All JPA Annotations: Mapping Annotations

(/users/3439641/ramesh-fadatare.html) by
Ramesh Fadatare (/users/3439641/ramesh-fadatare.html) ·
Nov. 27, 18 · Java Zone (/java-jdk-development-tutorials-tools-news) · Presentation

This article provides you with 89 JPA mapping annotations for quick reference and/or summary. Let's get started!

Learn JPA in depth at Java Persistence API Tutorial (http://www.javaguides.net/p/jpa-tutorial-java-persistence-api.html)

JPA Annotations

1. @Access

2. @AssociationOverride

3. @AssociationOverrides

4. @AttributeOverride

5. @AttributeOverrides

6. @Basic

7. @Cacheable

8. @CollectionTable

9. @Column

10. @ColumnResult

11. @ConstructorResult

12. @Convert

13. @Converter

14. @Converts

15. @DiscriminatorColumn

16. @DiscriminatorValue

17. @ElementCollection

18. @Embeddable

19. @Embedded

20. @EmbeddedId

21. @Entity

22. @EntityListeners

23. @EntityResult

24. @Enumerated

25. @ExcludeDefaultListeners

26. @ExcludeSuperclassListeners

27. @FieldResult

28. @ForeignKey

29. @GeneratedValue

30. @Id

31. @IdClass

32. @Index

33. @Inheritance

34. @JoinColumn

35. @JoinColumns

36. @JoinTable

37. @Lob

38. @ManyToMany

39. @ManyToOne

40. @MapKey

41. @MapKeyClass

42. @MapKeyColumn

43. @MapKeyEnumerated

44. @MapKeyJoinColumn

45. @MapKeyJoinColumns

46. @MapKeyTemporal

47. @MappedSuperclass

48. @MapsId

49. @NamedAttributeNode

50. @NamedEntityGraph

51. @NamedEntityGraphs

52. @NamedNativeQueries

53. @NamedNativeQuery

54. @NamedQueries

55. @NamedQuery

56. @NamedStoredProcedureQueries

57. @NamedStoredProcedureQuery

58. @NamedSubgraph

58. @NamedSubgraph

59. @OneToMany

60. @OneToOne

61. @OrderBy

62. @OrderColumn

63. @PersistenceContext

64. @PersistenceContexts

65. @PersistenceProperty

66. @PersistenceUnit

67. @PersistenceUnits

68. @PostLoad

69. @PostPersist

70. @PostRemove

71. @PostUpdate

72. @PrePersist

73. @PreRemove

74. @PreUpdate

75. @PrimaryKeyJoinColumn

76. @PrimaryKeyJoinColumns

77. @QueryHint

78. @SecondaryTable

79. @SecondaryTables

80. @SequenceGenerator

81. @SqlResultSetMapping

82. @SqlResultSetMappings

83. @StoredProcedureParameter

84. @Table

85. @TableGenerator

86. @Temporal

87. @Transient

88. @UniqueConstraint

89. @Version

Let's describe each annotation with links to their Java doc and their official documentation sections.

@Access

The @Access (http://docs.oracle.com/javaee/7/api/javax/persistence/Access.html) annotation is used to specify the access type of the associated entity class, mapped superclass, or the embeddable class and entity attribute.

See the Access type (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#access) section for more info.

@AssociationOverride

The @AssociationOverride (http://docs.oracle.com/javaee/7/api/javax/persistence/AssociationOverride.html) annotation is used to override an association mapping (e.g. `@ManyToOne`, `@OneToOne`, `@OneToMany`, `@ManyToMany`) inherited from a mapped superclass or an embeddable.

See the Overriding Embeddable types (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#embeddable-override) section for more info.

@AssociationOverrides

The @AssociationOverrides (http://docs.oracle.com/javaee/7/api/javax/persistence/AssociationOverrides.html) is used to group @AssociationOverride

is used to group several @AssociationOverride (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-associationoverride) annotations.

