

What is @Repeatable and why should you like it?

Sometimes, you want to apply the same annotation multiple times. Before Java 8, the only way to do that was to use an additional annotation and provide an array of the annotations you want to apply as a value.

With Java 8, you can define an annotation as repeatable and apply it multiple times without any wrapper annotations.

Which Hibernate annotations are repeatable?

Annotation	JavaDoc
AnyMetaDef	Used to provide metadata about an Any or ManyToMany mapping.
ColumnTransformer	Custom SQL expression used to read the value from and write a value to a column. Use for direct object loading/saving as well as queries. The write expression must contain exactly one '?' placeholder for the value. For example: read="decrypt(credit_card_num)" write="encrypt(?)"
FetchProfile	Define the fetching strategy profile.
Filter	Add filters to an entity or a target entity of a collection.
FilterDef	Filter definition. Defines a name, default condition and parameter types (if any).
FilterJoinTable	Add filters to a join table collection.
GenericGenerator	Generator annotation describing any kind of Hibernate generator in a generic (de-typed) manner.

@Repeatable annotations in Hibernate 5.2

Annotation	JavaDoc
JoinColumnOrFormula	Allows joins based on column or a formula. One of formula() or column() should be specified, but not both.
NamedNativeQuery	Extends NamedNativeQuery with Hibernate features.
Table	Complementary information to a table either primary or secondary.
Tuplizer	Define a tuplizer for an entity or a component.
TypeDef	A type definition. Much like Type, but here we can centralize the definition under a name and refer to that name elsewhere. The plural form is TypeDefs.

How to use @Repeatable annotations

With JPA and Hibernate versions before 5.2, you were not able to annotate an entity with multiple of the same annotations. If there was the need to do that, e.g. when you wanted to define multiple @NamedQuery for an entity, you had to provide them as the value to a @NamedQueries annotation.

```
@Entity
@NamedQueries({
    @NamedQuery(name = "Book.findByTitle", query = "SELECT b
FROM Book b WHERE b.title = :title"),
    @NamedQuery(name = "Book.findByPublishingDate", query =
"SELECT b FROM Book b WHERE b.publishingDate =
:publishingDate")
})
public class Book implements Serializable {
    ...
}
```

@Repeatable annotations in Hibernate 5.2

That's no longer required, if you use Hibernate's version of the `@NamedQuery` annotation or of any other annotation listed in the previous section. As you can see in the following code snippet, you can now add multiple `org.hibernate.annotations.NamedQuery` annotations directly to the entity.

```
@Entity
@NamedQuery(name = "Hibernate5Book.findByTitle", query =
"SELECT b FROM Hibernate5Book b WHERE b.title = :title")
@NamedQuery(name = "Hibernate5Book.findByPublishingDate",
query = "SELECT b FROM Hibernate5Book b WHERE
b.publishingDate = :publishingDate")
public class Hibernate5Book implements Serializable {
    ...
}
```