

# Fields vs Getters

In Java Persistence API (JPA), the annotations `@Access(AccessType.FIELD)` and `@Access(AccessType.PROPERTY)` are used to specify the access type for entity attributes within a persistent class. These annotations are typically applied to fields or methods within an entity class to indicate whether the JPA provider should access the attributes directly through fields or via getter/setter methods.

Here's a breakdown of their uses:

## 1. `**@Access(AccessType.FIELD)**`:

- When applied to a field in an entity class, it indicates that the JPA provider should access the field directly, bypassing any getter/setter methods.
- This means that the JPA provider will read and write the value of the field directly from/to the database, without invoking getter/setter methods.
- This access type is often preferred when the entity class mainly uses fields for attribute access, and there's no need for additional logic in getter/setter methods.

```
```java
@Entity
public class MyEntity {
    @Access(AccessType.FIELD)
    private String fieldName;
}
```
```

## 2. `**@Access(AccessType.PROPERTY)**`:

- When applied to a getter method or setter method in an entity class, it indicates that the JPA provider should access the attribute through the corresponding getter/setter methods.
- This means that the JPA provider will use the getter method to read the value from the attribute and the setter method to write the value to the attribute.
- This access type is often preferred when there's additional logic or validation required in getter/setter methods, or when using properties instead of fields for attribute access.

```
```java
@Entity
public class MyEntity {
    private String propertyName;

    @Access(AccessType.PROPERTY)
    public String getPropertyName() {
```

```

        return propertyName;
    }

    public void setPropertyName(String propertyName) {
        this.propertyName = propertyName;
    }
}
...

```

In summary, `@Access(AccessType.FIELD)` is used to specify field-level access for entity attributes, while `@Access(AccessType.PROPERTY)` is used to specify property-level access via getter/setter methods. The choice between these two access types depends on the design and requirements of the entity class.

We have 10 annotation on fields and 1 annotation on getter then jpa does not get the state of field from annotation but want getters for all.

So to allow both to work

We provide `AccessType.Field` on class and `AccessType.Property` on getter.

`@Entity(name = "orders")` changes the table name to orders and name of entity also.

So we should use it as

`@Entity`

`@Table(name = "orders")`