

[Home](#) › [Interview](#) › [Spring, Hibernate, & Maven Interview Q&A](#) › [Hibernate](#) › 09:

Hibernate entities with auditable, soft delete & optimistic locking fields

# 09: Hibernate entities with auditable, soft delete & optimistic locking fields

Posted on [December 22, 2014](#) by [Arulkumaran Kumaraswamipillai](#) — [No](#)

[Comments](#) ↓

There are certain generic columns that database tables have like

1. Auto generated identity columns.
2. **Auditing** columns like createdDtTm, createdBy, modifiedDtTm, and modifiedBy.
3. **Soft delete** or **logical delete** flags like inactive 'Y' or 'N'.
4. **Optimistic locking** detection based on columns like Timestamp or version number.

These logic can be mapped in a generic way using hibernate so that all the other entities can reuse. Let's look at some sample code using Hibernate annotations.

## 1. Identity columns.

## 600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

[open all](#) | [close all](#)

☒ [Ice Breaker Interview](#)

☒ [Core Java Interview C](#)

☒ [JEE Interview Q&A \(3](#)

☒ [Pressed for time? Jav](#)

☒ [SQL, XML, UML, JSC](#)

☒ [Hadoop & BigData Int](#)

☒ [Java Architecture Inte](#)

☒ [Scala Interview Q&As](#)

☒ [Spring, Hibernate, & I](#)

☒ [Spring \(18\)](#)

☒ [Hibernate \(13\)](#)

☒ [01: ♥♦ 15+ Hiber](#)

☒ [01b: ♦ 15+ Hiber](#)

☒ [02: Understandir](#)

☒ [03: Identifying ar](#)

☒ [04: Identifying ar](#)

☒ [05: Debugging I-](#)

☒ [06: Hibernate Fii](#)

☒ [07: Hibernate mi](#)

☒ [08: Hibernate au](#)

☒ [09: Hibernate en](#)

☒ [10: Spring, Java](#)

```

1 package com.myapp.common;
2
3 import static javax.persistence.GenerationType.IDENTITY;
4
5 import javax.persistence.Column;
6 import javax.persistence.GeneratedValue;
7 import javax.persistence.Id;
8 import javax.persistence.MappedSuperclass;
9
10 @SuppressWarnings("serial")
11 @MappedSuperclass
12 public class StandardIdLongBaseEntity extends AbstractStandardIdLongBaseEntity {
13     {
14         @Id
15         @GeneratedValue(strategy = IDENTITY)
16         @Column(name = "ID", unique = true, nullable = false)
17         @Override
18         public Long getId() {
19             return super.getId();
20         }
21     }
22 }
23

```

## 2. Audit fields and optimistic locking detection using timestamp. Firstly the "MappedSuperclass"

```

1 package com.myapp.common;
2
3 import javax.persistence.Column;
4 import javax.persistence.Embedded;
5 import javax.persistence.MappedSuperclass;
6 import javax.persistence.Version;
7
8 import org.apache.commons.lang.builder.ToStringBuilder;
9 import org.apache.commons.lang.builder.ToStringStyle;
10 import org.hibernate.annotations.Generated;
11 import org.hibernate.annotations.GenerationTime;
12 import org.hibernate.annotations.TypeDef;
13 import org.hibernate.annotations.TypeDefs;
14
15 @SuppressWarnings("serial")
16 @MappedSuperclass
17 public abstract class AuditableBaseEntity<pk> extends AbstractAuditableBaseEntity<pk> {
18     {
19         private byte[] timestamp;
20         private StandardAuditFields auditFields = new StandardAuditFields();
21     }
22
23     @Version
24     @Column(name = "Timestamp", insertable = false, updatable = true)
25     @Generated(GenerationTime.ALWAYS)
26     public byte[] getTimestamp() {
27         return timestamp;
28     }
29
30     public void setTimestamp(byte[] timestamp) {
31         this.timestamp = timestamp;
32     }
33

```

[11: Hibernate de](#)  
[12: Hibernate cu](#)  
[AngularJS \(2\)](#)  
[Git & SVN \(6\)](#)  
[JMeter \(2\)](#)  
[JSF \(2\)](#)  
[Maven \(3\)](#)  
[Testing & Profiling/Sa](#)  
[Other Interview Q&A 1](#)  
[Free Java Interview](#)

## 16 Technical Key Areas

[open all](#) | [close all](#)

- [Best Practice \(6\)](#)
- [Coding \(26\)](#)
- [Concurrency \(6\)](#)
- [Design Concepts \(7\)](#)
- [Design Patterns \(11\)](#)
- [Exception Handling \(3\)](#)
- [Java Debugging \(21\)](#)
- [Judging Experience I](#)
- [Low Latency \(7\)](#)
- [Memory Management](#)
- [Performance \(13\)](#)
- [QoS \(8\)](#)
- [Scalability \(4\)](#)
- [SDLC \(6\)](#)
- [Security \(13\)](#)
- [Transaction Managen](#)

## 80+ step by step Java Tutorials

[open all](#) | [close all](#)

