### JPA supported Identifier Generators

The @Id annotation lets you define which property is the identifier of your entity bean. This property can be set by the application itself or be generated by Hibernate (preferred). You can define the identifier generation strategy using **@GeneratedValue** annotation:

* **AUTO** - either **identity** column, **sequence** or **table** depending on the underlying Database
* **TABLE** – hilo algorithm
* **IDENTITY** - identity column
* **SEQUENCE** – sequence

**NOTE:** Hibernate provides more id generators than the basic JPA.

**1) Without any Generator:**

**Student.java**

1. **@Entity**
2. **@Table(name = "STUDENT")**
3. **public class Student {**
4. **@Id**
5. **@Column(name = "SNO")**
6. **private Integer sid;**
7. **@Column(name = "NAME")**
8. **private String name;**
9. **@Column(name = "COURSE", length = 20)**
10. **private String course;**
11. **@Column(name = "FEE")**
12. **private double fee;**
13. **// getters & setters**
14. **}**

**StudentDAO.java**

1. **public class StudentDAO {**
2. **public static void main(String[] args) {**
3. **Session session = SessionUtil.getSession();**
4. **session.getTransaction().begin();**
5. **for(int i = 1 ; I <= 5 ;i++){**
6. **Student student = new Student();**
7. **student.setCourse("java-"+i);**
8. **student.setName("sekhar-"+i);**
9. **student.setFee(111\*i);**
10. **session.save(student);**
11. **}**
12. **session.getTransaction().commit();**
13. **}**
14. **}**

**ERROR:**

**org.hibernate.id.IdentifierGenerationException: ids for this class must be manually assigned before calling save():**

**NOTE :** So if we don’t specify any Id generation strategy, By default hibernate internally takes it as assigned generator.

**2) With @GeneratedValue annotation**

* In this approach **@GeneratedValue** annotation default value is **“strategy=GenerationType.AUTO”**

1. **@Entity**
2. **@Table(name = "STUDENT")**
3. **public class Student {**
4. **@Id**
5. **@GeneratedValue**
6. **@Column(name = "SNO")**
7. **private Integer sid;**
8. **@Column(name = "NAME")**
9. **private String name;**
10. **@Column(name = "COURSE", length = 20)**
11. **private String course;**
12. **@Column(name = "FEE")**
13. **private double fee;**
14. **// getters & setters**
15. **}**

**NOTE:** As we didn’t sepecify any strategy, by default it will take **strategy=GenerationType.AUTO**. As we know **AUTO** work like “**native**” generator, so it will take either **identity** or **sequnce** or **hilo**. As we are working with Oracle Database, it will take **sequnce** generator. As we didn’t specify any sequence name, by default it will take sequence name as “**hibernate\_sequence**” So, we can write the above entity as follows.

1. **@Entity**
2. **@Table(name = "STUDENT")**
3. **public class Student {**
4. **@Id**
5. **@GeneratedValue(strategy=GenerationType.*AUTO*)**
6. **@Column(name = "SNO")**
7. **private Integer sid;**
8. **@Column(name = "NAME")**
9. **private String name;**
10. **@Column(name = "COURSE", length = 20)**
11. **private String course;**
12. **@Column(name = "FEE")**
13. **private double fee;**
14. **// getters & setters**
15. **}**

**If we want our own seqncence, then we need to configure sequence name also….**

1. **@Entity**
2. **@Table(name = "STUDENT")**
3. **public class Student {**
4. **@Id**
5. **@SequenceGenerator(name="myGenerator", sequenceName="STUDENT\_SEQ")**
6. **@GeneratedValue(strategy=GenerationType.*AUTO*, generator="myGenerator")**
7. **@Column(name = "SNO")**
8. **private Integer sid;**
9. **@Column(name = "NAME")**
10. **private String name;**
11. **@Column(name = "COURSE", length = 20)**
12. **private String course;**
13. **@Column(name = "FEE")**
14. **private double fee;**
15. **// getters & setters**
16. **}**

* This strategy supports either **identity column**, **sequence** or **table hilo generator** depending on the underlying DB
* If we work with Oracle database, It will consider sequence, because Oracle database doesn’t supports **identity columns**
* If we work with MYSQL database, It will consider **identity,** because MYSQL database supports i**dentity columns**

**Syntax to create identity columns in MYSQL database :**

CREATE TABLE STUDENT(

SNO INT(10) NOT NULL AUTO\_INCREMENT,

COURSE CHAR(20),

FEE FLOAT,

NAME CHAR(20),

PRIMARY KEY (SNO)

)

**NOTE:** It is similar to “**native**” generator of hibernate

**2) With @GeneratedValue(strategy=GenerationType.SEQUENCE)**

1. **@Entity**
2. **@Table(name = "STUDENT")**
3. **public class Student {**
4. **@Id**
5. **@SequenceGenerator(name="myGenerator", sequenceName="STUDENT\_SEQ")**
6. **@GeneratedValue(strategy=GenerationType.** *SEQUENCE***, generator="myGenerator")**
7. **@Column(name = "SNO")**
8. **private Integer sid;**
9. **@Column(name = "NAME")**
10. **private String name;**
11. **@Column(name = "COURSE", length = 20)**
12. **private String course;**
13. **@Column(name = "FEE")**
14. **private double fee;**
15. **// getters & setters**
16. **}**

