

# JSR-303 Bean Validation

Emmanuel Bernard

JBoss, by Red Hat

<http://in.relation.to/Bloggers/Emmanuel>



Copyright 2007-2010 Emmanuel Bernard and Red Hat Inc.

- Enable declarative validation in your applications
- Constrain Once, Validate



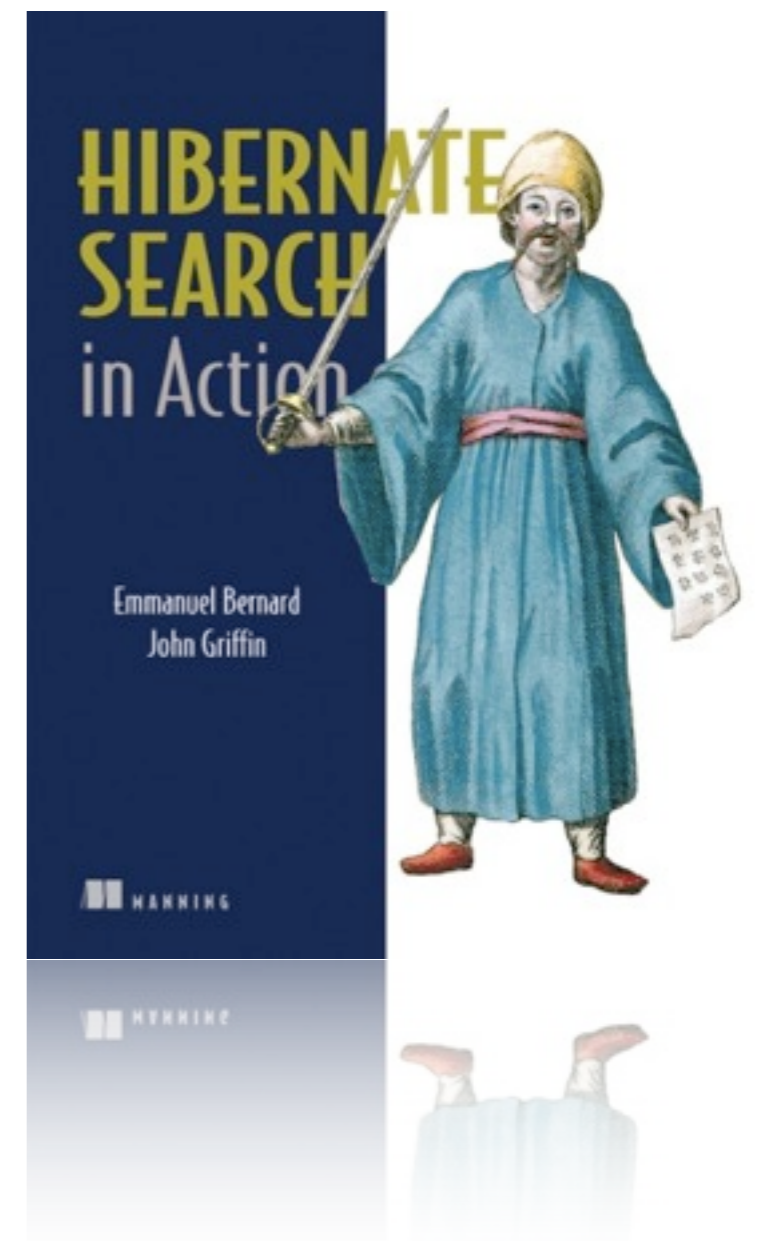
# Emmanuel Bernard

 Hibernate Search in Action

 [blog.emmanuelbernard.com](http://blog.emmanuelbernard.com)

 [twitter.com/emmanuelbernard](https://twitter.com/emmanuelbernard)

