



Java Enterprise Edition

Vad är JEE?



Java kod för Internetbaserade tjänster. Körs utan en main metod och skall helst inte stoppas utan bara skicka ut de svaren som begärs av programmet

WAR – Web Application Resource / Web Application aRchive

JAR – Java ARchive

EAR – Enterprise Application aRchive

Körs på en applikationsserver

GlassFish

Eclipse GlassFish

IBM WebSphere

JBoss

Wildfly

JEE Beståndsdelar



- ▶ EJB
 - ▶ Servlet
 - ▶ CDI
 - ▶ JavaMail
 - ▶ JPA
 - ▶ JMS
 - ▶ JCA
 - ▶ JTA
 - ▶ JASPIC
 - ▶ JACC
 - ▶ JAXB
 - ▶ JAX-RS
 - ▶ JAX-WS
 - ▶ EL
 - ▶ JSF
 - ▶ JSP
- ▶ EJB
 - ▶ CDI
 - ▶ JAX-RS
 - ▶ JPA
 - ▶ JMS



CDI

Contexts and Dependency Injection

Är en JEE feature som hjälper till att knyta ihop de olika lagrena som finns i en applikation. Hjälper utvecklare att kunna använda mer flexibilitet av integration av komponenter som är mycket löst sammansatta.

Vi kommer titta närmare på `@Inject` annoteringen, vilket gör att vi kan injicera en böna in i en annan.

Andra annotationer är:

`@Producer`, som skapar ett objekt som kan bli injicerat.

`@ConversationScoped`, böna som beskriver en användares interaktion i en JavaServer Faces applikation.



EJB

Enterprise Java Beans

Det är en Server-Side komponent som kapslar in affärslogiken i en applikation.

De har ansvar för transaktioner, integrationen med JPA.

Kontrollerar Events med JMS och JEE Connector Architecture

Namn och bibliotek service (JNDI)

Typ av bönor

Singleton

Stateful

Stateless



JPA

Java Persistent API (Application Program Interface)

- Persisterar (sparar) objekt till en databas.
- Är en ORM (Object Relational Mapping).
- Är inte ett verktyg eller framework.
Beskriver en uppsättning av begrepp.

Verktyg

- Hibernate
- Spring
- SpringBoot

JPA var ett subset till EJB 3.0, men idag är det sin egna specifikation. Idag är JPA implementerat i en EJB container, som innehåller ett Persistence lager, som är definerat av JPA specifikationen



JAX-RS

Java API for RESTful Web Services

Ett Java Programming Language som hjälper utvecklare att skapa web services enligt Representational State Transfer (REST) arkitektur. Det använder annotationer för att enklare utveckla klienter och end points.

- @Path
 - @GET, @PUT, @POST, @DELETE, @HEAD
 - @Produces
 - @Consumes
- @PathParam
 - @QueryParam
 - @HeaderParam
 - @DefaultValue
 - m.fl

JMS

Java Message Service



Kräver en provider.

Alla JEE Servers *måste* ha en provider inbyggd.

Wildfly startas med standalone-full.xml istället för standalone.xml

JMS stöder två modeller:

- ☐ point-to-point
- ☐ publish-and-subscribe