1. Tầng security:
   1. Filter -> authenticate -> filter api
   2. Servlet
   3. Intercepter -> keep authenticate.
      1. Prehandler: before controller execute
      2. Posthandler: after controller execute
      3. afterCompletion: when controller execute success.
   4. Controller.
2. Relationship of entity
3. Unidirectional
   1. One-to-one
      1. Customer ( foreign -> ) -> MemberCard
   2. One-to-many create new table, join two table, one is key , many is foreign key.

@JoinTable(name="ITEM\_IMAGES",

joinColumns=@JoinColumn(name="ITEM\_ID"),

1. inverseJoinColumns=@JoinColumn(name="IMAGE\_ID"))
2. Many-to-one
3. Many-to-many = one to many – many to one

@ManyToMany

@JoinTable(name="PERSON\_EVENT",

joinColumns=@JoinColumn(name="PERSON\_ID"),

1. inverseJoinColumns=@JoinColumn(name="EVENT\_ID"))

Bi-directional

* 1. One-to-one
     1. Customer ( @joinCollum@PrimaryJoinCollum foreign -> ) <-> MemberCard ( @OneToOne(Mappby – foreight customer)
  2. One-to-many
  3. Many-to-one
  4. Many-to-many @ManyToMany(mappby="events")

Embedded vs Embeddable

fetchType. Eager: whether the relationship is loaded when the entity is queried.

EAGER: load when query ( ONE to one & Many to oNE

LAZY: load when first access to attribute ( One to Many & Many to Many)

Cascade: do operation with one table, and relationship of that table.

PERSIST, MERGE, REMOVE, REFRESH

ALL.

1. Entity

New state: we create new row.

Manage state:we consider it exist in database

Detach state: we consider its just object, not persist . entity manager closed

Removed state: removed database

: query:

: name = bind paramater