- [Setting up Tutorial \(6\)](#)
- [Tutorial - Diagnosis \(2](#)

```

33
34 @Embedded
35 public StandardAuditFields getAuditFields() {
36     if (auditFields == null)
37     {
38         // TODO: Not thread safe.
39         auditFields = new StandardAuditFields()
40     }
41     return auditFields;
42 }
43
44 public void setAuditFields(StandardAuditFields
45     this.auditFields = auditFields;
46 }
47
48 @Override
49 public String toString()
50 {
51     return ToStringBuilder.reflectionToString
52 }
53 }
54

```

Next, the “Embeddable” class with fields for auditing.

```

1 package com.myapp.common;
2
3 import java.util.Date;
4
5 import javax.persistence.Column;
6 import javax.persistence.Embeddable;
7 import javax.persistence.Temporal;
8 import javax.persistence.TemporalType;
9 import javax.persistence.Transient;
10
11 import org.apache.commons.lang.builder.ToStringBuilder;
12 import org.apache.commons.lang.builder.ToStringStyle;
13 import org.hibernate.annotations.Generated;
14 import org.hibernate.annotations.GenerationTime;
15
16 @Embeddable
17 public class StandardAuditFields {
18     private String createdBy;
19     private Date createdDtTm;
20     private Date modifiedDtTm;
21     private String modifiedBy;
22
23
24     @Temporal(TemporalType.TIMESTAMP)
25     @Generated(GenerationTime.INSERT)
26     @Column(name="CreatedDtTm", length=23, insert
27     public Date getCreatedDtTm() {
28         return createdDtTm;
29     }
30
31     public void setCreatedDtTm(Date createdDtTm) {
32         this.createdDtTm = createdDtTm;
33     }
34
35     @Temporal(TemporalType.TIMESTAMP)
36     @Generated(GenerationTime.ALWAYS)
37     @Column(name="ModifiedDtTm", length=23, insert
38     public Date getModifiedDtTm() {

```

- [Akka Tutorial \(9\)](#)
- [Core Java Tutorials \(2\)](#)
- [Hadoop & Spark Tuto](#)
- [JEE Tutorials \(19\)](#)
- [Scala Tutorials \(1\)](#)
- [Spring & Hlbername Ti](#)
- [Tools Tutorials \(19\)](#)
- [Other Tutorials \(45\)](#)

## 100+ Java pre-interview coding tests

[open all](#) | [close all](#)

- [Can you write code? \(1\)](#)
- [♦ Complete the given](#)
- [Converting from A to I](#)
- [Designing your classe](#)
- [Java Data Structures](#)
- [Passing the unit tests](#)
- [What is wrong with th](#)
- [Writing Code Home A](#)
- [Written Test Core Jav](#)
- [Written Test JEE \(1\)](#)

## How good are your .....?

[open all](#) | [close all](#)

- [Career Making Know-](#)
- [Job Hunting & Resum](#)

```

39     return modifiedDtTm;
40 }
41
42 public void setModifiedDtTm(Date modifiedDtTm)
43     this.modifiedDtTm = modifiedDtTm;
44 }
45
46 @Column(name="ModifiedBy", length=30, insertable=true, updatable=true)
47 public String getModifiedBy() {
48     return modifiedBy;
49 }
50
51 public void setModifiedBy(String modifiedBy) {
52     this.modifiedBy = modifiedBy;
53 }
54
55 @Override
56 public String toString()
57 {
58     return ToStringBuilder.reflectionToString(this);
59 }
60 }
61
62

```

### 3. Mapping soft delete.

```

1  package com.myapp.common;
2
3  import javax.persistence.Column;
4  import javax.persistence.MappedSuperclass;
5
6  import org.hibernate.annotations.Type;
7  import org.hibernate.annotations.Where;
8
9  @SuppressWarnings("serial")
10 @MappedSuperclass
11 @Where(clause = "inactiveFlag = 'N'")
12 public class BaseSoftDeleteableLongEntity extends BaseEntity
13 {
14     private boolean isSoftDeleted;
15
16     @Column(name = "InactiveFlag", nullable = false)
17     @Type(type = "yes_no")
18     public boolean isSoftDeleted()
19     {
20         return isSoftDeleted;
21     }
22
23     public void setSoftDeleted(boolean isSoftDeleted)
24     {
25         this.isSoftDeleted = isSoftDeleted;
26     }
27 }
28
29

