



## EJB3 Annotations (Core)

### javax.annotation

[M] @PostConstruct  
[M] @PreDestroy  
[F, M, T] @Resource(authenticationType=[CONTAINER, APPLICATION], description=String, mappedName=String, name=String, shareable=boolean, type=Class)  
[T] @Resources(Resource[])

### javax.annotation.security

[T] @DeclareRoles(String[])  
[M] @DenyAll  
[M, T] @PermitAll  
[M, T] @RolesAllowed(String[])  
[T] @RunAs(String)

### javax.ejb

[] @ActivationConfigProperty(propertyName=String, propertyValue=String)  
[T] @ApplicationException(rollback=boolean)  
[F, M, T] @EJB(beanInterface=Class, beanName=String, description=String, mappedName=String, name=String)  
[T] @EJBs(EJB[])  
[M] @Init(String)  
[T] @Local(Class[])  
[T] @LocalHome(Class)  
[T] @MessageDriven(activationConfig=ActivationConfigProperty[], description=String, mappedName=String, messageListenerInterface=Class, name=String)  
[M] @PostActivate  
[M] @PrePassivate  
[T] @Remote(Class[])  
[T] @RemoteHome(Class)  
[M] @Remove(retainIfException=boolean)  
[T] @Stateful(description=String, mappedName=String, name=String)  
[T] @Stateless(description=String, mappedName=String, name=String)  
[M] @Timeout  
[M, T] @TransactionAttribute([MANDATORY, REQUIRED, REQUIRES\_NEW, SUPPORTS, NOT\_SUPPORTED, NEVER])  
[T] @TransactionManagement([CONTAINER, BEAN])

### javax.interceptor

[M] @AroundInvoke  
[M] @ExcludeClassInterceptors  
[M, T] @ExcludeDefaultInterceptors  
[M, T] @Interceptors(Class[])

## EJB3 Annotations (Persistence)

### javax.persistence

[F, M, T] @AssociationOverride(joinColumns=JoinColumn[], name=String)  
[F, M, T] @AssociationOverrides(AssociationOverride[])  
[F, M, T] @AttributeOverride(column=Column, name=String)  
[F, M, T] @AttributeOverrides(AttributeOverride[])  
[F, M] @Basic(fetch=[LAZY, EAGER], optional=boolean)  
[F, M] @Column(columnDefinition=String, insertable=boolean, length=int, name=String, nullable=boolean, precision=int, scale=int, table=String, unique=boolean, updatable=boolean)  
[] @ColumnResult(name=String)  
[T] @DiscriminatorColumn(columnDefinition=String, discriminatorType=[STRING, CHAR, INTEGER], length=int, name=String)  
[T] @DiscriminatorValue(String)  
[T] @Embeddable  
[F, M] @Embedded  
[F, M] @EmbeddedId  
[T] @Entity(name=String)  
[T] @EntityListeners(Class[])  
[] @EntityResult(discriminatorColumn=String, entityClass=Class, fields=FieldResult[])  
[F, M] @Enumerated([ORDINAL, STRING])  
[T] @ExcludeDefaultListeners  
[T] @ExcludeSuperclassListeners  
[] @FieldResult(column=String, name=String)  
[F, M] @GeneratedValue(generator=String, strategy=[TABLE, SEQUENCE, IDENTITY, AUTO])  
[F, M] @Id  
[T] @IdClass(Class)  
[T] @Inheritance(strategy=[SINGLE\_TABLE, JOINED, TABLE\_PER\_CLASS])  
[F, M] @JoinColumn(columnDefinition=String, insertable=boolean, name=String, nullable=boolean, referencedColumnName=String, table=String, unique=boolean, updatable=boolean)  
[F, M] @JoinColumns(JoinColumn[])  
[F, M] @JoinTable(catalog=String, inverseJoinColumns=JoinColumn[], joinColumns=JoinColumn[], name=String, schema=String, uniqueConstraints=UniqueConstraint[])  
[F, M] @Lob  
[F, M] @ManyToMany(cascade=[ALL, PERSIST, MERGE, REMOVE, REFRESH], fetch=[LAZY, EAGER], mappedBy=String, targetEntity=Class)  
[F, M] @ManyToOne(cascade=[ALL, PERSIST, MERGE, REMOVE, REFRESH], fetch=[LAZY, EAGER], optional=boolean, targetEntity=Class)  
[F, M] @MapKey(name=String)  
[T] @MappedSuperclass  
[T] @NamedNativeQueries(NamedNativeQuery[])  
[T] @NamedNativeQuery(hints=QueryHint[], name=String, query=String, resultClass=Class, resultSetMapping=String)  
[T] @NamedQueries(NamedQuery[])  
[T] @NamedQuery(hints=QueryHint[], name=String, query=String)  
[F, M] @OneToMany(cascade=[ALL, PERSIST, MERGE, REMOVE, REFRESH], fetch=[LAZY, EAGER], mappedBy=String, targetEntity=Class)