 <http://lescastcodeurs.com>



# Constraints

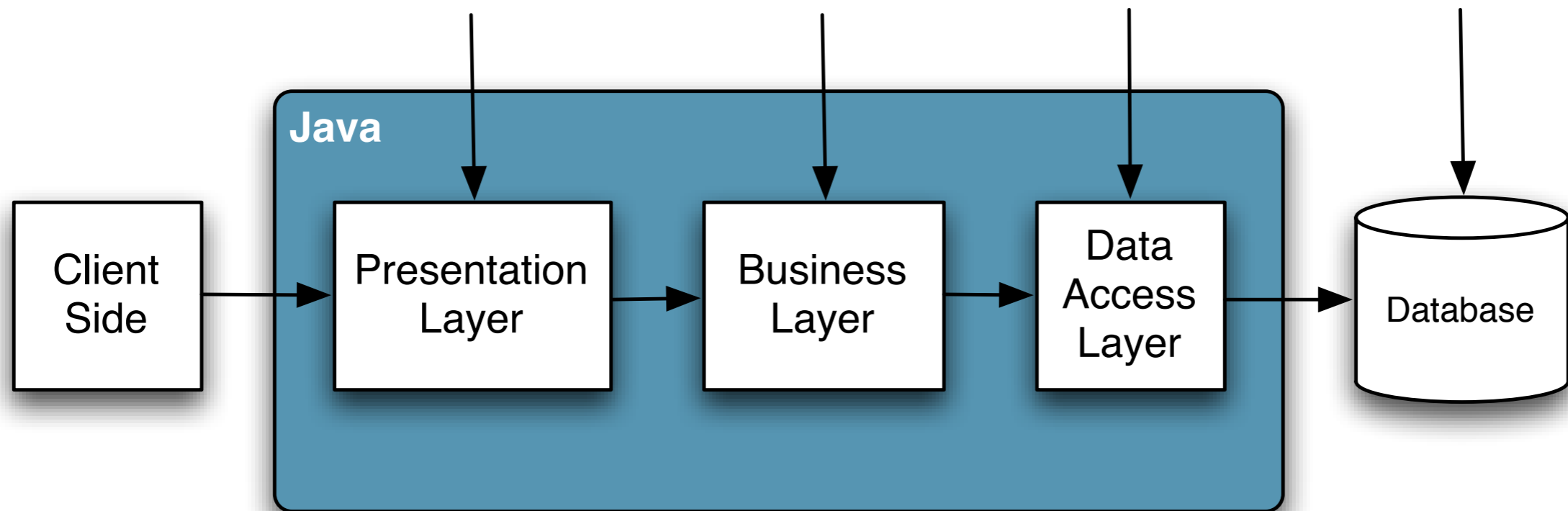
- Constraint
  - restriction on a bean, field or property
  - not null, between 10 and 45, valid email...
- How is that useful
  - give feedback to the user
  - ensure that a service will behave correctly
    - define service range of usability
  - avoid adding crap to the database
    - unless you like fixing the data manually

# Constraints in Java Ecosystems

- Where should they be applied?



# Constraints in the Java Ecosystem



- How many models do you have?
- ONE

# What is the solution?

- Uniform way to express a constraint
  - everybody speaks the same language
  - based on the domain model (JavaBeans™)
- Standard way to validate constraints
  - one runtime engine
  - same validation implementations shared
- Bridge for constraints out of Java™ land
  - API to access the constraint repository

# Declare a constraint

```
public class Address {  
    @NotNull  
    @Size(max=30,  
        message="longer than {max} characters")  
    private String street1;  
    ...  
    @NotNull @Valid  
    private Country country;  
}
```

```
public class Country {  
    @NotNull @Size(max=30)  
    private String name;  
    ...  
}
```



# Groups

- Subset of constraints
- Partial validation
  - screen of a wizard UI
- Constraints applied in a given use case
- Order constraint validations
  - which depends on other validations
  - when a constraint is resource/time intensive

```
interface Billable {}
```

```
interface BuyInOneClick extends Billable, Default {}
```

```
class User {  
    @NotNull(groups=BuyInOneClick.class)  
    PaymentMethod getDefaultCreditCard() {...}
```

```
    @NotNull //Default group  
    String getUsername() {...}
```

```
}
```

```
@GroupSequence(sequence={Default.class, Heavy.class})  
interface Complete {}
```

