

## Bean Types

T	<b>@Stateless</b> (name=“”)
T	<b>@Stateful</b> (name=“”)
M	<b>@Init</b> (init = “”)
M	<b>@Remove</b> (retainIfException=false)
T	<b>@MessageDriven</b> (name=“”, ActivationConfigProperty[] activationConfig={})
	<b>@ActivationConfigProperty</b> (propertyName=“”, propertyValue=“”)
T	<b>@Entity</b> (name=“”, access=[PROPERTY, FIELD], entityType=[CMP, BMP], version=3)
T	<b>@Remote</b>
T	<b>@Local</b>
M	<b>@BusinessMethod</b>

## Transaction and Security

T	<b>@TransactionManagement</b> ( value=[CONTAINER, BEAN])
TM	<b>@TransactionAttribute</b> (value=[MANDATORY, REQUIRED, REQUIRES_NEW, SUPPORTS, NOT_SUPPORTED, NEVER])
TM	<b>@MethodPermissions</b> (String[] roles)
TM	<b>@PermitAll</b>
M	<b>@DenyAll</b>
T	<b>@RunAs</b> (String rolaname)
T	<b>@DeclareRoles</b> (value={})
TM	<b>@RolesAllowed</b> (value={})

## Callbacks

T	<b>@CallbackListener</b> (String classname)
M	<b>@PostConstruct</b>
M	<b>@PreDestroy</b>
M	<b>@PostActivate</b>
M	<b>@PrePassivate</b>
Te	<b>@EntityListener</b> (String classname)
Me	<b>@PrePersist</b>
Me	<b>@PostPersist</b>
Me	<b>@PreRemove</b>
Me	<b>@PostRemove</b>
Me	<b>@PreUpdate</b>
Me	<b>@PostLoad</b>

## Resources

TMF	<b>@Resource</b> (name=“”, type=Object.class, AuthenticationType=[CONTAINER, Application], shareable=true, resourceType=“”, mappedName=“”)
T	<b>@Resources</b> (Resource[] value)
TMF	<b>@EJB</b> (name=“”, businessInterface=Object.class, mappedName=“”, description=“”)
T	<b>@EJBs</b> {EJB[] value}

## Interceptors, Exceptions

T	<b>@Interceptors</b> (Class[] value)
M	<b>@AroundInvoke</b>
TM	<b>@ExcludeDefaultInterceptors</b>
M	<b>@ExcludeClassInterceptors</b>
T	<b>@ApplicationException</b> (rollback=false)

## EJB 2.1 and Ealier Client View

Ts	<b>@RemoteHome</b> (Class value)
Ts	<b>@LocalHome</b> (Class value)

## Queries

TM	<b>@FlushMode</b> (value=[COMMIT, AUTO, NEVER])
PTM	<b>@NamedQuery</b> (name=“”, queryString, resultType, ejbInterfaceType[LOCAL, REMOTE])
PTM	<b>@NamedQueries</b> (NamedQuery[] value)

## ResultSetMapping

TM	<b>@SqlResultSetMapping</b> (String name, EntityResult[] entities={}, ColumnResult[] columns={})
	<b>@EntityResult</b> (String name, FieldResult[] fields = {})
	<b>@FieldResult</b> (String name, column)
	<b>@ColumnResult</b> (String name)

## Object-Relational-Mapping

T	<b>@Table</b> (name=“”, catalog=“”, schema=“”, UniqueConstraint[]={}, specified=true)
T	<b>@SecondaryTable</b> (name=“”, catalog=“”, schema=“”, JoinColumn[] join={}, UniqueConstraint[] uniqueConstraints={})
T	<b>@SecondaryTables</b> (SecondaryTable[] value)
	<b>@UniqueConstraint</b> (primary=false, String[] columnNames)
TMF	<b>@Column</b> (name=“”, primaryKey=false, unique=false, nullable=true, insertable=true, updatable=true, columnDefinition=“”, secondaryTable=“”, length=255, precision=0, scale=0, specified=true)
TMF	<b>@JoinColumn</b> (name=“”, referencedColumnName=“”, primaryKey=“”, unique=false, nullable=true, insertable=true, updatable=true, columnDefinition=“”, secondaryTable=“”)
TMF	<b>@JoinColumns</b> (JoinColumn[])
MF	<b>@GeneratedValue</b> (generationType=[NONE, TABLE, SEQUENCE, IDENTITY, AUTO), generator=“”)
MF	<b>@Id</b>
MF	<b>@EmbeddedId</b> (AttributeOverride[]={})
	<b>@AttributeOverride</b> (name, Column[]={})
T	<b>@IdClass</b>
MF	<b>@Transient</b>
MF	<b>@Version</b>
MF	<b>@Lob</b> (fetch=[LAZY, EAGER], type=[BLOB, CLOB])
MF	<b>@Serialized</b> (fetch=[LAZY, EAGER])

## Embedded Values

T	<b>@Embeddable</b>
MF	<b>@Embedded</b> (AttributeOverride[] override={})

## Relations

MF	<b>@ManyToOne</b> (targetEntity=void.class, cascade={ALL, MERGE, PERSIST, REFRESH, REMOVE}, fetch=[LAZY, EAGER], optional=true)
MF	<b>@OneToOne</b> (targetEntity=void.class, cascade={...}, fetch=[...], optional=true, mappedBy=“”, usePKasFK=false)
MF	<b>@OneToMany</b> (targetEntity=void.class, cascade={...}, fetch=[...], mappedBy=“”)
MF	<b>@AssociationTable</b> (table=@Table(specified=false), joinColumns={}, inverseJoinColumns={})
MF	<b>@ManyToMany</b> (targetEntity=void.class, cascade={...}, fetch=[...], mappedBy=“”)

## Inheritance

T	<b>@Inheritance</b> (strategy=[SINGLE_TABLE, TABLE_PER_CLASS, JOINED], AttributeOverride[] inheritedAttributes={})
T	<b>@DiscriminatorColumn</b> (name=“DTYPE”, discriminator Type=[STRING, CHAR, INTEGER], columnDefinition=“”, length=31)
T	<b>@DiscriminatorValue</b> (String value)
T	<b>@InheritanceJoinColumn</b> (name=“”, referencedColumnName=“”, columnDefinition=“”)
T	<b>@InheritanceJoinColumn</b> ( InheritanceJoinColumn[] columns)
T	<b>@DiscriminatorColumn</b> (name=“”, nullable=false, columnDefinition=“”, length=10)
T	<b>@MappedSuperclass</b>

## Generators

MF	<b>@GeneratedValue</b> (name=“”, strategy=[AUTO, IDENTITY, SEQUENCE, TABLE], pkColumnName=“”, valueColumnName=“”)
PTMF	<b>@SequenceGenerator</b> (String name, sequenceName=“”, initialValue=0, allocationSize=50)
PTMF	<b>@TableGenerator</b> (String name, table=“”, catalog=“”, schema=“”, pkColumnName=“”, valueColumnName=“”, pkColumnValue=“”, initialValue=0 allocationSize=50, uniqueConstraints={})

## Legend

PMTF	Annotation applies to Package, Method, Type, Field
Me	Annotation applies to Entity Bean Methods
Ts	Annotation applies to Session Bean
VALUE	Default Value

## EJB 3.0 Annotations Cheat Sheet

© 2005, 2006 Philipp Meier – [www.fnogol.de](http://www.fnogol.de)  
Version 1.2.1, based on EJB3.0 PFD  
Report errors to [meier@fnogol.de](mailto:meier@fnogol.de)

*This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 2.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/2.0/de/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.*