[F, M] @OneToOne(cascade=[ALL, PERSIST, MERGE, REMOVE, REFRESH], fetch=[LAZY, EAGER], mappedBy=String, optional=boolean, targetEntity=Class)  
[F, M] @OrderBy(String)  
[F, M, T] @PersistenceContext(name=String, properties=PersistenceProperty[], type=[TRANSACTION, EXTENDED], unitName=String)  
[T] @PersistenceContexts(PersistenceContext[])  
[] @PersistenceProperty(name=String, String)  
[F, M, T] @PersistenceUnit(name=String, unitName=String)  
[T] @PersistenceUnits(PersistenceUnit[])  
[M] @PostLoad  
[M] @PostPersist  
[M] @PostRemove  
[M] @PostUpdate  
[M] @PrePersist  
[M] @PreRemove  
[M] @PreUpdate  
[F, M, T] @PrimaryKeyJoinColumn(columnDefinition=String, name=String, referencedColumnName=String)  
[F, M, T] @PrimaryKeyJoinColumns(PrimaryKeyJoinColumn[])  
[] @QueryHint(name=String, String)  
[T] @SecondaryTable(catalog=String, name=String, pkJoinColumns=PrimaryKeyJoinColumn[], schema=String, uniqueConstraints=UniqueConstraint[])  
[T] @SecondaryTables(SecondaryTable[])  
[F, M, T] @SequenceGenerator(allocationSize=int, initialValue=int, name=String, sequenceName=String)  
[T] @SqlResultSetMapping(columns=ColumnResult[], entities=EntityResult[], name=String)  
[T] @SqlResultSetMappings(SqlResultSetMapping[])  
[T] @Table(catalog=String, name=String, schema=String, uniqueConstraints=UniqueConstraint[])  
[F, M, T] @TableGenerator(allocationSize=int, catalog=String, initialValue=int, name=String, pkColumnName=String, pkColumnValue=String, schema=String, table=String, uniqueConstraints=UniqueConstraint[], valueColumnName=String)  
[F, M] @Temporal([DATE, TIME, TIMESTAMP])  
[F, M] @Transient  
[] @UniqueConstraint(columnNames=String[])  
[F, M] @Version

*F: Field*

*M: Method*

*T: Type (class)*

*Pkg: Package*

*Par: Parameter*

Provided by EasyBeans (<http://www.easybeans.net>)  
Produced automatically by analyzing classes.



