

Persist creation and update timestamps

@CreationTimestamp and *@UpdateTimestamp*

Hibernate's *@CreationTimestamp* and *@UpdateTimestamp* annotations make it easy to track the timestamp of the creation and last update of an entity. The only thing you have to do is to add the annotation to an entity attribute.

When a new entity gets persisted, Hibernate gets the current timestamp from the VM and sets it as the value of the attribute annotated with *@CreationTimestamp*. After that, Hibernate will not change the value of this attribute.

The value of the attribute annotated with *@UpdateTimestamp* gets changed in a similar way with every SQL Update statement.

Hibernate gets the current timestamp from the VM and sets it as the update timestamp on the SQL Update statement.

Supported attribute types

You can use the *@CreationTimestamp* and *@UpdateTimestamp* with the following attribute types:

- *java.time.LocalDate* (since Hibernate 5.2.3)
- *java.time.LocalDateTime* (since Hibernate 5.2.3)
- *java.util.Date*
- *java.util.Calendar*
- *java.sql.Date*
- *java.sql.Time*
- *java.sql.Timestamp*

Persist creation and update timestamps

Example

As you can see in the following code snippet, I just added the *@CreationTimestamp* annotation to the *createDateTime* attribute and the *@UpdateTimestamp* annotation to the *updateDateTime* attribute.

```
@Entity
public class MyEntity {

    @Column
    @CreationTimestamp
    private LocalDateTime createDateTime;

    @Column
    @UpdateTimestamp
    private LocalDateTime updateDateTime;

    ...
}
```