@AttributeOverride

The @AttributeOverride (http://docs.oracle.com/javaee/7/api/javax/persistence/AttributeOverride.html) annotation is used to override an attribute mapping inherited from a mapped superclass or an embeddable.

See the Overriding Embeddable types (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#embeddable-override) section for more info.

@AttributeOverrides

The @AttributeOverrides (http://docs.oracle.com/javaee/7/api/javax/persistence/AttributeOverrides.html) is used to group several @AttributeOverride (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-attributeoverride) annotations.

@Basic

The @Basic (http://docs.oracle.com/javaee/7/api/javax/persistence/Basic.html) annotation is used to map a basic attribute type to a database column.

See the Basic types (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#basic) chapter for more info.

@Cacheable

The @Cacheable (http://docs.oracle.com/javaee/7/api/javax/persistence/Cacheable.html) annotation is used to specify whether an entity should be stored in the second-level cache.

See the Caching (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#caching) chapter for more info.

@CollectionTable

The @CollectionTable (http://docs.oracle.com/javaee/7/api/javax/persistence/CollectionTable.html) annotation is used to specify the database table that stores the values of a basic or an embeddable type collection.

See the Collections of embeddable types (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#embeddable-collections) section for more info.

@Column

The @Column (http://docs.oracle.com/javaee/7/api/javax/persistence/Column.html) annotation is used to specify the mapping between a basic entity attribute and the database table column.

See the @Column annotation (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#basic-column-annotation) section for more info.

@ColumnResult

The @ColumnResult (http://docs.oracle.com/javaee/7/api/javax/persistence/ColumnResult.html) annotation is used in conjunction with the @SqlResultSetMapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-sqlresultsetmapping) or @ConstructorResult (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-constructorresult) annotations to map a SQL column for a given SELECT query.

See the Entity associations with the named native queries (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-composite-key-entity-associations_named-query-example) section for more info.

@ConstructorResult

The @ConstructorResult (http://docs.oracle.com/javaee/7/api/javax/persistence/ConstructorResult.html) an

notation is used in conjunction with the @SqlResultSetMapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-sqlresultsetmapping) annotations to map columns of a given SELECT query to a certain object constructor.

See the Multiple scalar values NamedNativeQuery with ConstructorResult (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-multiple-scalar-values-dto-NamedNativeQuery-example) section for more info.

@Convert

The @Convert (http://docs.oracle.com/javaee/7/api/javax/persistence/Convert.html) annotation is used to specify the AttributeConverter (http://docs.oracle.com/javaee/7/api/javax/persistence/AttributeConverter.html) implementation used to convert the currently annotated basic attribute.

See the AttributeConverter (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#basic-enums-attribute-converter) section for more info.

24.1.13. @Converter

The @Converter (http://docs.oracle.com/javaee/7/api/javax/persistence/Converter.html) annotation is used to specify that the current annotate AttributeConverter (http://docs.oracle.com/javaee/7/api/javax/persistence/AttributeConverter.html) implementation can be used as a JPA basic attribute converter.

See the AttributeConverter (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#basic-enums-attribute-converter) section for more info.

@Converts

The @Converts (http://docs.oracle.com/javaee/7/api/javax/persistence/Converts.html) annotation is used to group multiple @Convert (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-convert) annotations.

See the AttributeConverter

See the AttributeConverter (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#basic-enums-attribute-converter) section for more info.

@DiscriminatorColumn

The @DiscriminatorColumn (http://docs.oracle.com/javaee/7/api/javax/persistence/DiscriminatorColumn.html) annotation is used to specify the discriminator column name and the discriminator type (http://docs.oracle.com/javaee/7/api/javax/persistence/DiscriminatorColumn.html#discriminatorType--) for the SINGLE_TABLE and JOINED Inheritance strategies.

See the Discriminator (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#entity-inheritance-discriminator) section for more info.