* This is supported by Oracle, It will take the given sequence and by executing the sequence it will retrieve the identifier.
* This is not supported by MYSQL

**3) With @GeneratedValue(strategy=GenerationType.IDENTITY)**

1. **@Entity**
2. **@Table(name = "STUDENT")**
3. **public class Student {**
4. **@Id**
5. **@GeneratedValue(strategy=GenerationType.***IDENTITY***)**
6. **@Column(name = "SNO")**
7. **private Integer sid;**
8. **@Column(name = "NAME")**
9. **private String name;**
10. **@Column(name = "COURSE", length = 20)**
11. **private String course;**
12. **@Column(name = "FEE")**
13. **private double fee;**
14. **// getters & setters**
15. **}**

* This is not supported by Oracle, But supported by MYSQL database

**4) With @GeneratedValue(strategy=GenerationType.TABLE)**

1. **@Entity**
2. **@Table(name = "STUDENT")**
3. **public class Student {**
4. **@Id**
5. **@TableGenerator(**
6. **name="myGenerator",**
7. **initialValue=2000,**
8. **allocationSize=1,**
9. **table="PK\_VALUE\_TAB",**
10. **pkColumnName="PK\_COLUMN",**
11. **pkColumnValue="PK\_VALUE",**
12. **valueColumnName="PK\_VALUE\_COLUMN")**
13. **@GeneratedValue(strategy=GenerationType.** *TABLE***, generator="myGenerator")**
14. **@Column(name = "SNO")**
15. **private Integer sid;**
16. **@Column(name = "NAME")**
17. **private String name;**
18. **@Column(name = "COURSE", length = 20)**
19. **private String course;**
20. **@Column(name = "FEE")**
21. **private double fee;**
22. **// getters & setters**
23. **}**

### Hibernate supported Identifier Generators

To integrate hibernate speicific genertors into JPA we can use either **@org.hibernate.annotations.GenericGenerator** or **@org.hibernate.annotations.GenericGenerators** annotations.

**1.)assigned**(**org.hibernate.id.Assigned):**

1. **@Entity**
2. **@Table(name = "STUDENT")**
3. **public class Student {**
4. **@Id**
5. **@GenericGenerator(name="myGenerator", strategy="assigned")**
6. **@GeneratedValue(strategy = GenerationType.*AUTO*, generator = "myGenerator")**
7. **@Column(name = "SNO")**
8. **private Integer sid;**
9. **@Column(name = "NAME")**
10. **private String name;**
11. **@Column(name = "COURSE", length = 20)**
12. **private String course;**
13. **@Column(name = "FEE")**
14. **private double fee;**
15. **// getters & setters**
16. **}**

OR

1. **@Entity**
2. **@Table(name = "STUDENT")**
3. **public class Student {**
4. **@Id**
5. **@GenericGenerator(name="myGenerator", strategy="**org.hibernate.id.Assigned **")**
6. **@GeneratedValue(strategy = GenerationType.*AUTO*, generator = "myGenerator")**
7. **@Column(name = "SNO")**
8. **private Integer sid;**
9. **@Column(name = "NAME")**
10. **private String name;**
11. **@Column(name = "COURSE", length = 20)**
12. **private String course;**
13. **@Column(name = "FEE")**
14. **private double fee;**
15. **// getters & setters**
16. **}**

**2.)increment**(**org.hibernate.id.IncrementGenerator)**

1. **@Entity**
2. **@Table(name = "STUDENT")**
3. **public class Student {**
4. **@Id**
5. **@GenericGenerator(name="myGenerator", strategy="**increment **")**
6. **@GeneratedValue(strategy = GenerationType.*AUTO*, generator = "myGenerator")**
7. **@Column(name = "SNO")**
8. **private Integer sid;**
9. **@Column(name = "NAME")**
10. **private String name;**
11. **@Column(name = "COURSE", length = 20)**
12. **private String course;**
13. **@Column(name = "FEE")**
14. **private double fee;**
15. **// getters & setters**
16. **}**

**3.)sequence (org.hibernate.id.SequenceGenerator)**

1. **@Entity**
2. **@Table(name = "STUDENT")**
3. **public class Student {**
4. **@Id**
5. **@GenericGenerator(name="myGenerator", strategy="**sequence **")**
6. **@GeneratedValue(strategy = GenerationType.*AUTO*, generator = "myGenerator")**
7. **@Column(name = "SNO")**
8. **private Integer sid;**
9. **@Column(name = "NAME")**
10. **private String name;**
11. **@Column(name = "COURSE", length = 20)**
12. **private String course;**
13. **@Column(name = "FEE")**
14. **private double fee;**
15. **// getters & setters**
16. **}**

