

## Entities

It is the most basic class that composes the ORM service. It can compose the tables in the applications and has the properties of the column that includes tables.

### Required Elements

The [jpa setup](#) command provides the following options and attributes

#### [Essential] Entity annotation declaration ( or defining XML configuration file )

```
@Entity
public class User {
}
```

#### [Essential] Generator without Argument

```
public User(){
}
```

#### [Essential] generated as the best class and cannot be defined as enum,interface

#### [Essential] cannot be declared as the final class

#### [Essential] requires Primary Key: @Id Annotation stated

```
@Id
private String userId;
```

#### [Recommended] Serializable interface implementation

```
public class User implements Serializable {
    private static final long serialVersionUID = -8077677670915867738L;
}
```

#### [Recommended] defining getter and setter to assess property information

```
private String userName;

public String getUserName() {
    return userName;
}

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

### Major Annotations

Annotation consisting the entity is as following:

#### @Entity

Express that the corresponding class is Entity class and is indicated over the class declaration. If table name and entity name differ, indicate the relevant table name in the Name.

```
@Entity(name="USER_TB")
public class User {
```

```
}
```

## @Id

Express that the relevant Attribute is key and is indicated over Attribute.

```
@Id  
private String userId;
```

## @Column

To enter the column information mapped to the relevant Attribute, it is indicated over Attribute. No need to indicate if the column name and Attribute name do not match

```
@Column(name = "DEPT_NAME", length = 30)  
private String deptName;
```

## @OneToOne, @OneToMany, @ManyToOne, @ManyToMany

Indicated over Attribute defined to configure relationship between tables. Each represents relationship between 1:1,1:N,N:1,N:N. See [association mapping](#) for details.

```
@ManyToMany  
private Set<Role> roles = new HashSet(0);
```

## @Transient

Indicated over the Attribute to define the Attribute used without mapping to column of table.

```
@Transient  
private String roleName;
```

## Entity Status

- **New (transient)**: indicate the status where Entity object is initialized simply.
- **Managed (persistent)**: indicate the status when Entity is managed by Entity Manager.
- **Detached**: status when Entity object is not related to Persistence Context any more.
- **Removed**: status when Entity object managed is deleted.