@DiscriminatorValue

The @DiscriminatorValue (http://docs.oracle.com/javaee/7/api/javax/persistence/DiscriminatorValue.html) annotation is used to specify what value of the discriminator column is used for mapping the currently annotated entity.

See the Discriminator (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#entity-inheritance-discriminator) section for more info.

@ElementCollection

The @ElementCollection (http://docs.oracle.com/javaee/7/api/javax/persistence/ElementCollection.html) annotation is used to specify a collection of basic or embeddable types.

See the Collections (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections) section for more info.

@Embeddable

The @Embeddable (http://docs.oracle.com/javaee/7/api/javax/persistence/Embeddable.html) annotation is used to specify embeddable types. Like basic types, embeddable types do not

have any identity, being managed by their owning entity.

See the Embeddables (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#embeddables) section for more info.

@Embedded

The @Embedded (http://docs.oracle.com/javaee/7/api/javax/persistence/Embedded.html) annotation is used to specify that a given entity attribute represents an embeddable type.

See the Embeddables (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#embeddables) section for more info.

@EmbeddedId

The @EmbeddedId (http://docs.oracle.com/javaee/7/api/javax/persistence/EmbeddedId.html) annotation is used to specify the entity identifier is an embeddable type.

See the Composite identifiers with @EmbeddedId (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#identifiers-composite-aggregated) section for more info.

@Entity

The @Entity (http://docs.oracle.com/javaee/7/api/javax/persistence/Entity.html) annotation is used to specify that the currently annotate class represents an entity type. Unlike basic and embeddable types, entity types have an identity and their state is managed by the underlying Persistence Context.

See the Entity (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#entity) section for more info.

@EntityListeners

The @EntityListeners (http://docs.oracle.com/javaee/7/api/javax/persistence/EntityListeners.html) annotation is used to specify an array of callback listener classes that are used by the

currently annotated entity.

See the JPA callbacks (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-jpa-callbacks-example) section for more info.

@EntityResult

The @EntityResult (http://docs.oracle.com/javaee/7/api/javax/persistence/EntityResult.html) annotation is used with the @SqlResultSetMapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-sqlresultsetmapping) annotation to map the selected columns to an entity.

See the Entity associations with the named native queries (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-composite-key-entity-associations_named-query-example) section for more info.

@Enumerated

The @Enumerated (http://docs.oracle.com/javaee/7/api/javax/persistence/Enumerated.html) annotation is used to specify that an entity attribute represents an enumerated type.

See the @Enumerated basic type (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#basic-enums-Enumerated) section for more info.

@ExcludeDefaultListeners

The @ExcludeDefaultListeners (http://docs.oracle.com/javaee/7/api/javax/persistence/ExcludeDefaultListeners.html) annotation is used to specify that the currently annotated entity skips the invocation of any default listener.

See the Exclude default entity listeners (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-exclude-default-listener) section for more info.

@ExcludeSuperclassListeners

The @ExcludeSuperclassListeners (http://docs.oracle.com/javaee/7/api/javax/persistence/ExcludeSuperclassListeners.html) annotation is used to specify that the currently annotated entity skips the invocation of listeners declared by its superclass.

See the Exclude default entity listeners (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-exclude-default-listener) section for more info.

@FieldResult

The @FieldResult (http://docs.oracle.com/javaee/7/api/javax/persistence/FieldResult.html) annotation is used with the @EntityResult (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-entityresult) annotation to map the selected columns to the fields of some specific entity.

See the Entity associations with the named native queries (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-composite-key-entity-associations_named-query-example) section for more info.

@ForeignKey

The @ForeignKey (http://docs.oracle.com/javaee/7/api/javax/persistence/ForeignKey.html) annotation is used to specify the associated foreign key of a @JoinColumn (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-joincolumn) mapping. The `@ForeignKeyannotation` is only used if the automated schema generation tool is enabled. In which case, it allows you to customize the underlying foreign key definition.

