Java EE 6 Annotations

Alternatives for management

CDI	JSF	EJB

CDI: javax.inject

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

CDI: javax.enterprise.context

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

CDI: javax.enterprise.inject

TMF	**@New**(value=Class.class)
TMF	**@Alternative**
TMF	**@Any**
MF	**@Produces, @Disposes**

JSF management: javax.faces.bean

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

EJB injection: javax.ejb

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

Resource injection: javax.annotation

TMF	**@Resource**(name="", type=Class.class, authenticationType= [AuthenticationType.*CONTAINER*, APPLICATION], shareable=true, lookup="", mappedName="")
T	**@Resources**(@Resource[])

EJB Types: javax.ejb

Session beans

T	**@Stateless**(name="*ClassName*")
T	**@Stateful**(name="*ClassName*")
T	**@Singleton**(name="*ClassName*")
T	**@Local**(Class.class[])
T	**@Remote**(Class.class[])
T	**@LocalBean**
TM	**@Asynchronous**
TM	**@Lock**([LockType.*WRITE*,READ])
T	**@ConcurrencyManagement**([*CONTAINER*, BEAN])
T	**@DependsOn**(String[])
T	**@Startup**

Non-session beans

T	**@MessageDriven**(name="*ClassName*", activationConfig=@ActivationConfigProperty[])
	@ActivationConfigProperty(propertyName="", propertyValue="")
T	**@ManagedBean**(value="") [in javax.annotation.*]

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**([*CONTAINER*, BEAN])
TM	**@TransactionAttribute**([TransactionAttributeType. MANDATORY, *REQUIRED*, REQUIRES_NEW, SUPPORTS, NOT_SUPPORTED, NEVER])
M	**@AfterBegin**
M	**@BeforeCompletion**
M	**@AfterCompletion**

Lifecycle: javax.ejb

M	**@PostConstruct** [in javax.annotation.*]
M	**@PreDestroy** [in javax.annotation.*]
M	**@PostActivate**
M	**@PrePassivate**
M	**@Remove**(retainIfException=false)

Interceptors: javax.interceptor

TM	**@Interceptors**(Class.class[])
TM	**@ExcludeDefaultInterceptors**
M	**@ExcludeClassInterceptors**
M	**@AroundInvoke**
M	**@AroundTimeout**
T	**@Interceptor** [only required with @InterceptorBinding]

Security: javax.annotation.security

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

```
       -- Possible source file layout for web app –
1  lib/           [potentially copied to /lib/ inside an EAR]
   `-- extra.jar      [jar shared between all modules]
2  src/java/      [potentially packaged as EJB-JAR inside EAR -
   |-- ValidationMessages.properties           or under
   |-- JSFStrings.properties            WEB-INF/classes/
   |-- META-INF/                          inside WAR]
   |   |-- persistence.xml    [for JPA config]
   |   `-- ejb-jar.xml        [for deployment descriptors]
   `-- com/
       `-- myBusiness/
           |-- entities/
           |   `-- Entities.java
           `-- EJBs.java
   web/        [potentially packaged as a WAR inside an EAR]
3  |-- WEB-INF/
   |   |-- beans.xml        [for CDI config]
   |   |-- faces-config.xml  [for JSF config]
   |   `-- web.xml          [for Servlet 2.5 config]
   |-- resources/
   |   |-- css/
   |   |   `-- standard.css
   |   `-- javascript/
   |       `-- standard.js
   `-- jsfpages.xhtml
```
EAR class loader levels

Legend

TCMF	Annotation for Type, Constructor, Method, Field
VALUE	Default Value

Persistence provider: javax.persistence

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

Object 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.class)
TMF	**@Access**(value=[AccessType.FIELD, PROPERTY])
TMF	**@AttributeOverride**(name="", column=@Column)

Prop/Field mapping: javax.persistence

MF	**@Id**	
MF	**@Column**(name="", table="", unique=false, nullable=true, insertable=true, updatable=true, columnDefinition="", length=255, precision=0, scale=0)	
MF	**@Basic**(fetch=[LAZY, *EAGER*], optional=true)	
MF	**@Enumerated**([EnumType.*ORDINAL*,STRING])	
MF	**@Temporal**([TemporalType.DATE, TIME, TIMESTAMP])	
MF	**@Transient**	**!!** Near-redundancy:
MF	**@Lob**	'nullable' describes DB fields
MF	**@Version**	'optional' describes entity properties

