

Java EE 6 Annotations

Alternatives for EJB management

CDI	JSF	EJB3
-----	-----	------

CDI: javax.inject

CMF **@Inject**
@Named(value="")

CDI: javax.enterprise.inject

TMF **@New**(value=className)
 TMF **@Alternative**
 TMF **@Any**
 MF **@Produces**

CDI: javax.enterprise.context

TMF **@ApplicationScoped**
 TMF **@ConversationScoped**
 TMF **@RequestScoped**
 TMF **@SessionScoped**
 TMF **@Dependent**

JSF management: javax.faces.bean

T **@ManagedBean**(name="", eager=false)
 F **@ManagedProperty**(name="", value="")
 T **@ApplicationScoped**
 T **@CustomScoped**(value="")
 T **@NoneScoped**
 T **@ReferencedBean**(name="")
 T **@RequestScoped**
 T **@SessionScoped**
 T **@ViewScoped**

EJB3 injection: javax.ejb

TMF **@EJB**(name="", beanInterface=Object.class, mappedName="", lookup="" | beanName="", description="")
 T **@EJBs**(EJB[])

EJB3 injection: javax.annotation

T **@ManagedBean**(value="")
 TMF **@Resource**(name="", type=Object.class, AuthenticationType=[CONTAINER, Application], shareable=true, lookup="", mappedName="")
 T **@Resources**(Resource[])

Bean Types: javax.ejb

T **@Stateless**(name=className)
 T **@Stateful**(name=className)
 T **@Singleton**(name=className)
 T **@MessageDriven**(name="", activationConfig=ActivationConfigProperty[])
@ActivationConfigProperty(propertyName="", propertyValue="")
 T **@Local**(value=Class[])
 T **@Remote**(value=Class[])
 T **@LocalBean**
 TM **@Asynchronous**
 TM **@Lock**(value=[WRITE, READ])
 T **@ConcurrencyManagement**(value=[CONTAINER, BEAN])
 T **@DependsOn**(value=String[])
 T **@Startup**

Timeouts: javax.ejb

TM **@AccessTimeout**(value="0", unit=MILLISECONDS)
 T **@StatefulTimeout**(value="0", unit=MILLISECONDS)
 M **@Timeout**
 M **@Schedule**(year="*", month="*", bimonthly="*", dayOfWeek="*", hour="0", minute="0", info="", persistent=true, timezone="")
 M **@Schedules**(Schedule[])

Transaction: javax.ejb

T **@TransactionManagement**(value=[CONTAINER, BEAN])
 TM **@TransactionAttribute**(value=[MANDATORY, REQUIRED, REQUIRES_NEW, SUPPORTS, NOT_SUPPORTED, NEVER])

EJB Callbacks: javax.ejb and .annotation

M **@AfterBegin**
 M **@AfterCompletion**
 M **@BeforeCompletion**
 M **@PostConstruct**
 M **@PreDestroy**
 M **@PostActivate**
 M **@PrePassivate**
 M **@Remove**(retainIfException=false)

Interceptors: javax.interceptor

T **@Interceptor**

TM **@Interceptors**(Class[])
 M **@AroundInvoke**
 M **@AroundTimeout**
 TM **@ExcludeDefaultInterceptors**
 M **@ExcludeClassInterceptors**

Security: javax.annotation.security

T **@RunAs**(String rolename)
 T **@DeclareRoles**(value=String[])
 TM **@RolesAllowed**(value=String[])
 TM **@PermitAll**
 TM **@DenyAll**

```
src/java/ [potentially packaged as a .jar inside a .ear]
|-- META-INF/
|   |-- beans.xml          [or in WEB-INF/]
|   |-- persistence.xml
|   |-- com/
|       |-- myBusiness/
|           |-- entities/
|               |-- Entities.java
|           |-- resources/
|               |-- validationMessages.properties
|               |-- messages.properties
|           |-- EJBs.java
web/ [potentially packaged as a .war inside a .ear]
|-- WEB-INF/
|   |-- beans.xml          [or in META-INF/]
|   |-- faces-config.xml
|   |-- web.xml
|   |-- resources/
|       |-- css/
|           |-- standard.css
|       |-- javascript/
|           |-- standard.js
|       |-- jsfpages.xhtml
```

Legend

TCMF Annotation applies to Type, Constructor, Method, Field
 VALUE Default Value

Java EE 6 Annotations Cheat Sheet

Version 1.3, based on Java EE 6 API Doc

©2005,2011 Philipp Meier – www.fnogol.de

2011 Updated for EJB 3.1, JSF 2.0, JPA 2.0 by Chris Rennie
 Report errors to c.rennie@physics.usyd.edu.au

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.

Persistence provider: javax.persistence

- TMF **@PersistenceContext**(name="", unitName="", type=[TRANSACTION,EXTENDED], properties=PersistenceProperty[])
- T **@PersistenceContexts**(PersistenceContext[])
- TMF **@PersistenceUnit**(name="", unitName="")
- T **@PersistenceUnits**(PersistenceUnit[])
- @PersistenceProperty**(name="", value="")
- T **@Cacheable**(value=true)