See the @ManyToOne with @ForeignKey (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#associations-many-to-one-example) section for more info.

@GeneratedValue

The @GeneratedValue (http://docs.oracle.com/javaee/7/api/javax/persistence/GeneratedValue.html) annotation specifies that the entity identifier value is automatically generated using an

tation specifies that the entity identifier value is automatically generated using an identity column, a database sequence, or a table generator. Hibernate supports the `@GeneratedValue` mapping even for UUIDidentifiers.

See the Automatically-generated identifiers (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#identifiers-simple-generated) section for more info.

@Id

The @Id (http://docs.oracle.com/javaee/7/api/javax/persistence/Id.html) annotation specifies the entity identifier. An entity must always have an identifier attribute, which is used when loading the entity in a given Persistence Context.

See the Identifiers (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#identifiers) section for more info.

@IdClass

The @IdClass (http://docs.oracle.com/javaee/7/api/javax/persistence/IdClass.html) annotation is used if the current entity defined a composite identifier. A separate class encapsulates all the identifier attributes, which are mirrored by the current entity mapping.

See the Composite identifiers with @IdClass (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#identifiers-composite-nonaggregated) section for more info.

@Index

The @Index (http://docs.oracle.com/javaee/7/api/javax/persistence/Index.html) annotation is used by the automated schema generation tool to create a database index.

See the Columns index (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#schema-generation-columns-index) chapter for more info.

24.1.33. @Inheritance

The @Inheritance (http://docs.oracle.com/javaee/7/api/javax/persistence/Inheritance.html) annotatio

(http://docs.oracle.com/javaee/7/api/javax/persistence/Inheritance.html) annotation is used to specify the inheritance strategy of a given entity class hierarchy.

See the Inheritance (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#entity-inheritance) section for more info.

@JoinColumn

The @JoinColumn (http://docs.oracle.com/javaee/7/api/javax/persistence/JoinColumn.html) annotation is used to specify the FOREIGN KEY column used when joining an entity association or an embeddable collection.

See the @ManyToOne with @JoinColumn (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#associations-many-to-one-example) section for more info.

@JoinColumns

The @JoinColumns (http://docs.oracle.com/javaee/7/api/javax/persistence/JoinColumns.html) annotation is used to group multiple @JoinColumn (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-joincolumn) annotations, which are used when mapping entity association or an embeddable collection using a composite identifier

@JoinTable

The @JoinTable (http://docs.oracle.com/javaee/7/api/javax/persistence/JoinTable.html) annotation is used to specify the link table between two other database tables.

See the @JoinTable mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-map-unidirectional-example) section for more info.

@Lob

The @Lob (http://docs.oracle.com/javaee/7/api/javax/persistence/Lob.html) annotation is used to specify that the currently annotated entity attribute represents a large object type.

See the BLOB mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#basic-blob-example) section for more info.

@ManyToMany

The @ManyToMany (http://docs.oracle.com/javaee/7/api/javax/persistence/ManyToMany.html) annotation is used to specify a many-to-many database relationship.

See the @ManyToMany mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#associations-many-to-many) section for more info.

@ManyToOne

The @ManyToOne (http://docs.oracle.com/javaee/7/api/javax/persistence/ManyToOne.html) annotation is used to specify a many-to-one database relationship.

See the @ManyToOne mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#associations-many-to-one) section for more info.

@MapKey

The @MapKey (http://docs.oracle.com/javaee/7/api/javax/persistence/MapKey.html) annotation is used to specify the key of a java.util.Map association for which the key type is either the primary key or an attribute of the entity that represents the value of the map.

See the @MapKey mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-map-unidirectional-example) section for more info.

@MapKeyClass

The @MapKeyClass (http://docs.oracle.com/javaee/7/api/javax/persistence/MapKeyClass.html) annotation is used to specify the type of the map key of a java.util.Map associations.

