

## EJB-3 für Umsteiger

### Annotationen aus dem Buch sortiert nach API

#### javax.annotation (JSR-250)

- @PostConstruct  
Callback, der nach der Instanziierung und der Dependency Injection einer Klasse aufgerufen wird. Seite 25
- @PreDestroy  
Callback, der aufgerufen wird, wenn das Objekt gelöscht werden soll. Seite 44
- @Resource  
Markiert eine allgemeine Ressource, die injiziert werden soll. Seite 157
- @RolesAllowed  
Gibt die Rollen an, die Zugriff zu einer Klasse oder Methode haben. Seite 65

#### javax.ejb (JSR-220)

- @ActivationConfigProperty  
Definiert einzelne Parameter, die bestimmen, wie ein Message-Driven Bean mit Nachrichten versorgt wird. Seite 42
- @ApplicationException  
Kennzeichnet die Exception als Application Exception. Seite 58
- @AroundInvoke  
Kennzeichnet die Methode als einen Interceptor. Seite 49
- @EJB  
In die markierte Variable soll ein Session Bean injiziert werden. Seite 155
- @EJBs  
Fasst mehrere @EJB-Annotationen zusammen. Seite 156
- @ExcludeClassInterceptors  
Listet klassenweite Interceptoren auf, die an dieser Methode nicht verwendet werden sollen. Seite 50
- @ExcludeDefaultInterceptors  
Fasst mehrere @ExcludeClassInterceptors-Annotationen zusammen. Seite 51
- @Init  
Definiert eine EJB-2.1-create-Methode für ein EJB-3.0 Stateful Session Bean. Seite 33
- @Interceptors  
Listet die auf Klassen- oder Methodenebene anzuwendenden Interceptor-Klassen. Seite Interceptors
- @Local  
Kennzeichnet die Geschäftsschnittstelle eines Session Beans als Local-sichtbar. Seite 35
- @LocalHome  
Gibt eine Klasse an, die das Local Home Interface für die Ansprache durch EJB-2.1-Komponenten bereitstellt. Seite 38
- @MessageDriven  
Definiert die Klasse als Message-Driven Bean. Seite 41

- @PostActivate  
Callback, der nach dem Aktivieren von Stateful Session Beans aufgerufen wird. Seite 32
- @PrePassivate  
Markiert eine Callback-Methode, die vor dem Passivieren eines Stateful Session Beans aufgerufen wird. Seite 32
- @Remote  
Kennzeichnet die Geschäftsschnittstelle eines Session Beans als Remote-sichtbar. Seite 35
- @RemoteHome  
Gibt eine Klasse an, die das Remote Home Interface für die Ansprache durch EJB-2.1-Komponenten bereitstellt. Seite 38
- @Remove  
Bezeichnet eine Methode, die bei Stateful Session Beans zum Löschen des Beans dient. Seite 33
- @Stateful  
Kennzeichnet die Klasse als Stateful Session Bean. Seite 31
- @Stateless  
Kennzeichnet die Klasse als Stateless Session Bean. Seite 29
- @Timeout  
Markiert eine Methode als Timeout-Callback. Seite 55
- @TransactionAttribute  
Setzt die Transaktionsflags (Required etc.) an eine Klasse oder Methode. Seite 60
- @TransactionManagement  
Gibt an, ob eine Resource Bean-Managed Transactions oder Container-Managed Transactions nutzen soll. Seite 59

#### javax.persistence (JSR-220)

- @AssociationOverride  
Überschreibt Informationen einer (in einer Superklasse definierten) Relation. Seite 96
- @AssociationOverrides  
Fasst mehrere @AssociationOverride-Annotationen zusammen. Seite 97
- @AttributeOverride  
Überschreibt Informationen für Felder z.B. für eine eingebettete Klasse. Seite 95
- @AttributeOverrides  
Fasst mehrere @AttributeOverride-Annotationen zusammen. Seite 95
- @Basic  
Gibt für eine Feld an, wann es geladen werden soll und ob in der zugehörigen Spalte NULL stehen darf. Seite 74
- @Column  
Gibt zusätzliche Daten zu einem Feld, wie einen expliziten Spaltennamen oder die maximale Länge, an. Seite 75
- @ColumnResult  
Definiert das Mapping von Ergebnisspalten bei nativen Quers. Seite 133
- @DiscriminatorColumn  
Gibt den Namen einer sog. Discriminator-Spalte an, über welche Daten unterschiedlicher Klassen, die auf eine Tabelle gemappt wurden, unterschieden werden können. Seite 87
- @DiscriminatorValue  
Gibt den Wert an, der für eine Klasse in die Discriminator-Spalte geschrieben werden soll. Seite

