Working with Entities

Cătălin Tudose

PhD in Computer Science, Java and Web Technologies Expert

www.catalintudose.com | https://www.linkedin.com/in/catalin-tudose-847667a1

Overview

What is an entity?

How to map objects

Entity access types

Define entity primary keys and entity identity

What Is an Entity?

Annotated with @Entity or defined through XML configuration

Top-level class

Public or a protected no-arguments constructor

Final classes not recommended

What Does an Entity Support?

Inheritance

Polymorphic associations

Polymorphic queries

The Persistent State

Primitive types

Entity types

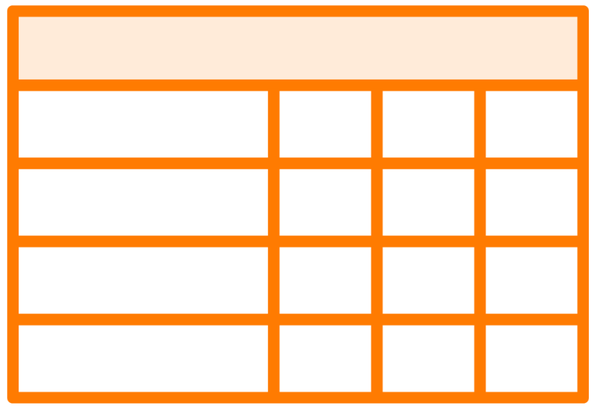
Embeddable types

Serializable types

Enums

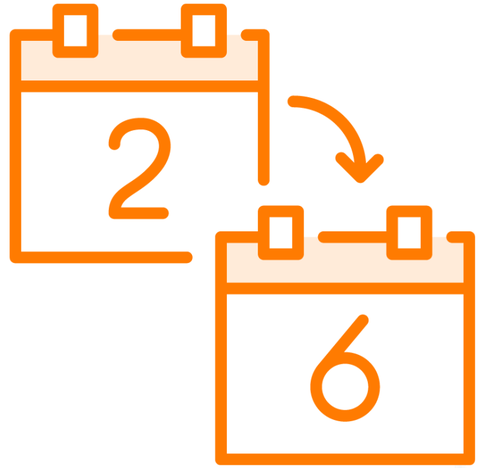
Collections

Mapping the Objects



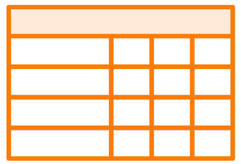
@Table

@SecondaryTable



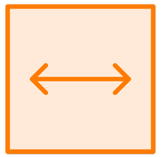
@SecondaryTables

The @Table Annotation



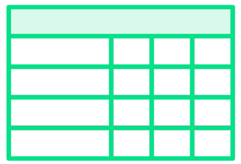
Primary table

Default values may apply



Parameters

The @SecondaryTable Annotation

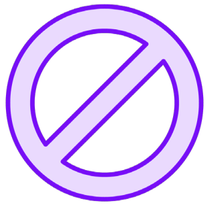


Primary table

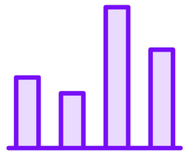
Default values may apply

Parameters

The @SecondaryTables Annotation



Primary table



Default values may apply

Parameters

Demo

A secondary table with one field

Passenger with an address

- The address has one field

Demo

A secondary table with multiple fields

Passenger with an address

- The address has multiple fields

Demo

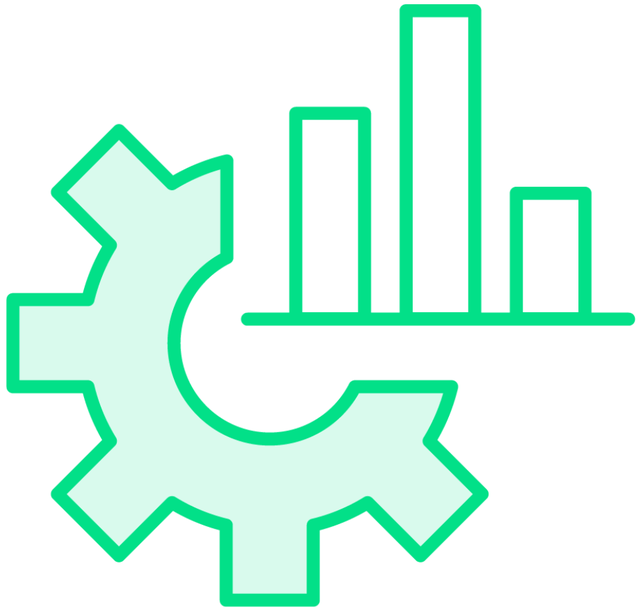
Multiple secondary tables

Passenger with address and phone

Entity Access Type

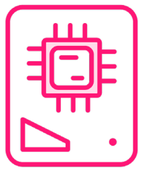
Accessing the Persistent State

Field access

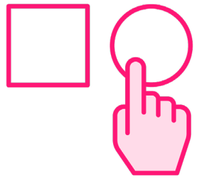


Property access

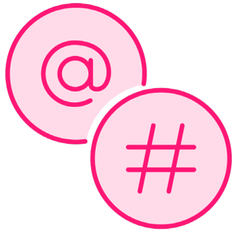
Entity Access Type



Method to access the persistent state of the entity



Single access type by default



Determined by placing the mapping annotations

Entity Access Type

```java
@Entity
@Table(name = "PASSENGERS")
public class Passenger {
    @Id
    @Column(name = "PASSENGER_ID")
    private int id;

    @Column(name = "PASSENGER_NAME", table = "PASSENGERS")
    private String name;
```

Entity Access Type

```
@Entity
@Table(name = "PASSENGERS")
public class Passenger {
     private int id;

     @Id
     @Column(name = "PASSENGER_ID")