Relationship mapping: javax.persistence

MF	**@OneToOne**(targetEntity=*void.class*, cascade={CascadeType.ALL, DETACH, MERGE, PERSIST, REFRESH, REMOVE}, fetch=[FetchType.LAZY, *EAGER*], optional=true, mappedBy="propOrField", orphanRemoval=false)
MF	**@OneToMany**(targetEntity=*void.class* , cascade={...}, fetch=[*LAZY*,EAGER], mappedBy="", orphanRemoval=false)
MF	**@ManyToOne**(targetEntity=*void.class*, cascade={...}, fetch=[LAZY,*EAGER*], optional=true)
MF	**@ManyToMany**(targetEntity=*void.class*, cascade={...}, fetch=[*LAZY*,EAGER], mappedBy="")

TMF	**@PrimaryKeyJoinColumn**(name="THIS_PK", columnDefinition="", referencedColumnName= "OTHER_FK")
TMF	**@PrimaryKeyJoinColumns**(@PrimaryKeyJoinColumn[])
MF	**@JoinColumn**(name="THIS_FK", table="", unique=false, nullable=true, insertable=true, updatable=true, columnDefinition="", referencedColumnName="OTHER_PK")
MF	**@JoinColumns**(@JoinColumn[])
MF	**@JoinTable**(name="", schema="", catalog="", joinColumns={}, inverseJoinColumns={}, uniqueConstraints=@UniqueConstraint[])
TMF	**@AssociationOverride**(name="", joinTable= @JoinTable || joinColumns=@JoinColumn[])
MF	**@MapsId**(value="propOrField")
MF	**@OrderBy**(value="")
MF	**@OrderColumn**(name="FIELD_ORDER",nullable=true, insertable=true, updatable=true, columnDefinition="")

Queries: javax.persistence

T	**@NamedQuery**(name="", query="JPQL...", hints= @QueryHint[], lockMode=...)
T	**@NamedQueries**(@NamedQuery[])
T	**@NamedNativeQuery**(name="", query="SQL...", resultClass=void.class, hints=..., lockMode=...)
T	**@NamedNativeQueries**(@NamedQuery[])
	@QueryHint(name="", value="")

SqlResultSet mapping: javax.persistence

T	**@SqlResultSetMappings**(@SqlResultSetMapping[])
T	**@SqlResultSetMapping**(name="", entities= @EntityResult[], columns=@ColumnResult[])
	@EntityResult(entityResult=Class.class, disriminatorColumn="", fields=@FieldResult[])
	@FieldResult(name="", column="")
	@ColumnResult(name="")

Inheritance: javax.persistence

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

Generators: javax.persistence

MF	**@GeneratedValue**(generator="", strategy= [GenerationType.*AUTO*,IDENTITY,SEQUENCE,TABLE])

TMF	**@SequenceGenerator**(name="", schema="", catalog="", sequenceName="", initialValue=0, allocationSize=50)
TMF	**@TableGenerator**(name="", table="", schema="", catalog="", pkColumnName="", valueColumnName="", pkColumnValue="", initialValue=0, allocationSize=50, uniqueConstraints=@UniqueConstraint[])

Embedded objects: javax.persistence

T	**@Embeddable**
MF	**@EmbeddedId**
MF	**@Embedded**

Collections and Maps: javax.persistence

MF	**@CollectionTable**(name="", schema="", catalog="", joinColumns={}, uniqueConstraints={})
MF	**@ElementCollection**(targetClass=*void.class*, fetch=[FetchType.*LAZY*, EAGER])
MF	**@MapKeyColumn**(name="FK", table="", unique=false, nullable=true, insertable=true, updatable=true, columnDefinition="", length=255, precision=0, scale=0)
MF	**@MapKeyJoinColumn**(name="FK", table="", unique=false, nullable=true, insertable=true, updatable=true, columnDefinition="", referencedColumnName="OTHER_PK")
MF	**@MapKeyJoinColumns**(@MapKeyJoinColumn[])
MF	**@MapKey**("propOrField")
MF	**@MapKeyClass**(Class.class)
MF	**@MapKeyEnumerated**([EnumType.*ORDINAL*,STRNG])
MF	**@MapKeyTemporal**([TemporalType.DATE, TIME, TIMESTAMP])

JPA Lifecycle: javax.persistence

M	**@PrePersist**	[C]
M	**@PostPersist**	[C]
M	**@PostLoad**	[R]
M	**@PreUpdate**	[U]
M	**@PostUpdate**	[U]
M	**@PreRemove**	[D]
M	**@PostRemove**	[D]

Listeners: javax.persistence

T	**@EntityListeners**(Class.class[])
T	**@ExcludeDefaultListeners**
T	**@ExcludeSuperclassListeners**
See also	http://www.oracle.com/technetwork/middleware/ias/top link-jpa-annotations-096251.html