

Hibernate Envers - Extend the standard revision

Hibernate Envers' default audit log misses some important information. It just audits what had happened but not who did it.

The default revision entity doesn't store any information of the user who performed the operations. If you also want to store user information, like the username or IP address, you have to use a custom entity to store the revision.

Create a custom revision entity

You need to implement 2 classes to create and register a custom revision entity: the revision entity and a *RevisionListener*.

Implement a revision entity

The easiest way to implement your own revision entity is to extend the as I do in the following code snippet. If you don't want to do that, your entity needs to have at least 2 attributes:

1. a revision number of type *int/Integer* or *long/Long* annotated with *@RevisionNumber* and
2. a revision timestamp of type *long/Long* or *java.util.Date* annotated with *@RevisionTimestamp*.

Hibernate Envers - Extend the standard revision

In this example, I extend the *DefaultRevisionEntity* because I just want to store an additional *userName* attribute for each revision. I also reference my implementation of the *RevisionListener* interface in the *@RevisionEntity* annotation.

```
@Entity
@RevisionEntity(MyRevisionListener.class)
public class MyRevision extends DefaultRevisionEntity {
    private String userName;

    public String getUserName() {
        return userName;
    }

    public void setUserName(String userName) {
        this.userName = userName;
    }
}
```

Implement a RevisionListener

The *RevisionListener* tells Hibernate Envers how to set the attributes of the revision entity.

You can see an example of it in the following code snippet. You just need to implement the *newRevision(Object revisionEntity)* method which gets the newly instantiated revision entity as a parameter. The only thing you have to do is to set the additional attributes. In this example, I just need to set the *userName* attribute.

Hibernate Envers - Extend the standard revision

```
public class MyRevisionListener implements RevisionListener
{
    @Override
    public void newRevision(Object revisionEntity) {
        MyRevision rev = (MyRevision) revisionEntity;
        rev.setUserName(getUserName());
    }

    ...
}
```

Use revision data in queries

When you create your *AuditQuery*, you can use the attributes of the revision entity in the same way as any attribute of an audited entity.

The following code snippet shows an example in which I select the numbers of all revisions in which a user with *userName* "User 1" created, updated or deleted a Book entity.

```
AuditQuery q =
auditReader.createQuery().forRevisionsOfEntity(Book.class,
false, true);
q.addProjection(AuditEntity.revisionNumber());
q.add(AuditEntity.revisionProperty("userName").eq("User
1"));
List<Number> revisionNumbers = q.getResultList();
```