See the @MapKeyClass mapping

(http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-map-key-class) section for more info.

@MapKeyColumn

The @MapKeyColumn (http://docs.oracle.com/javaee/7/api/javax/persistence/MapKeyColumn.html) annotation is used to specify the database column, which stores the key of a java.util.Mapassociation for which the map key is a basic type.

See the @MapKeyType mapping section (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-map-custom-key-type-mapping-example) for an example of `@MapKeyColumn` annotation usage.

@MapKeyEnumerated

The @MapKeyEnumerated (http://docs.oracle.com/javaee/7/api/javax/persistence/MapKeyEnumerated.html) annotation is used to specify that the key of java.util.Map association is a Java Enum.

See the @MapKeyEnumerated mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-map-bidirectional-example) section for more info.

@MapKeyJoinColumn

The @MapKeyJoinColumn (http://docs.oracle.com/javaee/7/api/javax/persistence/MapKeyJoinColumn.html) annotation is used to specify that the key of java.util.Map association is an entity association. The map key column is a FOREIGN KEY in a link table that also joins the Map owner's table with the table where the Map value resides.

See the @MapKeyJoinColumn mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-map-value-type-entity-key-example) section for more info.

@MapKeyJoinColumns

The @MapKeyJoinColumns (http://docs.oracle.com/javaee/7/api/javax/persistence/MapKeyJoinColumns.html) annotation is used to group several @MapKeyJoinColumn (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_G

uide.html#annotations-jpa-mapkeyjoincolumn) mappings when the java.util.Map association key uses a composite identifier.

@MapKeyTemporal

The @MapKeyTemporal (http://docs.oracle.com/javaee/7/api/javax/persistence/MapKeyTemporal.html) annotation is used to specify that the key of java.util.Map association is a @TemporalType (http://docs.oracle.com/javaee/7/api/javax/persistence/TemporalType.html) (e.g. DATE, TIME, TIMESTAMP).

See the @MapKeyTemporal mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-map-unidirectional-example) section for more info.

@MappedSuperclass

The @MappedSuperclass (http://docs.oracle.com/javaee/7/api/javax/persistence/MappedSuperclass.html) annotation is used to specify that the currently annotated type attributes are inherited by any subclass entity.

See the @MappedSuperclass (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#entity-inheritance-mapped-superclass) section for more info.

@MapsId

The @MapsId (http://docs.oracle.com/javaee/7/api/javax/persistence/MapsId.html) annotation is used to specify that the entity identifier is mapped by the currently annotated `@ManyToOne` or `@OneToOne` association.

See the @MapsId mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#identifiers-derived-mapsid) section for more info.

@NamedAttributeNode

The @NamedAttributeNode (http://docs.oracle.com/javaee/7/api/javax/persistence/NamedAttributeNode.html

J annotation is used to specify each individual attribute node that needs to be fetched by an Entity Graph.

See the Fetch graph (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#fetching-strategies-dynamic-fetching-entity-graph-example) section for more info.

@NamedEntityGraph

The @NamedEntityGraph (http://docs.oracle.com/javaee/7/api/javax/persistence/NamedEntityGraph.html) annotation is used to specify an Entity Graph that can be used by an entity query to override the default fetch plan.

See the Fetch graph (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#fetching-strategies-dynamic-fetching-entity-graph-example) section for more info.

@NamedEntityGraphs

The @NamedEntityGraphs (http://docs.oracle.com/javaee/7/api/javax/persistence/NamedEntityGraphs.html) annotation is used to group multiple @NamedEntityGraph (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-namedentitygraph) annotations.

24.1.52. @NamedNativeQueries

The @NamedNativeQueries (http://docs.oracle.com/javaee/7/api/javax/persistence/NamedNativeQueries.html) annotation is used to group multiple @NamedNativeQuery (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-namednativequery) annotations.

See the Custom CRUD mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-custom-crud-example) section for more info.