Object-relational mapping: javax.persistence

- T **@Entity**(name="")
- T **@Table**(name="", schema="", catalog="", UniqueConstraints=UniqueConstraint[])
- T **@SecondaryTable**(name="", schema="", catalog="", pkJoinColumns=PrimaryKeyJoinColumn[], uniqueConstraints=UniqueConstraint[])
- T **@SecondaryTables**(SecondaryTable[])
- @UniqueConstraint**(columnNames=String[], name="")
- T **@IdClass**(Class)
- TMF **@Access**(value=[FIELD,PROPERTY])
- TMF **@PrimaryKeyJoinColumn**(name="", columnDefinition="", referencedColumnName="")
- TMF **@PrimaryKeyJoinColumns**(PrimaryKeyJoinColumn[])
- TMF **@AttributeOverride**(name=String, column=Column)
- TMF **@AssociationOverride**(name=String, joinTable=JoinTable, JoinColumns=JoinColumn[])
- MF **@Column**(name="", table="", unique=false, nullable=true, insertable=true, updatable=true, columnDefinition="", length=255, precision=0, scale=0)
- MF **@JoinColumn**(name="", table="", referencedColumnName="", unique=false, nullable=true, insertable=true, updatable=true, columnDefinition="")
- MF **@JoinColumns**(JoinColumn[])
- MF **@JoinTable**(name="", schema="", catalog="", joinColumns={}, inverseJoinColumns={}, uniqueConstraints=UniqueConstraint[])
- MF **@CollectionTable**(name="", schema="", catalog="", joinColumns={}, uniqueConstraints={})
- MF **@ElementCollection**(targetClass=void.class, fetch=[LAZY,EAGER])
- MF **@Id**
- MF **@Basic**(fetch=[LAZY, EAGER], optional=true)
- MF **@Enumerated**(EnumType)
- MF **@Temporal**([DATE, TIME, TIMESTAMP])
- MF **@Transient**

- MF **@Lob**
- MF **@Version**

Embedded values: javax.persistence

- T **@Embeddable**
- MF **@EmbeddedId**
- MF **@Embedded**
- MF **@MapsId**(value="")

Relations: javax.persistence

- MF **@OneToOne**(targetEntity=void.class, cascade={ALL,DETACH,MERGE,PERSIST,REFRESH,REMOVE}, fetch=[LAZY, EAGER], optional=true, mappedBy="", orphanRemoval=false)
- MF **@OneToMany**(targetEntity=void.class, cascade={...}, fetch=[...], mappedBy="", orphanRemoval=false)
- MF **@ManyToOne**(targetEntity=void.class, cascade={...}, fetch=[...], optional=true)
- MF **@ManyToMany**(targetEntity=void.class, cascade={...}, fetch=[...], mappedBy="")
- MF **@OrderBy**(value="")
- MF **@OrderColumn**(name="", columnDefinition="", insertable=true, updateable=true, nullable=true)

Queries: javax.persistence

- T **@NamedQuery**(name=String, query=String)
- T **@NamedQueries**(NamedQuery[])
- T **@NamedNativeQuery**(name=String, query=String, resultClass=void.class)
- T **@NamedNativeQueries**(NamedQuery[])
- @QueryHint**(name="", value="")

ResultSet mapping: javax.persistence

- T **@SqlResultSetMapping**(name=String, entities=EntityResult[], columns=ColumnResult[])
- T **@SqlResultSetMappings**(SqlResultSetMapping[])
- @EntityResult**(entityResult=Class, discriminatorColumn="", fields=FieldResult[])
- @FieldResult**(name="", column="")
- @ColumnResult**(name="")

Inheritance: javax.persistence

- T **@Inheritance**(strategy=[SINGLE_TABLE, TABLE_PER_CLASS, JOINED])
- T **@DiscriminatorColumn**(name="DTYPE", discriminatorType=[STRING, CHAR, INTEGER], columnDefinition="", length=31)
- T **@DiscriminatorValue**(value="")

- T **@MappedSuperclass**

Generators: javax.persistence

- MF **@GeneratedValue**(generator="", strategy=[AUTO,IDENTITY,SEQUENCE,TABLE])
- TMF **@SequenceGenerator**(name="", schema="", catalog="", sequenceName="", initialValue=0, allocationSize=50)
- TMF **@TableGenerator**(name="", table="", schema="", catalog="", pkColumnName="", valueColumnName="", pkColumnName="", initialValue=0, allocationSize=50, uniqueConstraints={})

MapKey: javax.persistence

- MF **@MapKey**(value="")
- MF **@MapKeyClass**(value=Class)
- MF **@MapKeyColumn**(name="", table="", unique=false, nullable=true, insertable=true, updatable=true, columnDefinition="", length=255, precision=0, scale=0)
- MF **@MapKeyEnumerated**(value=[ORDINAL,STRING])
- MF **@MapKeyJoinColumn**(name="", table="", referencedColumnName="", unique=false, nullable=true, insertable=true, updatable=true, columnDefinition="")
- MF **@MapKeyJoinColumns**(MapKeyJoinColumn[])
- MF **@MapKeyTemporal**([DATE,TIME,TIMESTAMP])

JPA Callbacks: javax.persistence

- M **@PostLoad**
- M **@PrePersist**
- M **@PostPersist**
- M **@PreRemove**
- M **@PostRemove**
- M **@PreUpdate**
- M **@PostUpdate**

Listeners: javax.persistence

- T **@EntityListeners**(Class[])
- T **@ExcludeDefaultListeners**
- T **@ExcludeSuperclassListeners**

See also:

- JSF <http://horstmann.com/corejsf/refcard.html>
- JPA <http://www.oracle.com/technetwork/middleware/ias/top-link-jpa-annotations-096251.html>
- CDI <http://www.theserverside.com/news/1373391/Dependency-Injection-in-Java-EE-6-Part-1>