
SPRING FRAMEWORK
GURU

JPA Entity Relationships

Types of Relationships

- One to One - @OneToOne
  - One entity is related to one other entity
- One to Many - @OneToMany
  - One entity is related to many entities (List, Set, Map, SortedSet, SortedMap)

Types of Relationships

- Many to One - @ManyToOne
  - The inverse relationship of One to Many
- Many to Many - @ManyToMany
  - Many entities are related to many entities
  - Each has a List or Set reference to the other
  - A join table is used to define the relationships

Unidirectional vs Bidirectional

- Unidirectional is one-way
  - Mapping is only done one way. One side of the relationship will not know about the other
- Bidirectional is two way
  - Both sides know about each other
  - Generally recommended to use Bidirectional, since you can navigate the object graph in either direction

“Owning Side”

- The Owning side in the relationship will hold the foreign key in the database
- One to One is the side where the foreign key is specified
- OneToMany and ManyToOne is the ‘Many’ side
- ‘mappedBy’ is used to define the field with “owns” the reference of the relationship

Fetch Type

- Lazy Fetch Type - Data is not queried until referenced
- Eager Fetch Type - Data is queried up front
- Hibernate 5 Supports the JPA 2.1 Fetch Type Defaults
- JPA 2.1 Fetch Type Defaults:
  - OneToMany - Lazy
  - ManyToOne - Eager
  - ManyToMany - Lazy
  - OneToOne - Eager

JPA Cascade Types

- JPA Cascade Types Control how state changes are cascaded from parent objects to child objects.
- JPA Cascade Types:
  - PERSIST - Save operations will cascade to related entities
  - MERGE - related entities are merged when the owning entity is merged
  - REFRESH - related entities are refreshed when the owning entity is refreshed

JPA Cascade Types - Cont

- JPA Cascade Types (continued):
  - REMOVE - Removes all related entities when the owning entity is deleted
  - DETACH - detaches all related entities if a manual detach occurs
  - ALL - Applies all the above cascade options
- By default, no operations are cascaded

Embeddable Types

- JPA / Hibernate support embeddable types
- These are used to define a common set of properties
- For example, an order might have a billing address, and a shipping address
- An embeddable type could be used for the address properties

Inheritance

- MappedSuperclass - Entities inherit from a super class. A database table IS NOT created for the super class
- Single Table - (Hibernate Default) - One Table is used for all subclasses
- Joined Table - Base class and subclasses have their own tables. Fetching subclass entities require a join to the parent table
- Table Per Class - Each subclass has its own table

Create and Update Timestamps

- Often a best practice to use create and update timestamps on your entities for audit purposes
- JPA supports @PrePersist and @PreUpdate which can be used to support audit timestamps via JPA lifecycle callbacks
- Hibernate provides @CreationTimestamp and @UpdateTimestamp


SPRING FRAMEWORK
GURU