@NamedNativeQuery

The @NamedNativeQuery

(http://docs.oracle.com/javaee/7/api/javax/persistence/NamedNativeQuery.html) annotation is used to specify a native SQL query that can be retrieved later by its name.

See the Custom CRUD mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-custom-crud-example) section for more info.

@NamedQueries

The @NamedQueries (http://docs.oracle.com/javaee/7/api/javax/persistence/NamedQueries.html) annotation is used to group multiple @NamedQuery (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-namedquery) annotations.

@NamedQuery

The @NamedQuery (http://docs.oracle.com/javaee/7/api/javax/persistence/NamedQuery.html) annotation is used to specify a JPQL query that can be retrieved later by its name.

See the @NamedQuery (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#jpql-api-named-query-example) section for more info.

@NamedStoredProcedureQueries

The @NamedStoredProcedureQueries (http://docs.oracle.com/javaee/7/api/javax/persistence/NamedStoredProcedureQueries.html) annotation is used to group multiple @NamedStoredProcedureQuery (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-namedstoredprocedurequery) annotations.

@NamedStoredProcedureQuery

The @NamedStoredProcedureQuery (http://docs.oracle.com/javaee/7/api/javax/persistence/NamedStoredProcedureQuery.html) annotation is used to specify a stored procedure query that can be retrieved later by its name.

See the Using named queries to call stored procedures (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_G

uide.html#sql-sp-named-query) section for more info.

@NamedSubgraph

The @NamedSubgraph (http://docs.oracle.com/javaee/7/api/javax/persistence/NamedSubgraph.html) annotation used to specify a subgraph in an Entity Graph.

See the Fetch subgraph (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#fetching-strategies-dynamic-fetching-entity-subgraph) section for more info.

@OneToMany

The @OneToMany (http://docs.oracle.com/javaee/7/api/javax/persistence/OneToMany.html) annotation is used to specify a one-to-many database relationship.

See the @OneToMany mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#associations-one-to-many) section for more info.

@OneToOne

The @OneToOne (http://docs.oracle.com/javaee/7/api/javax/persistence/OneToOne.html) annotation is used to specify a one-to-one database relationship.

See the @OneToOne mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#associations-one-to-one) section for more info.

@OrderBy

The @OrderBy (http://docs.oracle.com/javaee/7/api/javax/persistence/OrderBy.html) annotation is used to specify the entity attributes used for sorting when fetching the currently annotated collection.

See the @OrderBy mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-unidirectional-ordered-list) section for more info.

@OrderColumn

The @OrderColumn (http://docs.oracle.com/javaee/7/api/javax/persistence/OrderColumn.html) annotation is used to specify that the current annotation collection order should be materialized in the database.

See the @OrderColumn mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#collections-unidirectional-ordered-list-order-column-example) section for more info.

@PersistenceContext

The @PersistenceContext (http://docs.oracle.com/javaee/7/api/javax/persistence/PersistenceContext.html) annotation is used to specify the `EntityManager` that needs to be injected as a dependency.

See the @PersistenceContext mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#bootstrap-jpa-compliant-PersistenceContext-example) section for more info.

@PersistenceContexts

The @PersistenceContexts (http://docs.oracle.com/javaee/7/api/javax/persistence/PersistenceContexts.html) annotation is used to group multiple @PersistenceContext (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-persistencecontext) annotations.

@PersistenceProperty

The @PersistenceProperty (http://docs.oracle.com/javaee/7/api/javax/persistence/PersistenceProperty.html) annotation is used by the @PersistenceContext (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-persistencecontext) annotation to declare JPA provider properties that are passed to the underlying container when the `EntityManager` instance is created.

See the @PersistenceProperty mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#bootstrap-jpa-compliant-PersistenceContext-configurable-example) section for more info.