- @Embeddable  
Markiert die Klasse als in ein Entity einbettbar. Seite 93
- @Embedded  
Markiert das Feld als eine eingebettete Klasse. Seite 93
- @EmbeddedId  
Definiert, dass eine eingebettete Klasse als Primärschlüssel dienen soll. Seite 82
- @Entity  
Definiert die markierte Klasse als ein Entity Bean. Seite 71
- @EntityListeners  
Listet Klassen, in denen Callbacks für Entity Manager-Operationen hinterlegt sind. Seite 127
- @EntityResult  
Definiert das Mapping der Ergebnisse einer nativen Query auf Entitys. Seite 132
- @Enumerated  
Bestimmt das Mapping von Java SE 5.0 Enums auf Entity-Werte. Seite 77
- @ExcludeDefaultListeners  
Schließt globale Callbacks für ein Entity aus. Seite
- @ExcludeSuperclassListeners  
Schließt Callbacks der Superklasse für ein Entity aus. Seite 128
- @FieldResult  
Mapping von Spalten in der Select-Klausel einer nativen Query auf Felder einer Entity. Seite 133
- @GeneratedValue  
Bestimmt, dass der Primärschlüssel nicht von der Applikation bereitgestellt wird und extern erzeugt werden soll. Seite 83
- @Id  
Deklariert das Feld als Primärschlüsselfeld. Seite 81
- @IdClass  
Bestimmt, dass ein Primärschlüssel aus mehreren Feldern genutzt werden soll. Diese Felder müssen zusätzlich in einer Primärschlüsselklasse hinterlegt werden. Seite 82
- @Inheritance  
Bestimmt die Abbildungsstrategie einer Klassenhierarchie auf eine Menge von Datenbanktabellen. Seite 87
- @JoinColumn  
Definiert eine Fremdschlüsselspalte für Relationen. Seite 108
- @JoinColumns  
Fasst mehrere @JoinColumn-Annotationen zusammen. Seite 109
- @JoinTable  
N-Relationen. Seite 110
- @Lob  
Gibt an, dass es sich bei dieser Spalte um ein BLOB oder CLOB handelt. Seite 78
- @ManyToOne  
Definiert eine M:N-Relation. Seite 104
- @ManyToOne  
Definiert eine N:1-Relation. Seite 103
- @MapKey  
Bestimmt das Feld, das als Schlüssel in eine java.util.Map dient. Seite 111
- @MappedSuperclass  
Bestimmt eine Klasse, die selbst kein Entity ist, aber als Superklasse von Entitys dienen kann. Seite

