

# Anotações do JPA

Vamos descrever cada anotação com links para o seu documento Java e suas seções documentação oficial.

## @Access

A anotação `@Access` é usada para especificar o tipo de acesso da classe de entidade associada, superclasse mapeada ou atributo de classe e entidade incorporável.

## @AssociationOverride

A anotação `@AssociationOverride` é usada para substituir um mapeamento de associação (por exemplo, `@ManyToOne`, `@OneToOne`, `@OneToMany`, `@ManyToMany`) herdado de uma superclasse mapeada ou de um incorporável.

## @AssociationOverrides

O `@AssociationOverrides` é usado para agrupar várias anotações `@AssociationOverride`.

## @AttributeOverride

A anotação `@AttributeOverride` é usada para substituir um mapeamento de atributo herdado de uma superclasse mapeada ou de um incorporável.

## @AttributeOverrides

O `@AttributeOverrides` é usado para agrupar várias anotações `@AttributeOverride`.

## @Basic

A anotação `@Basic` é usada para mapear um tipo de atributo básico para uma coluna do banco de dados.

## @Cacheable

A anotação `@Cacheable` é usada para especificar se uma entidade deve ser armazenada no cache de segundo nível.

## @CollectionTable

A anotação `@CollectionTable` é usada para especificar a tabela de banco de dados que armazena os valores de uma coleção de tipos básica ou incorporável.

## @Column

A anotação `@Column` é usada para especificar o mapeamento entre um atributo de entidade básico e a coluna da tabela de banco de dados.

## @ColumnResult

A anotação `@ColumnResult` é usada em conjunto com as anotações `@SqlResultSetMapping` ou `@ConstructorResult` para mapear uma coluna SQL para uma determinada consulta SELECT.

## @ConstructorResult

A anotação `@ConstructorResult` é usada em conjunto com as anotações `@SqlResultSetMapping` para mapear colunas de uma determinada consulta SELECT para um determinado construtor de objeto.

## @Convert

A anotação `@Convert` é usada para especificar a implementação de `AttributeConverter` usada para converter o atributo básico atualmente anotado.

## @Converter

A anotação `@Converter` é usada para especificar que a implementação do atributo atual `AttributeConverter` pode ser usada como um conversor de atributo básico JPA.

## @Converts

A anotação `@Converts` é usada para agrupar várias anotações `@Convert`.

## @DiscriminatorColumn

A anotação `@DiscriminatorColumn` é usada para especificar o nome da coluna discriminadora e o tipo discriminador para as estratégias `SINGLE_TABLE` e `JOINED`.

## @DiscriminatorValue

A anotação `@DiscriminatorValue` é usada para especificar qual valor da coluna discriminada é usado para mapear a entidade atualmente anotada.

# @ElementCollection

A anotação @ElementCollection é usada para especificar uma coleção de tipos básicos ou incorporáveis.

# @Embeddable

A anotação @Embeddable é usada para especificar tipos incorporáveis. Como os tipos básicos, os tipos incorporáveis não tem identidade, sendo gerenciados por sua entidade proprietária.

# @Embedded

A anotação @Embedded é usada para especificar que um determinado atributo de entidade representa um tipo incorporável.

# @EmbeddedId

A anotação @EmbeddedId é usada para especificar que o identificador de entidade é um tipo incorporável.

# @Entity

A anotação @Entity é usada para especificar que a classe anotada atualmente representa um tipo de entidade. Ao contrário dos tipos básicos e incorporável, os tipos de entidade tem uma identidade e seu estado é gerenciado pelo contexto de persistência subjacente.

# @EntityListeners

A anotação @EntityListeners é usada para especificar uma matriz de classes de ouvintes de retorno de chamada que são usadas pela entidade anotada atualmente.

# @EntityResult

A anotação @EntityResult é usada com a anotação @SqlResultSetMapping para mapear as colunas selecionadas para uma entidade.

# @Enumerated

A anotação @Enumerated é usada para especificar que um atributo de entidade representa um tipo enumerado.

## @ExcludeDefaultListeners

A anotação `@ExcludeDefaultListeners` é usada para especificar que a entidade atualmente anotada pula a invocação de qualquer ouvinte padrão;

## @ExcludeSuperclassListeners

A anotação `@ExcludeSuperclassListeners` é usada para especificar que a entidade atualmente anotada pula a invocação de ouvintes declarados por sua superclasse.

## @FieldResult

A anotação `@FieldResult` é usada com a anotação `@EntityResult` para mapear as colunas selecionadas para os campos de alguma entidade específica.

## @ForeignKey

A anotação `@ForeignKey` é usada para especificar a chave estrangeira associada ao mapeamento `@JoinColumn`. A `@ForeignKeyannotation` é usada apenas se a ferramenta de geração automática de esquema estiver ativada. Nesse caso, ele permite que você personalize a definição de chave estrangeira subjacente.

