Generator classes in Hibernate

The <generator> subelement of id used to generate the unique identifier for the objects of persistent class. There are many generator classes defined in the Hibernate Framework.

All the generator classes implements the **org.hibernate.id.IdentifierGenerator interface**. The application programmer may create one's own generator classes by implementing the IdentifierGenerator interface. Hibernate framework provides many built-in generator classes:

1. assigned
2. increment
3. sequence
4. hilo
5. native
6. identity
7. seqhilo
8. uuid
9. guid
10. select
11. foreign
12. sequence-identity

1) assigned

It is the default generator strategy if there is no <generator> element . In this case, application assigns the id. For example:

```
....
 <hibernate-mapping>
  <class ...>
    <id ...>
```

```
     <generator class="assigned"></generator>
    </id>


    .....


  </class>
 </hibernate-mapping>
```

2) increment

It generates the unique id only if no other process is inserting data into this table. It generates **short**, **int** or **long** type identifier. The first generated identifier is 1 normally and incremented as 1. Syntax:

```
....
 <hibernate-mapping>
  <class ...>
    <id ...>
     <generator class="increment"></generator>
    </id>


    .....


  </class>
 </hibernate-mapping>
```

3) sequence

It uses the sequence of the database. if there is no sequence defined, it creates a sequence automatically e.g. in case of Oracle database, it creates a sequence named HIBERNATE_SEQUENCE. In case of Oracle, DB2, SAP DB, Postgre SQL or McKoi, it uses sequence but it uses generator in interbase. Syntax:

```
.....
 <id ...>
```

```
  <generator class="sequence"></generator>
 </id>
 .....
```

For defining your own sequence, use the param subelement of generator.

```
.....
 <id ...>
  <generator class="sequence">
      <param name="sequence">your_sequence_name</param>
  </generator>
 </id>
 .....
```

4) hilo

It uses high and low algorithm to generate the id of type short, int and long. Syntax:

```
.....
 <id ...>
  <generator class="hilo"></generator>
 </id>
 .....
```

5) native

It uses identity, sequence or hilo depending on the database vendor. Syntax:

```
.....
 <id ...>
  <generator class="native"></generator>
 </id>
 .....
```

6) identity

It is used in Sybase, My SQL, MS SQL Server, DB2 and HypersonicSQL to support the id column. The returned id is of type short, int or long.

7) seqhilo

It uses high and low algorithm on the specified sequence name. The returned id is of type short, int or long.

8) uuid

It uses 128-bit UUID algorithm to generate the id. The returned id is of type String, unique within a network (because IP is used). The UUID is represented in hexadecimal digits, 32 in length.

9) guid

It uses GUID generated by database of type string. It works on MS SQL Server and MySQL.

10) select

It uses the primary key returned by the database trigger.

11) foreign

It uses the id of another associated object, mostly used with <one-to-one> association.

12) sequence-identity

It uses a special sequence generation strategy. It is supported in Oracle 10g drivers only.

download the example of assigned generator

← prev next →