# Create your own constraint

- Annotations with expressive names
- List of ConstraintValidators
- Constraint composition

```
@Constraint(validatedBy={
    SizeValidatorForCollections.class),
    SizeValidatorForString.class } )
public @interface Size {
    String message() default "{constraint.size}";
    Class<?>[] groups() default {};
    Class<? extends Payload> payload() default {};

    //custom attributes
    int min() default 0;
    int max() default Integer.MAX_VALUE;
}
```

```
public class SizeValidatorForString
    implements ConstraintValidator<Size, String> {

    public void initialize(Size annotation) {}

    public boolean isValid(String value,
        ConstraintValidatorContext context) {}
}
```

# Composition

- Reuse constraints
- Expose meta-informations

```
@NotNull @Size(min=5, max=5)
@Constraint(validatedBy=FrenchZipCodeValidator.class)
public @interface FrenchZipCode {
    String message() default "{constraint.frenchzipcode}";
    Class<?>[] groups() default {};
    Class<? extends Payload>[] payload() default {};
}
```

# Message

- Can be externalized
- Internationalization
- Interpolate constraint parameters
  - must be shorter than {min}
- Custom MessageInterpolator strategy
  - Useful for application frameworks
  - Contextual data
  - Locale

# How to use Bean Validation

- Standalone
- JPA 2.0
- JSF 2.0
- EE 6



# Bootstrap API

- extensible
- support multiple implementations
- type-safe
- can override some attributes contextually
  
- XML configuration optional
  - META-INF/validation.xml

```
ValidatorFactory vf =  
    Validation.buildDefaultValidatorFactory();
```

```
ValidatorFactory vf = Validation.byDefaultProvider()  
    .configure()  
        .messageInterpolator( containerMI )  
        .traversableResolver( jpaTR )  
        .constraintValidatorFactory( webBeansDI )  
        .buildValidatorFactory();
```

```
ValidatorFactory vf = Validation  
    .byProvider(HibernateValidationProvider.class)  
    .configure()  
        .messageInterpolator( containerMI )  
        .failFast()  
        .enableLegacyConstraints()  
        .buildValidatorFactory();
```

# Manual validation

- Get a Validator from a ValidatorFactory

```
Set<ConstraintViolation<User>> errors =  
    validator.validate(user);
```

```
Set<ConstraintViolation<User>> errors =  
    validator.validate(user, BuyInOneClick.class);
```

- ConstraintViolation
  - error message / message template
  - invalid value

# Accessing the metadata

- DDL generation, tools, JavaScript generators
- Metadata API

`PropertyDescriptor - validator.getConstraintsForClass (User.class)`

`PropertyDescriptor - beanDescr.getConstrainedProperties ()`

`ConstraintDescriptor - descr.getConstraintDescriptors ()`

`- constrDescr.getComposingConstraints ()`

- Shines with:
  - composition
  - built-in annotations

# JSF 2 integration

- Zero conf
- Validate input components
  - find property via Expression Language
  - call Bean Validation on input value
  - return localized error messages
  - use JSF user Locale
    - custom MessageInterpolator

# Java Persistence 2

- On entity change
  - validation
  - can select the groups validated
- Make use of a custom TraversableResolver
  - do not traverse associations

# Java EE 6

- Validator as an injectable resource

```
@Resource Validator validator;  
//or  
@Resource ValidatorFactory vf;
```

- Or in CDI (Java Context and Dependency Injection)

```
@Inject Validator validator;  
//or  
@Inject ValidatorFactory vf;
```

# Disclaimer

- I am NOT a designer



# Disclaimer

- I am REALLY NOT a designer

# Demo

## Validating in Java EE 6

# Bean Validation

- Status. It's done.

# Hibernate Validator 4

- Released too. 4.0.2
  - What are you waiting for?
- Road Map
  - backward compatible with legacy Hibernate Validator usage
  - some cool ideas out of the spec scope
- License
  - ASL 2.0

# Questions? Or not?

- JCP.org
  - search “303”
  - <http://in.relation.to>
    - “bean validation” tag
- Hibernate Validator
  - <http://validator.hibernate.org>
  - <http://forum.hibernate.org/viewforum.php?f=26>