## @GeneratedValue

A anotação `@GeneratedValue` especifica que o valor do identificador de entidade é gerado automaticamente usando uma coluna de identidade, uma sequência de banco de dados ou um gerador de tabelas. O Hibernate suporta o mapeamento `@GeneratedValue` mesmo para os identificadores de UUID.

## @Id

A anotação `@Id` especifica o identificador da entidade. Uma entidade deve sempre ter um atributo identificador, que é usado ao carregar a entidade em um determinado contexto de persistência.

## @IdClass

A anotação `@IdClass` é usada se a entidade atual definiu um identificador composto. Uma classe separada encapsula todos os atributos do identificador, que são espelhados pelo mapeamento de entidade atual.

## @Index

A anotação `@Index` é usada pela ferramenta de geração automática de esquemas para criar um índice do banco de dados.

## **@Inheritance**

A anotação `@Inheritance` é usada para especificar a estratégia de herança de uma determinada hierarquia de classes de entidade.

## **@JoinColumn**

A anotação `@JoinColumn` é usada para especificar a coluna FOREIGN KEY usada ao ingressar em uma associação de entidades ou em uma coleção incorporável.

## **@JoinColumns**

A anotação `@JoinColumns` é usada para agrupar várias anotações `@JoinColumn`, que são usadas ao mapear a associação de entidades ou uma coleção incorporável usando um identificador composto.

## **@JoinTable**

A anotação `@JoinTable` é usada para especificar a tabela de links entre duas outras tabelas de banco de dados.

## **@Lob**

A anotação `@Lob` é usada para especificar que o atributo de entidade atualmente anotado representa um tipo de objeto grande.

## **@ManyToMany**

A anotação `@ManyToMany` é usada para especificar um relacionamento de banco muitos para muitos.

## **@ManyToOne**

A anotação `@ManyToOne` é usada para especificar um relacionamento de banco de dados muitos-para-um.

## **@Mapkey**

A anotação `@MapKey` é usada para especificar a chave de uma associação `java.util.Map` para a qual o tipo de chave é a chave primária ou um atributo da entidade que representa o valor do mapa.

# @MapKeyClass

A anotação @MapKeyClass é usada para especificar o tipo da chave do mapa de uma associação java.util.Map.

# @MapKeyColumn

A anotação @MapKeyColumn é usada para especificar a coluna do banco de dados, que armazena a chave de uma java.util.Mapassociation para a qual a chave do mapa é um tipo básico.

# @MapKeyEnumerated

A anotação @MapKeyEnumerated é usada para especificar que a chave da associação java.util.Map é um Java Enum.

# @MapKeyJoinColumn

A anotação @MapKeyJoinColumn é usada para especificar que a chave da associação java.util.Map é uma associação de entidades. A coluna da chave do mapa é uma FOREIGN KEY em uma tabela de links que também se junta à tabela do proprietário do Mapa com a tabela em que reside o valor do Mapa.

# @MapKeyJoinColumns

A anotação @MapKeyJoinColumns é usada para agrupar vários mapeamentos @MapKeyJoinColumn quando a chave de associação java.util.Map usa um identificador composto.

# @MapKeyTemporal

A anotação @MapKeyTemporal é usada para especificar que a chave da associação java.util.Map é um @TemporalType (por exemplo, DATE, TIME, TIMESTAMP).

# @MappedSuperclass

A anotação @MappedSuperclass é usada para especificar que os atributos de tipo anotados atualmente são herdados por qualquer entidade de subclasse.

# @MapsId

A anotação @MapsId é usada para especificar que o identificador de entidade é mapeado pelo @ManyToOne atualmente anotado ou @OneToOne associado.

## **@NamedAttributeNode**

A anotação @NamedAttributeNode é usada para especificar cada nó de atributo individual que precisa ser buscado por um Gráfico de Entidade.

## **@NamedEntityGraph**

A anotação @NamedEntityGraph é usada para especificar um Gráfico de Entidade que pode ser usado por uma consulta de entidade para substituir o plano de carregamento padrão.

## **@NamedEntityGraphs**

A anotação @NamedEntityGraphs é usada para agrupar várias anotações @NamedEntityGraph.

## **@NamedNativeQueries**

A anotação @NamedNativeQueries é usada para agrupar várias anotações @NamedNativeQuery.

## **@NamedNativeQuery**

A anotação @NamedNativeQuery é usada para especificar uma consulta SQL nativa que pode ser recuperada posteriormente por seu nome.

## **@NamedQueries**

A anotação @NamedQueries é usada para agrupar várias anotações @NamedQuery .

## **@NamedQuery**

A anotação @NamedQuery é usada para especificar uma consulta JPQL que pode ser recuperada posteriormente por seu nome.

## **@NamedStoredProcedureQueries**

A anotação @NamedStoredProcedureQueries é usada para agrupar várias anotações de @NamedStoredProcedureQuery .

## **@NamedStoredProcedureQuery**

A anotação `@NamedStoredProcedureQuery` é usada para especificar uma consulta de procedimento armazenado que pode ser recuperada posteriormente por seu nome.