     public int getId () {
          return id;
     }
```

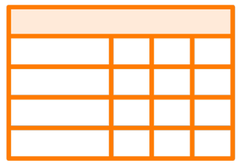
Entity Access Type

```
@Entity
@Table(name = "PASSENGERS")
public class Passenger {
    @Id
    private int id;

    private String name;
}
```

Entity Access Type

```
@Entity
@Table(name = "PASSENGERS")
public class Passenger {
    private int id;

    @Id
    public int getId() {
         return id;
    }
```

@Access (AccessType.FIELD)

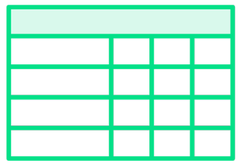


Default access type for the class is field-based

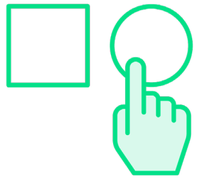
Persistent state accessed via the instance variables

@Access(AccessType.PROPERTY) for needed properties

@Access(AccessType.PROPERTY)



Default access type for the class is property-based



Persistent state accessed via the properties

@Access(AccessType.FIELD) for needed instance variables

Mixing Access Types

```
@Entity
@Table(name = "PASSENGERS")
@Access(AccessType.FIELD)
public class Passenger {
    @Id
    private int id;
    private String name;

    public int getId() {
        return id;
    }

    @Access(AccessType.PROPERTY)
    public String getName() {
        return name;
    }
```

Field-based vs. Property-based Access

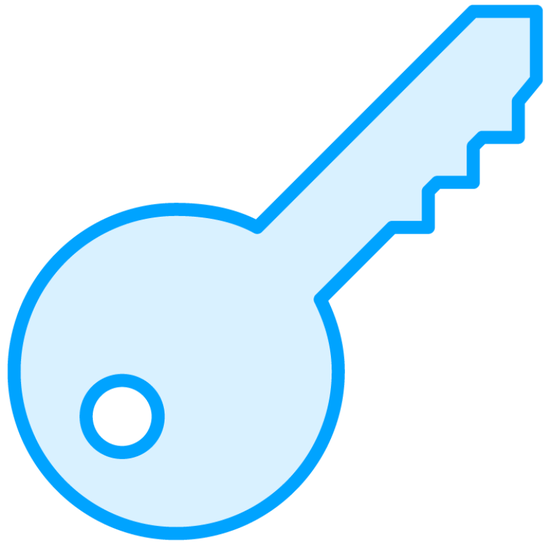
Omit the getter methods for the fields that should not be exposed

Easier readability using field-based access

Additional logic through accessors

Entity Primary Keys and Entity Identity

Defining Primary Keys



Simple Primary Key

Composite Primary Key

Rules for Composite Primary Keys

Public class, public no-arguments constructor

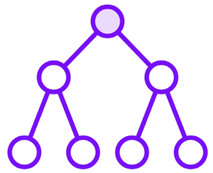
Serializable

equals() and hashCode()

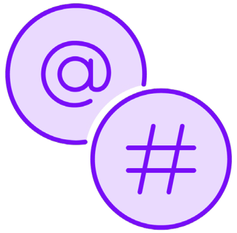
Represented as embeddable class or as an id class

Fields or properties correspondence

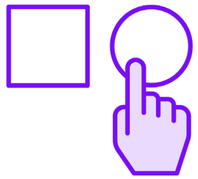
The @GeneratedValue Annotation



Generation strategy



Used in conjunction with @Id



May be applied to a persistent field or property

Demo

Primary key

@GeneratedValue

Demo

Primary key

Embeddable primary key/embedded id

Demo

Primary key

Embeddable primary key/id class

Summary

Entities

Mapping objects

Entity access types

- Field-based
- Property-based

Entity primary keys and entity identity

- @GeneratedValue
- Embeddable primary key/embedded id
- Embeddable primary key/id class