## Other Java EE 5 annotations

### **javax.jws**

[F, M, T] @HandlerChain(file=String, name=String)  
[M] @Oneway  
[M] @WebMethod(action=String, exclude=boolean, operationName=String)  
[Par] @WebParam(header=boolean, mode=[IN, OUT, INOUT], name=String, partName=String, targetNamespace=String)  
[M] @WebResult(header=boolean, name=String, partName=String, targetNamespace=String)  
[T] @WebService(endpointInterface=String, name=String, portName=String, serviceName=String, targetNamespace=String, wsdlLocation=String)

### **javax.jws.soap**

[] @InitParam(name=String, String)  
[M, T] @SOAPBinding(parameterStyle=[BARE, WRAPPED], style=[DOCUMENT, RPC], use=[LITERAL, ENCODED])  
[] @SOAPMessageHandler(className=String, headers=String[], initParams=InitParam[], name=String, roles=String[])  
[T] @SOAPMessageHandlers(SOAPMessageHandler[])

### **javax.xml.bind.annotation**

[Pkg, T] @XmlAccessorOrder([UNDEFINED, ALPHABETICAL])  
[Pkg, T] @XmlAccessorType([PROPERTY, FIELD, PUBLIC\_MEMBER, NONE])  
[F, M] @XmlAnyAttribute  
[F, M] @XmlAnyElement(lax=boolean, Class)  
[F, M, Par] @XmlAttachmentRef  
[F, M] @XmlAttribute(name=String, namespace=String, required=boolean)  
[F, M] @XmlElement(defaultValue=String, name=String, namespace=String, nillable=boolean, required=boolean, type=Class)  
[M] @XmlElementDecl(defaultValue=String, name=String, namespace=String, scope=Class, substitutionHeadName=String, substitutionHeadNamespace=String)  
[F, M] @XmlElementRef(name=String, namespace=String, type=Class)  
[F, M] @XmlElementRefs(XmlElementRef[])  
[F, M] @XmlElementWrapper(name=String, namespace=String, nillable=boolean)  
[F, M] @XmlElements(XmlElement[])  
[T] @XmlEnum(Class)  
[F] @XmlEnumValue(String)  
[F, M] @XmlID

[F, M] @XmlIDREF  
[F, M, T] @XmlInlineBinaryData  
[F, M, Par] @XmlList  
[F, M, Par] @XmlMimeType(String)  
[F, M] @XmlMixed  
[] @XmlNs(namespaceURI=String, prefix=String)  
[T] @XmlRegistry  
[T] @XmlRootElement(name=String, namespace=String)  
[Pkg] @XmlSchema(attributeFormDefault=[UNQUALIFIED, QUALIFIED, UNSET], elementFormDefault=[UNQUALIFIED, QUALIFIED, UNSET], namespace=String, xmlns=XmlNs[])  
[F, M, Pkg] @XmlSchemaType(name=String, namespace=String, type=Class)  
[Pkg] @XmlSchemaTypes(XmlSchemaType[])  
[F, M] @XmlTransient  
[T] @XmlType(factoryClass=Class, factoryMethod=String, name=String, namespace=String, propOrder=String[])  
[F, M] @XmlValue

### **javax.xml.bind.annotation.adapters**

[F, M, Pkg, Par, T] @XmlJavaTypeAdapter(type=Class, Class)  
[Pkg] @XmlJavaTypeAdapters(XmlJavaTypeAdapter[])

### **javax.xml.ws**

[T] @BindingType(String)  
[M] @RequestWrapper(className=String, localName=String, targetNamespace=String)  
[M] @ResponseWrapper(className=String, localName=String, targetNamespace=String)  
[T] @ServiceMode([MESSAGE, PAYLOAD])  
[M] @WebEndpoint(name=String)  
[T] @WebFault(faultBean=String, name=String, targetNamespace=String)  
[T] @WebServiceClient(name=String, targetNamespace=String, wsdlLocation=String)  
[T] @WebServiceProvider(portName=String, serviceName=String, targetNamespace=String, wsdlLocation=String)  
[F, M, T] @WebServiceRef(mappedName=String, name=String, type=Class, Class, wsdlLocation=String)  
[T] @WebServiceRefs(WebServiceRef[])

*F: Field*  
*M: Method*  
*T: Type (class)*  
*Pkg: Package*  
*Par: Parameter*