## **@NamedSubgraph**

A anotação `@NamedSubgraph` usada para especificar um subgráfico em um gráfico de entidades.

## **@OneToMany**

A anotação `@OneToMany` é usada para especificar um relacionamento de banco de dados um-para-muitos.

## **@OneToOne**

A anotação `@OneToOne` é usada para especificar um relacionamento de banco de dados um-para-um.

## **@OrderBy**

A anotação `@OrderBy` é usada para especificar os atributos da entidade usados para classificação ao buscar a coleção atualmente anotada.

## **@OrderColumn**

A anotação `@OrderColumn` é usada para especificar que a ordem de coleta de anotação atual deve ser materializada no banco de dados.

## **@PersistenceContext**

A anotação `@PersistenceContext` é usada para especificar o `EntityManager` que precisa ser injetado como dependência.

## **@PersistenceContexts**

A anotação `@PersistenceContexts` é usada para agrupar várias anotações `@PersistenceContext`.

## **@PersistenceProperty**

A anotação `@PersistenceProperty` é usada pela anotação `@PersistenceContext` para declarar propriedades do provedor JPA que são passadas para o contêiner subjacente quando o `EntityManager` instância é criada.



# @PersistenceUnit

A anotação @PersistenceUnit é usada para especificar o EntityManagerFactory que precisa ser injetado como dependência.

# @PersistenceUnits

A anotação @PersistenceUnits é usada para agrupar várias anotações @PersistenceUnit.

# @PostLoad

A anotação @PostLoad é usada para especificar um método de retorno de chamada que é acionado depois que uma entidade é carregada.

# @PostPersist

A anotação @PostPersist é usada para especificar um método de retorno de chamada que é acionado após uma entidade ser persistida.

# @PostRemove

A anotação @PostRemove é usada para especificar um método de retorno de chamada que é acionado depois que uma entidade é removida.

# @PostUpdate

A anotação @PostUpdate é usada para especificar um método de retorno de chamada que é acionado depois que uma entidade é atualizada.

# @PrePersist

A anotação @PrePersist é usada para especificar um método de retorno de chamada que é acionado antes que uma entidade seja persistida.

# @PreRemove

A anotação @PreRemove é usada para especificar um método de retorno de chamada que é acionado antes que uma entidade seja removida.

# @PreUpdate

A anotação `@PreUpdate` é usada para especificar um método de retorno de chamada que é acionado antes de uma entidade ser atualizada.

## **@PrimaryKeyJoinColumn**

A anotação `@PrimaryKeyJoinColumn` é usada para especificar que a coluna de chave primária da entidade atualmente anotada também é uma chave estrangeira para alguma outra entidade (por exemplo, uma tabela de classe base em uma estratégia de herança JOINED, a tabela primária em um mapeamento de tabela secundário ou tabela pai em um `@OneToOne` relação).

## **@PrimaryKeyJoinColumns**

A anotação `@PrimaryKeyJoinColumns` é usada para agrupar várias anotações `@PrimaryKeyJoinColumn`.

## **@QueryHint**

A anotação `@QueryHint` é usada para especificar uma dica do provedor JPA usada por uma `@NamedQuery` ou por uma `@NamedNativeQuery`.

## **@SecondaryTable**

A anotação `@SecondaryTable` é usada para especificar uma tabela secundária para a entidade atualmente anotada.

## **@SecondaryTables**

A anotação `@SecondaryTables` é usada para agrupar várias anotações de `@SecondaryTable`.

## **@SequenceGenerator**

A anotação `@SequenceGenerator` é usada para especificar a sequência de banco de dados usada pelo gerador de identificadores da entidade anotada atualmente.

## **@SqlResultSetMapping**

A anotação `@SqlResultSetMapping` é usada para especificar o `ResultSet` mapeamento de uma consulta SQL nativa ou procedimento armazenado.

## **@SqlResultSetMappings**

A anotação `@SqlResultSetMappings` é usada para agrupar várias anotações `@SqlResultSetMapping` .

## **@StoredProcedureParameter**

A anotação `@StoredProcedureParameter` é usada para especificar um parâmetro de um `@NamedStoredProcedureQuery` .

## **@Table**

A anotação `@Table` é usada para especificar a tabela principal da entidade atualmente anotada.

## **@TableGenerator**

A anotação `@TableGenerator` é usada para especificar a tabela de banco de dados usada pelo gerador de identidade da entidade atualmente anotada.

## **@Temporal**

A anotação `@temporal` é usada para especificar o tipo de tempo do atributo de entidade `java.util.Date` ou `java.util.Calendar` atualmente anotado.

## **@Transient**

A anotação `@Transient` é usada para especificar que um determinado atributo de entidade não deve ser persistido.

## **@UniqueConstraint**

A anotação `@UniqueConstraint` é usada para especificar uma restrição exclusiva a ser incluída pelo gerador automático de esquemas para a tabela primária ou secundária associada à entidade anotada atualmente.

## **@Version**

A anotação `@Version` é usada para especificar o atributo de versão usado para bloqueio otimista.