=================================================================================

One-To-One - Unidirectional

@Entity

@Table(name="shampoos")

public class BasicShampoo {

...

@OneToOne

@JoinColumn(name = "label\_id", referencedColumnName = "id")

private BasicLabel label;

...

}

One-To-One - Bidirectional

@Entity

@Table(name = "labels")

public class BasicLabel imploments Label{

...

@OneToOne(mappedBy = "label", targetEntity = BasicShampoo.class)

private BasicShampoo basicShampoo;

...

}

=================================================================================

Many-To-One Unidirectional

@Entity

@Table(name = "shampoos")

public class BasicShampoo {

...

@ManyToOne

@JoinColumn(name="batch\_id", referencedColumName = "id")

private ProductionBatch batch;

...

}

Many-To-One Bidirectional

@Entity

@Table(name = "batches")

public class ProductionBatch {

...

@OneToMany(mappedBy = "batch", targetEntity = BasicShampoo.class,

fetch = FetchType.LAZY, cascade = CascadeType.ALL)

private Set<Shampoo> shampoos;

...

}

===================================================================================

Many-To-Many Unidirectional

@Table(name = "shampoos")

public class BasicShampoo {

...

@ManyToMany

@JoinTable(name = "shampoos\_ingredients",

joinColumns = @JoinColumn(name="shampoo\_id", referencedColumnName = "id"),

inverseJoinColumns = @JoinColumn(name="ingredient\_id", referencedColumName="id"))

private Set<BasicIngredient> ingredients;

...

}

Many-To-Many Bidirectional

@Entity

@Table(name = "ingredients")

public class BasicIngredient {

...

@ManyToMany(mappedBy = "ingredients", targetEntity = BasicShampoo.class)

private Set<BasicShampoo> shampoos;

...

}

=======================================================================================