```

### 4 Finally, defining your entity.

```
1  package com.myapp.batch;
2
3  import com.google.common.base.Objects;
4  import java.util.Date;
5
6  import javax.persistence.AttributeOverride;
7  import javax.persistence.Column;
8  import javax.persistence.Entity;
9  import javax.persistence.EnumType;
10 import javax.persistence.Enumerated;
11 import javax.persistence.Table;
12 import javax.persistence.Transient;
13
14 import org.apache.commons.lang.StringUtils;
15 import org.codehaus.jackson.annotate.JsonIgnore;
16 import org.hibernate.annotations.SQLDelete;
17 import org.hibernate.annotations.Type;
18 import org.hibernate.annotations.TypeDef;
19 import org.hibernate.annotations.TypeDefs;
20 import org.hibernate.annotations.Where;
21
22 @Entity
23 @Table(name = "Tbl_Batch_Run", schema = "dbo",
24 @AttributeOverride(name = "id", column = @Column
25 @Where(clause = "inactiveFlag = 'N'")
26 @SQLDelete(sql = "UPDATE Tbl_Batch_Run set inac
27 @JsonIgnoreProperties(
28 {
29     "auditFields", "id", "softDeleted"})
30 @TypeDefs(
31 {@TypeDef(name = "BatchTypeCd", typeClass = Bat
32 public class BatchRun extends BaseSoftDeleteabl
33 {
34
35     private String entity;
36     private long batchId;
37     private BatchTypeCd batchTypeCd;
38     private Status status;
39
40
41     /**
42      * Required constructor for JSON Marshal
43      */
44     public BatchRun()
45     {
46     }
47
48     public BatchRun(BatchRun aBatchRun)
49     {
50         this(aBatchRun.getEntity(), aBatchRun.g
51     }
52
53
54     public BatchRun(String entity, long batchId
55     {
56         this.entity = entity;
57         this.batchId = batchId;
58         this.batchTypeCd = batchTypeCd;
59         this.status = status;
60     }
61
62     @Column(name = "Entity", nullable = false)
63     public String getEntity()
64     {
65         return this.entity;
```

```

66     }
67
68     public void setEntity(String entity)
69     {
70         this.entity = entity;
71     }
72
73     @Column(name = "BatchId", nullable = false)
74     public long getBatchId()
75     {
76         return this.batchId;
77     }
78
79     public void setBatchId(long batchId)
80     {
81         this.batchId = batchId;
82     }
83
84     @Enumerated(EnumType.STRING)
85     @Column(name = "Status", nullable = false,
86     public Status getStatus()
87     {
88         return status;
89     }
90
91     public void setStatus(Status status)
92     {
93         this.status = status;
94     }
95
96
97     @Type(type = "BatchTypeCd")
98     @Column(name = "BatchType", nullable = false)
99     public BatchTypeCd getBatchTypeCd()
100    {
101        return batchTypeCd;
102    }
103
104    public void setBatchTypeCd(BatchTypeCd batchTypeCd)
105    {
106        this.batchTypeCd = batchTypeCd;
107    }
108
109    /**
110     * Business key for this object is the batchId
111     * @see java.lang.Object#equals(java.lang.Object)
112     */
113    @Override
114    public boolean equals(Object obj)
115    {
116        if (obj == null)
117        {
118            return false;
119        }
120        if (getClass() != obj.getClass())
121        {
122            return false;
123        }
124        final BatchRun other = (BatchRun) obj;
125
126        return Objects.equal(batchId, other.batchId)
127            && Objects.equal(entity, other.entity)
128            && Objects.equal(batchTypeCd, other.batchTypeCd);
129    }
130
131

```

```
132     @Override
133     public int hashCode()
134     {
135         return Objects.hashCode(batchId, entity
136     }
137
138     @Override
139     public String toString()
140     {
141         return Objects.toStringHelper(this)
142             .addValue(batchId)
143             .addValue(entity)
144             .addValue(batchTypeCd)
145             .toString();
146     }
147 }
148
```

## Popular Member Posts

♦ 11 Spring boot interview questions & answers

904 views

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

816 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

427 views

18 Java scenarios based interview Questions and Answers

408 views

♦ 7 Java debugging interview questions & answers

324 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

311 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

304 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

301 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

251 views



## ♦ Object equals Vs == and pass by reference Vs value

234 views

Bio

Latest Posts



### Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via [Amazon.com](https://www.amazon.com) in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site. **945+** paid members. [join my LinkedIn Group](#). [Reviews](#)



### About Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via [Amazon.com](https://www.amazon.com) in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site. **945+** paid members. [join my LinkedIn Group](#). [Reviews](#)

◀ 06: Hibernate First & second level cache interview questions and answers



Are you ready for digital recruitment? Can you be found in Google search? >

**Posted in** Hibernate, member-paid

## Leave a Reply

Logged in as geethika. [Log out?](#)

### Comment

Post Comment

## Empowers you to open more doors, and fast-track

### Technical Know Hows

☀ [Java generics in no time](#) ☀ [Top 6 tips to transforming your thinking from OOP to FP](#) ☀ [How does a HashMap internally work? What is a hashing function?](#)

☀ [10+ Java String class interview Q&As](#) ☀ [Java auto un/boxing benefits & caveats](#) ☀ [Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect](#)

### Non-Technical Know Hows

☀ [6 Aspects that can motivate you to fast-track your career & go places](#) ☀ [Are you reinventing yourself as a Java developer?](#) ☀ [8 tips to safeguard your Java career against offshoring](#) ☀ [My top 5 career mistakes](#)

## Prepare to succeed

☀ [Turn readers of your Java CV go from “Blah blah” to “Wow”?](#) ☀ [How to prepare for Java job interviews?](#) ☀ [16 Technical Key Areas](#) ☀ [How to choose from multiple Java job offers?](#)

Select Category ▼

## © Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.