 - Standardization of several frameworks
 - Annotation based
 - Common set API
 - Expected use by other JSRs
- EJB 3.0 architecture interceptors
 - Simple (yet powerful) interception
 - Metadata through annotations
- Yes, Java EE platform is extensible
 - Annotation and interceptor driven frameworks on top of Java EE platform

For More Information

- Bean Validation
 - <http://jcp.org/en/jsr/detail?id=303>
 - Hibernate validator <http://validator.hibernate.org>
- Web Beans
 - <http://jcp.org/en/jsr/detail?id=299>
 - JBoss Seam <http://www.jboss.com/products/seam>
- EJB 3.0 Architecture
 - Enterprise JavaBeans 3.0, O'Reilly
 - JBoss Embedded
 - Spec <http://jcp.org/en/jsr/detail?id=220>



Q&A

<code>



Declarative Programming: Tighten EJB 3.0 and JSR 303 Beans Validation

Emmanuel Bernard
Core Developer
JBoss, A Division of Red Hat

<http://www.jboss.com>
<http://www.hibernate.org>

TS-4112