- 92
- @NamedNativeQueries  
Fasst mehrere @NamedNativeQuery-Annotationen zusammen. Seite 132
- @NamedNativeQuery  
Definiert eine Named Query in SQL. Seite 132
- @NamedQueries  
Fasst mehrere @NamedQuery-Annotationen zusammen. Seite 131
- @NamedQuery  
Definiert eine Named Query in JP-QL. Seite 131
- @OneToMany  
Definiert eine 1:N-Relation. Seite 101
- @OneToOne  
Definiert eine 1:1-Relation. Seite 99
- @OrderBy  
Bestimmt, wie die N-Seite einer Relation sortiert werden soll. Seite 107
- @PersistenceContext  
Injiziert einen Persistenzkontext. Seite 116
- @PersistenceContexts  
Fasst mehrere @PersistenceContext-Annotationen zusammen. Seite 116
- @PersistenceProperty  
Ein Schlüssel-Wert-Paar, das an den Persistenz-Provider gereicht wird. Seite 116
- @PostLoad  
Callback, der nach dem Laden einer Entity aufgerufen wird. Seite 127
- @PostPersist  
Callback, der nach dem Sql-Insert einer Entity aufgerufen wird. Seite 127
- @PostRemove  
Callback, der vor nach Sql-Delete einer Entity aufgerufen wird Seite 127
- @PostUpdate  
Callback, der vor nach Sql-Update einer Entity aufgerufen wird. Seite 127
- @PrePersist  
Callback, der vor dem Sql-Insert einer Entity aufgerufen wird. Seite 127
- @PreRemove  
Callback, der vor dem SQL-Delete aufgerufen wird. Seite 127
- @PreUpdate  
Callback, der vor dem Sql-Update aufgerufen wird. Seite 127
- @PrimaryKeyJoinColumn  
Definition für Primärschlüssel bei der Abbildung einer Vererbungshierarchie. Seite 91
- @PrimaryKeyJoinColumns  
Fasst mehrere @PrimaryKeyJoinColumn-Annotationen zusammen. Seite 92
- @QueryHint  
Produktspezifische Hinweise an den Persistenz-Provider zur Ausführung einer Named Query. Seite 131
- @SecondaryTable  
Gibt eine sekundäre Tabelle für ein Entity Bean an. Seite 72
- @SecondaryTables  
Fasst mehrere @SecondaryTable-Annotationen zusammen. Seite 73
- @SequenceGenerator

- Parametriert den Primärschlüsselgenerator auf Sequenzbasis. Seite 84
- @SqlResultSetMapping  
Definiert das Mapping der Resultate von nativen Quers auf Java-Felder. Seite 132
- @SqlResultSetMappings  
Fasst mehrere @SqlResultSetMapping-Annotationen zusammen. Seite 132
- @Table  
Definiert den Tabellennamen für das Entity Bean. Seite 72
- @TableGenerator  
Parametriert den Primärschlüsselgenerator auf Tabellenbasis. Seite 85
- @Temporal  
Definiert die Spalte als einen Zeit-Wert. Seite 78
- @Transient  
Kennzeichnet das Feld oder die Property als nicht zu persistierend. Seite 73
- @UniqueConstraint  
Definition einer zusätzlichen (Eindeutigkeits-)Bedingung für eine Spalte. Seite 72
- @Version  
Kennzeichnet die Spalte als Versionspalte für das Optimistic Locking. Seite 79

#### javax.annotation.security (JSR-220)

- @DeclareRoles  
Deklariert in der Applikation verwendete Rollen. Seite 66
- @DenyAll  
Gewährt keiner Rolle Zugriff auf die Methode. Seite 65
- @PermitAll  
Gibt an, dass die bezeichnete Methode oder Klasse von allen Rollen genutzt werden kann. Seite 65
- @Resources  
Fasst mehrere @Resource-Annotationen zusammen. Seite 158
- @RunAs  
Gibt die Rolle an, in der das Bean laufen soll. Seite 65

#### javax.jws (JSR-181)

- @HandlerChain  
Definiert eine Kette von Interceptoren für einen Webservice. Seite 176
- @Oneway  
Gibt an, dass diese Webservice-Operation keinen Rückgabewert hat. Seite 170
- @SOAPBinding  
Definiert die Bindung des Webservices auf das SOAP-Protokoll. Seite 173
- @WebMethod  
Deutet an, dass die Methode als Webservice-Operation exportiert werden soll. Seite 169
- @WebParam  
Überschreibt die Standard-Mappings für die Parameter einer Webservice-Operation. Seite 170
- @WebResult  
Überschreibt das Standard-Mapping für den Rückgabewert einer Webservice-Operation. Seite 172
- @WebService  
Deklariert ein Stateless Session Bean als Webservice. Seite 168

#### javax.xml.ws (JSR-224)

- @WebServiceRef  
Injiziert einen Webservice in einen Webservice Client. Seite 174
- @WebServiceRefs  
Fasst mehrere @WebServiceRefs-Annotationen zusammen. Seite 175