**NOTE:** As there are no parameters it will take default sequence name(HIBERNATE\_SEQUENCE)

1. **@Entity**
2. **@Table(name = "STUDENT")**
3. **public class Student {**
4. **@Id**
5. **@GenericGenerator(**
6. **name="myGenerator",**
7. **strategy="**sequence**"**
8. **parameters**={**@Parameter**(name="sequence",value="STUDENT\_SEQ")}
9. **)**
10. **@GeneratedValue(strategy = GenerationType.*AUTO*, generator = "myGenerator")**
11. **@Column(name = "SNO")**
12. **private Integer sid;**
13. **@Column(name = "NAME")**
14. **private String name;**
15. **@Column(name = "COURSE", length = 20)**
16. **private String course;**
17. **@Column(name = "FEE")**
18. **private double fee;**
19. **// getters & setters**
20. **}**

**4.) User defined generator:(com.sekharit.hibernate.id.AccnoGenerator)**

1. **@Entity**
2. **@Table(name = "STUDENT")**
3. **public class Student {**
4. **@Id**
5. **@GenericGenerator(**
6. **name="myGenerator",**
7. **strategy="** com.sekharit.hibernate.id.StudnetNumberGenerator **"**
8. **)**
9. **@GeneratedValue(strategy = GenerationType.*AUTO*, generator = "myGenerator")**
10. **@Column(name = "SNO")**
11. **private Integer sid;**
12. **@Column(name = "NAME")**
13. **private String name;**
14. **@Column(name = "COURSE", length = 20)**
15. **private String course;**
16. **@Column(name = "FEE")**
17. **private double fee;**
18. **// getters & setters**
19. **}**

**5.) hilo(org.hibernate.id.TableHiLoGenerator)**

1. **@Entity**
2. **@Table(name = "STUDENT")**
3. **public class Student {**
4. **@Id**
5. **@GenericGenerator(**
6. **name="myGenerator",**
7. **strategy="**sequence**"**
8. **parameters**={
9. **@Parameter**(name="table", value="PK\_VALUE\_TAB"),
10. **@Parameter**(name="column", value="PK\_VALUE\_COLUMN")
11. }
12. **)**
13. **@GeneratedValue(strategy = GenerationType.*AUTO*, generator = "myGenerator")**
14. **@Column(name = "SNO")**
15. **private Integer sid;**
16. **@Column(name = "NAME")**
17. **private String name;**
18. **@Column(name = "COURSE", length = 20)**
19. **private String course;**
20. **@Column(name = "FEE")**
21. **private double fee;**
22. **// getters & setters**
23. **}**

**6.) identity(org.hibernate.id.IdentityGenerator)**

1. **@Entity**
2. **@Table(name = "STUDENT")**
3. **public class Student {**
4. **@Id**
5. **@GenericGenerator(name="myGenerator", strategy="**identity **")**
6. **@GeneratedValue(strategy = GenerationType.*AUTO*, generator = "myGenerator")**
7. **@Column(name = "SNO")**
8. **private Integer sid;**
9. **@Column(name = "NAME")**
10. **private String name;**
11. **@Column(name = "COURSE", length = 20)**
12. **private String course;**
13. **@Column(name = "FEE")**
14. **private double fee;**
15. **// getters & setters**
16. **}**

**7.) native(identity/sequence/hilo)**

1. **@Entity**
2. **@Table(name = "STUDENT")**
3. **public class Student {**
4. **@Id**
5. **@GenericGenerator(**
6. **name="myGenerator",**
7. **strategy="**native**"**
8. **parameters**={
9. **@Parameter**(name="sequence",value="STUDENT\_SEQ"),
10. **@Parameter**(name="table", value="PK\_VALUE\_TAB"),
11. **@Parameter**(name="column", value="PK\_VALUE\_COLUMN")
12. }
13. **)**
14. **@GeneratedValue(strategy = GenerationType.*AUTO*, generator = "myGenerator")**
15. **@Column(name = "SNO")**
16. **private Integer sid;**
17. **@Column(name = "NAME")**
18. **private String name;**
19. **@Column(name = "COURSE", length = 20)**
20. **private String course;**
21. **@Column(name = "FEE")**
22. **private double fee;**
23. **// getters & setters**
24. **}**

**8.) uuid(org.hibernate.id.UUIDHexGenerator)**

1. **@Entity**
2. **@Table(name = "STUDENT")**
3. **public class Student {**
4. **@Id**
5. **@GenericGenerator(name="myGenerator", strategy="**uuid **")**
6. **@GeneratedValue(strategy = GenerationType.*AUTO*, generator = "myGenerator")**
7. **@Column(name = "SNO")**
8. **private String sid;**
9. **@Column(name = "NAME")**
10. **private String name;**
11. **@Column(name = "COURSE", length = 20)**
12. **private String course;**
13. **@Column(name = "FEE")**
14. **private double fee;**
15. **// getters & setters**
16. **}**

**NOTE:** @GenericGenerator and @GenericGenerators can be used in package level annotations, making them application level generators (just like if they were in a JPA XML file).

@GenericGenerators(

    {

    @GenericGenerator(

        name="hibseq",

        strategy = "seqhilo",

        parameters = {

            @Parameter(name="max\_lo", value = "5"),

            @Parameter(name="sequence", value="heybabyhey")

        }

     ),

     @GenericGenerator(...)

     }

)

**package** org.hibernate.test.model