

Hibernate Tip: Validate that an entity attribute value is within a defined range

Question:

The value of one of the attributes of my entity has to be within a defined range. What is the best approach to automatically validate the attribute value before persisting it?

Solution:

The Bean Validation Specification and its reference implementation Hibernate Validator provide easy to use and powerful validation options. It integrates with JPA and Hibernate ORM so that it triggers the validation for all entity attributes before persisting or updating an entity.

It only takes 2 steps to automatically validate that the value of an entity attribute is within a defined range. You need to:

- add the required dependencies to your project
- annotate the attribute with the *BeanValidation* annotations

You need to add the following maven dependencies to your project.

```
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-validator</artifactId>
  <version>5.4.1.Final</version>
</dependency>
<dependency>
  <groupId>javax.el</groupId>
  <artifactId>javax.el-api</artifactId>
  <version>2.2.4</version>
</dependency>
```

Hibernate Tip: Validate that an entity attribute value is within a defined range

```
<dependency>
  <groupId>org.glassfish.web</groupId>
  <artifactId>javax.el</artifactId>
  <version>2.2.4</version>
</dependency>
```

In the next step, you need to define the validation. In this example, you need to annotate the entity attribute with *@Min* and *@Max* to define the lower and upper boundary of the valid range.

You can see an example of it in the following code snippet. It checks that the *numPages* attribute has a value between *100* and *1000*.

```
@Entity
public class Book {

    @Min(100)
    @Max(1000)
    private int numPages;

    ...

}
```

When you now try to persist a *Book* entity and the value of *numPages* is outside of the allowed range, Hibernate Validator throws one of the following exceptions.

Hibernate Tip: Validate that an entity attribute value is within a defined range

14:54:05,684 ERROR

[org.hibernate.internal.ExceptionMapperStandardImpl] -
HHH000346: Error during managed flush [Validation failed for
classes [org.thoughts.on.java.model.Book] during persist time
for groups [javax.validation.groups.Default,]

List of constraint violations:[

ConstraintViolationImpl{interpolatedMessage='must be
greater than or equal to 100', propertyPath=numPages,
rootBeanClass=class org.thoughts.on.java.model.Book,
messageTemplate='{javax.validation.constraints.Min.messag
e}}'

14:57:24,905 ERROR

[org.hibernate.internal.ExceptionMapperStandardImpl] -
HHH000346: Error during managed flush [Validation failed for
classes [org.thoughts.on.java.model.Book] during persist time
for groups [javax.validation.groups.Default,]

List of constraint violations:[

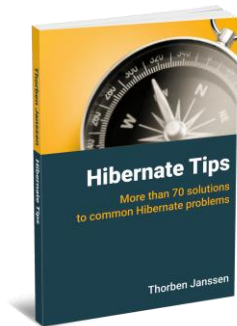
ConstraintViolationImpl{interpolatedMessage='must be
less than or equal to 1000', propertyPath=numPages,
rootBeanClass=class org.thoughts.on.java.model.Book,
messageTemplate='{javax.validation.constraints.Max.messag
e}}'

Hibernate Tip: Validate that an entity attribute value is within a defined range

Learn more

You can do a lot more than just the simple validation I showed in this Hibernate Tip. I explain the *BeanValidation* integration in more detail in [How to automatically validate entities with Hibernate Validator](#).

Hibernate Tips Book



Get more recipes like this one in my book [Hibernate Tips: More than 70 solutions to common Hibernate problems](#).

It gives you more than 70 ready-to-use recipes for topics like basic and advanced mappings, logging, Java 8 support, caching and statically and dynamically defined queries.