@PersistenceUnit

The @PersistenceUnit (http://docs.oracle.com/javaee/7/api/javax/persistence/PersistenceUnit.html) annotation is used to specify the `EntityManagerFactory` that needs to be injected as a dependency.

See the @PersistenceUnit mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#bootstrap-jpa-compliant-PersistenceUnit-example) section for more info.

@PersistenceUnits

The @PersistenceUnits (http://docs.oracle.com/javaee/7/api/javax/persistence/PersistenceUnits.html) annotation is used to group multiple @PersistenceUnit (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-persistenceunit) annotations.

@PostLoad

The @PostLoad (http://docs.oracle.com/javaee/7/api/javax/persistence/PostLoad.html) annotation is used to specify a callback method that fires after an entity is loaded.

See the JPA callbacks (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-jpa-callbacks-example) section for more info.

@PostPersist

The @PostPersist (http://docs.oracle.com/javaee/7/api/javax/persistence/PostPersist.html) annotation is used to specify a callback method that fires after an entity is persisted.

See the JPA callbacks (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_G

uide.html#events-jpa-callbacks-example) section for more info.

@PostRemove

The @PostRemove (http://docs.oracle.com/javaee/7/api/javax/persistence/PostRemove.html) annotation is used to specify a callback method that fires after an entity is removed.

See the JPA callbacks (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-jpa-callbacks-example) section for more info.

@PostUpdate

The @PostUpdate (http://docs.oracle.com/javaee/7/api/javax/persistence/PostUpdate.html) annotation is used to specify a callback method that fires after an entity is updated.

See the JPA callbacks (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-jpa-callbacks-example) section for more info.

@PrePersist

The @PrePersist (http://docs.oracle.com/javaee/7/api/javax/persistence/PrePersist.html) annotation is used to specify a callback method that fires before an entity is persisted.

See the JPA callbacks (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-jpa-callbacks-example) section for more info.

@PreRemove

The @PreRemove (http://docs.oracle.com/javaee/7/api/javax/persistence/PreRemove.html) annotation is used to specify a callback method that fires before an entity is removed.

See the JPA callbacks (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-jpa-callbacks-example) section for more info.

@PreUpdate

The @PreUpdate (http://docs.oracle.com/javaee/7/api/javax/persistence/PreUpdate.html) annotation is used to specify a callback method that fires before an entity is updated.

See the JPA callbacks (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-jpa-callbacks-example) section for more info.

@PrimaryKeyJoinColumn

The @PrimaryKeyJoinColumn (http://docs.oracle.com/javaee/7/api/javax/persistence/PrimaryKeyJoinColumn.html) annotation is used to specify that the primary key column of the currently annotated entity is also a foreign key to some other entity (e.g. a base class table in a JOINED inheritance strategy, the primary table in a secondary table mapping, or the parent table in a `@OneToOne` relationship).

See the @PrimaryKeyJoinColumn mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#identifiers-derived-primarykeyjoincolumn) section for more info.

@PrimaryKeyJoinColumns

The @PrimaryKeyJoinColumns (http://docs.oracle.com/javaee/7/api/javax/persistence/PrimaryKeyJoinColumns.html) annotation is used to group multiple @PrimaryKeyJoinColumn (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-primarykeyjoincolumn) annotations.

@QueryHint

The @QueryHint (http://docs.oracle.com/javaee/7/api/javax/persistence/QueryHint.html) annotation is used to specify a JPA provider hint used by a `@NamedQuery` or a `@NamedNativeQueryannotation`.

See the @QueryHint (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#jpa-read-only-entities-native-example) section for more info.

@SecondaryTable

The @SecondaryTable

The @SecondaryTable (http://docs.oracle.com/javaee/7/api/javax/persistence/SecondaryTable.html) annotation is used to specify a secondary table for the currently annotated entity.

See the @SecondaryTable mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-custom-crud-secondary-table-example) section for more info.

