JPA Creating an Entity

A Java class can be easily transformed into an entity. For transformation the basic requirements are: -

- No-argument Constructor
- Annotation

Here, we will learn how to transform a regular Java class into an entity class with the help of an example: -

Simple Student class

```
public class Student {

   private int id;
   private String name;
   private long fees;
   public Student() {}
   public Student(int id)
    {
      this.id = id;
       }
   public int getId()
   {
      return id;
       }
   public void setId(int id)
   {
      this.id = id;
       }
   public String getName()
   {
      return name;
       }
   public void setName(String name)
   {
```

```
        this.name = name;
        }
    public long getFees()
    {
        return fees;
        }
    public void setFees (long fees)
    {
        this.fees = fees;
    }
}
```

Above class is a regular java class having three attributes id, name and fees. To transform this class into an entity add @Entity and @Id annotation in it.

- **@Entity -** This is a marker annotation which indicates that this class is an entity. This annotation must be placed on the class name.
- **@Id -** This annotation is placed on a specific field that holds the persistent identifying properties. This field is treated as a primary key in database.

Simple Entity Class

```
import javax.persistence.*;
@Entity
public class Student {
    @Id
    private int id;
    private String name;
    private long fees;
    public Student() {}
    public Student(int id)
    {
        this.id = id;
        }
    public int getId()
    {
        return id;
```

```
        }
    public void setId(int id)
    {
        this.id = id;
        }
    public String getName()
    {
        return name;
        }
    public void setName(String name)
    {
        this.name = name;
        }
    public long getFees()
    {
        return fees;
        }
    public void setFees (long fees)
    {
        this.fees = fees;
    }
}
```