8097534258 -Saurabh

**Hibernate validators** :

 **@NotNull**: Ensures the annotated element is not null.

 **@NotEmpty**: Validates that the annotated string, collection, map, or array is not null or empty.

 **@NotBlank**: Validates that the annotated string is not null and the trimmed length is greater than zero.

 **@Size**: Validates that the annotated element’s size or length falls within the specified boundaries.

 **@Min**: Validates that the annotated element is a number and its value must be greater than or equal to the specified minimum.

 **@Max**: Validates that the annotated element is a number and its value must be less than or equal to the specified maximum.

 **@DecimalMin**: Validates that the annotated element is a number and its value must be greater than or equal to the specified minimum, with a specific decimal precision.

 **@DecimalMax**: Validates that the annotated element is a number and its value must be less than or equal to the specified maximum, with a specific decimal precision.

 **@Negative**: Ensures that the annotated element is a strictly negative number.

 **@NegativeOrZero**: Ensures that the annotated element is a negative number or zero.

 **@Positive**: Ensures that the annotated element is a strictly positive number.

 **@PositiveOrZero**: Ensures that the annotated element is a positive number or zero.

 **@Digits**: Validates that the annotated element is a number within the accepted range of integer and fraction digits.

 **@Past**: Validates that the annotated element is a date in the past.

 **@PastOrPresent**: Validates that the annotated element is a date in the past or present.

 **@Future**: Validates that the annotated element is a date in the future.

 **@FutureOrPresent**: Validates that the annotated element is a date in the future or present.

 **@Pattern**: Validates that the annotated string matches the regular expression (regexp) provided.

 **@Email**: Validates that the annotated string is a valid email address.

 **@URL**: Validates that the annotated string is a valid URL.

 **@CreditCardNumber**: Validates that the annotated string is a valid credit card number.

 **@Length**: Validates that the annotated string’s length is between the specified min and max.

 **@Range**: Validates that the annotated element’s value falls between the specified min and max values.

 **@UniqueElements**: Validates that the annotated collection only contains unique elements.

 **@Valid**: Used for validating associated objects or collections of objects

**Entity Class Annotations** :

 **@Entity**: Specifies that the class is an entity and is mapped to a database table.

 **@Table**: Specifies the primary table for the annotated entity.

 **@Id**: Specifies the primary key of an entity.

 **@GeneratedValue**: Provides the specification of generation strategies for the values of primary keys.

 **@Column**: Specifies the mapped column for a persistent property or field.

 **@JoinColumn**: Specifies the column used for joining an entity association.

 **@OneToOne**: Defines a single-valued association to another entity that has a one-to-one multiplicity.

 **@OneToMany**: Defines a one-to-many association with another entity.

 **@ManyToOne**: Defines a many-to-one association with another entity.

 **@ManyToMany**: Defines a many-to-many association with another entity.

 **@JoinTable**: Specifies the join table used in the mapping of associations.

 **@Enumerated**: Specifies that a persistent property or field should be persisted as an enumerated type.

 **@Temporal**: Specifies that a persistent property or field should be persisted as a temporal type.

 **@Lob**: Specifies that a persistent property or field should be persisted as a large object to a database-supported large object type.

 **@Transient**: Specifies that a property or field is not persistent.

 **@Embedded**: Specifies a persistent field or property of an entity whose value is an instance of an embeddable class.

 **@Embeddable**: Specifies a class whose instances are stored as an intrinsic part of an owning entity and share the identity of the entity.

 **@EmbeddedId**: Specifies a composite primary key that is an embeddable class.

 **@MapKey**: Specifies a map key for an entity association.

 **@MapKeyColumn**: Specifies the column used as a map key for associations of type java.util.Map.

 **@SequenceGenerator**: Defines a primary key generator that may be referenced by name when a generator element is specified for the GeneratedValue annotation.

 **@NamedQuery**: Specifies a static, named query in the Java Persistence query language.

 **@NamedQueries**: Specifies multiple named queries.

 **@SqlResultSetMapping**: Specifies the mapping of the result of a native SQL query.

 **@Inheritance**: Specifies the inheritance strategy to be used for an entity class hierarchy.

 **@DiscriminatorColumn**: Specifies the column that will be used to differentiate between entity types in an inheritance hierarchy.

* **@DiscriminatorValue**: Specifies the value to be used for the discriminator column fo
* @version, @MappedSuperclass

@EntityListner:------ ++++++++++++++++++++++++++++++++++++++++++++++++++++

 **@PrePersist**: Invoked before the entity is persisted (added to the database).

 **@PostPersist**: Invoked after the entity is persisted.

 **@PreUpdate**: Invoked before the entity is updated.

 **@PostUpdate**: Invoked after the entity is updated.

 **@PreRemove**: Invoked before the entity is removed (deleted from the database).

 **@PostRemove**: Invoked after the entity is removed.

*  **@PostLoad**: Invoked after the entity is loaded from the database.