@SecondaryTables

The @SecondaryTables (http://docs.oracle.com/javaee/7/api/javax/persistence/SecondaryTables.html) annotation is used to group multiple @SecondaryTable (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-secondarytable) annotations.

@SequenceGenerator

The @SequenceGenerator (http://docs.oracle.com/javaee/7/api/javax/persistence/SequenceGenerator.html) annotation is used to specify the database sequence used by the identifier generator of the currently annotated entity.

See the @SequenceGenerator mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#identifiers-generators-sequence-configured) section for more info.

@SqlResultSetMapping

The @SqlResultSetMapping (http://docs.oracle.com/javaee/7/api/javax/persistence/SqlResultSetMapping.html) annotation is used to specify the `ResultSet` mapping of a native SQL query or stored procedure.

See the SqlResultSetMapping mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-composite-key-entity-associations_named-query-example) section for more info.

@SqlResultSetMappings

The @SqlResultSetMappings (http://docs.oracle.com/javaee/7/api/javax/persistence/SqlResultSetMappings.html

) annotation is used to group multiple @SqlResultSetMapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-sqlresultsetmapping) annotations.

@StoredProcedureParameter

The @StoredProcedureParameter (http://docs.oracle.com/javaee/7/api/javax/persistence/StoredProcedureParameter.html) annotation is used to specify a parameter of a @NamedStoredProcedureQuery (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#annotations-jpa-namedstoredprocedurequery).

See the Using named queries to call stored procedures (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-sp-named-query) section for more info.

@Table

The @Table (http://docs.oracle.com/javaee/7/api/javax/persistence/Table.html) annotation is used to specify the primary table of the currently annotated entity.

See the @Table mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#sql-custom-crud-secondary-table-example) section for more info.

@TableGenerator

The @TableGenerator (http://docs.oracle.com/javaee/7/api/javax/persistence/TableGenerator.html) annotation is used to specify the database table used by the identity generator of the currently annotated entity.

See the @TableGenerator mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#identifiers-generators-table-configured-mapping-example) section for more info.

@Temporal

The @Temporal (http://docs.oracle.com/javaee/7/api/javax/persistence/Temporal.html) annotation

is used to specify the `TemporalType` of the currently annotated java.util.Date or java.util.Calendar entity attribute.

See the Basic temporal types (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#basic-datetime) chapter for more info.

@Transient

The @Transient (http://docs.oracle.com/javaee/7/api/javax/persistence/Transient.html) annotation is used to specify that a given entity attribute should not be persisted.

See the @Transient mapping (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#events-jpa-callbacks-example) section for more info.

@UniqueConstraint

The @UniqueConstraint (http://docs.oracle.com/javaee/7/api/javax/persistence/UniqueConstraint.html) annotation is used to specify a unique constraint to be included by the automated schema generator for the primary or secondary table associated with the currently annotated entity.

See the Columns unique constraint (http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#schema-generation-columns-unique-constraint) chapter for more info.

@Version

The @Version (http://docs.oracle.com/javaee/7/api/javax/persistence/Version.html) annotation is used to specify the version attribute used for optimistic locking.

Topics: JAVA, ANNOTATIONS, JPA, MAPPING, JPA MAPPING ANNOTATIONS, MAPPING ANNOTATIONS, CODE

Popular on DZone

- **Cloud Computing Security Parameters on Various Cloud Platforms (/articles/cloud-computing-security-parameters-on-various-cloud-platforms?fromrel=true)**
- **8 Easy Steps To Set Up Multiple Git Accounts (Cheat Sheet) (/articles/8-easy-steps-to-set-up-multiple-git-accounts-cheat?fromrel=true)**
- **Multiple Spring Boot Applications in the Same Project (/articles/multiple-spring-boot-applications-in-the-same-proj?fromrel=true)**
- **Why General Inheritance Is Flawed and How to Finally Fix It! (/articles/why-general-inheritance-is-flawed-and-how-to-final?fromrel=true)**




x