Hibernate Validator

Conception et Developpement d'application d'Entreprise à Large echelle

Olivier Devoisin
Kevin Gallardo

Université Pierre et Marie Curie

9 Decembre 2014

Introduction

- Hello

Introduction

- Hello
- Hello, again.

Hibernate Validator

What is it ?

Hibernate Validator

- Application layer
- Independent
- Works with or without Hibernate

Hibernate Validator

What is it ?

Hibernate Validator

- Application layer
- Independent
- Works with or without Hibernate

2 ways to use it

- Can work with Annotations
- With XML configuration file

Hibernate Validator

What is it used for ?

- Add a new verifiaction layer for object attributes

Hibernate Validator

What is it used for ?

- Add a new verifiaction layer for object attributes
- Verifications on multiple criteria

Hibernate Validator

What is it used for ?

- Add a new verifiaction layer for object attributes
- Verifications on multiple criteria
- Equivalent to JSF Verification module, but executes on server side

Hello World

Simple example

```
public class Address {
  private long id;

  private String street;
  @Pattern(regexp="^[A-Za-z -]+$", message="Le nom de la ville est
      invalide")
  private String city;
  @Digits(integer="6")
  private String zip;
  @NotNull(message="Le pays est obligatoire")
  private String country;

  [...]
}
```

Installation and configuration

Simple example

- Version : 5.1.3 Final (Validator Download page)
  `http://hibernate.org/validator/downloads/`
- Unizp the archive and import libs from dist/ and dist/lib/required in your WebContent/WEB-INF/lib/

Usage

Without Hibernate

```
import java.util.Set;
import javax.validation.ConstraintViolation;
import javax.validation.Validation;
import javax.validation.Validator;
import javax.validation.ValidatorFactory;
[...]
ValidatorFactory factory = Validation.buildDefaultValidatorFactory
    ();
validator = factory.getValidator();
Address a = new Address(street, city, zip, null);
Set<ConstraintViolation<Address>> constraintViolations = validator.
    validate(a);
System.out.println("----------------------"+constraintViolations.
    size());
```

Prints *1* because *country* is *null*

Usage

With Hibernate

- Hibernate checks automatically validation fields defined with Annotations, before executing persistance operation.

Usage

With Hibernate

- Hibernate checks automatically validation fields defined with Annotations, before executing persistance operation.

Exception

- If constraint violation when persist : ConstraintViolationException

Most Used Annotations

With Hibernate

- *@AssertTrue, @AssertFalse*
- *@Max(value =), @Min(value =)*
- *@Size(min = , max =)*, for collections arrays, etc..
- *@Null, @NotNull*
- *@Future, @Past*, checks whether the annotated date is in the future/past
- *@Pattern(regex =)*, use a regex to check validity of the field

Addtional Annotations

With Hibernate

- *@Email*
- *@NotBlank*
- *@NotEmpty*
- *@CreditCardNuber*
- *@SafeHtml*, checks whether the annotated value contains potentialliy malicious fragments such as <script/>
- ...

Custom Annotations

Definition

- You can create custom annotations, with your own code for verification

Custom Annotations

Definition

- You can create custom annotations, with your own code for verification
- Available with or without Hibernate

Custom Annotations

Example

- Here we want to check if a phoneKind is valid
- phoneKind can be one of the three defined values
- @ValidPhoneKind

Custom Annotations

Example

```
package myConstraints;

public class PhoneKindStrings{
  public static String mobileKind = "mobile";
  public static String homeKind = "home";
  public static String officeKind = "office";
}
```

Custom Annotations

Example

```
package myConstraints;

/* Import right libs */
import **

@Target( { METHOD, FIELD, ANNOTATION_TYPE })
@Retention(RUNTIME)
@Constraint(validatedBy = PhoneKindValidator.class)
public @interface CheckPhoneKind {
    String message() default "Invalid␣Phone␣Kind";
    Class<?>[] groups() default {};
    Class<? extends Payload>[] payload() default {};
}
```

Custom Annotations

Example

```
package myConstraints;
/* Imports */
public class PhoneKindValidator implements ConstraintValidator<
    CheckPhoneKind, String> {
    public void initialize(CheckCase constraintAnnotation) {

    }
    public boolean isValid(String object,
        ConstraintValidatorContext constraintContext) {
        if (object == null)
            return true;
        boolean isValid = false;
        if (object == PhoneKindStrings.mobileKind) {isValid = true
            ;}
        else if(object == PhoneKindStrings.homeKind){isValid = true
            ;}
        else if(object == PhoneKindStrings.officeKind){isValid =
            true;}
        return isValid;
    }
}
```

Custom Annotations

Example

```
Class PhoneNumber {
   [...]
   @ValidPhoneKind
   private String phoneKind;
   [...]
}

[...]

ValidatorFactory factory = Validation.buildDefaultValidatorFactory
    ();
validator = factory.getValidator();
PhoneNumber phone = new PhoneNumber("coucou", "0304050607")
Set<ConstraintViolation<PhoneKind>> constraintViolations =
    validator.validate(phone);
System.out.println("Violation␣number␣:␣"+constraintViolations.size
    ());
for (ConstraintViolation<PhoneKind> violations :
    constraintViolations){
  System.out.println("Violation␣message␣:␣"+violations.getMessage()
      );
}
```

TME

Overview

- Add common annotations in your Contact project
- Iterate through possible errors while making validation faults
- Use Hibernate and catch validation exceptions
- Create a custom annotation : subject below

TME

Subjects for annotations

- Common Annotations : all required fields not null; Email format verification; Phone number only digits; etc...

TME

Subjects for annotations

- Common Annotations : all required fields not null; Email format verification; Phone number only digits; etc...
- Custom Annotation : Create an annotation to check, for an Email field, if the provider of the address (@mail.com), exists in a dynamic providers collection, in which you can separately add values.

References

- Hibernate Validator `http://hibernate.org/validator/`
- Hibernate Validator FAQ `http://hibernate.org/validator/faq/`
- Documentation Hibernate Validator `http://docs.jboss.org/hibernate/validator/5.1/reference/en-US/html/`
- Custom Constraints in Hibernate `https://docs.jboss.org/hibernate/validator/4.0.1/reference/en/html/validator-customconstraints.html`

References