

Java EE

- Java Enterprise Edition
 - J2EE, JEE, Jakarta
- Web server vs Application Server
 - Tomcat
 - JBoss EAP / WildFly
- Sviluppo con Eclipse / Maven
 - Dynamic web application

Java Enterprise Edition (Java EE)

- Prima di JDK 5 nota come J2EE
- In transizione da Oracle verso Eclipse Foundation
 - Jakarta EE <https://jakarta.ee/>
- Estende la Standard Edition con specifiche per
 - sviluppo enterprise, distributed computing, web services, ...
- Applicazioni Enterprise eseguite da un reference runtime
 - Gestisce il ciclo di vita delle componenti, multithreading, sicurezza, ...
- Dal client (web browser o altro): request all'applicazione → response

Apache Tomcat

- **Web Server** (o Container) → implementa parzialmente le specifiche Java EE
 - <https://tomcat.apache.org/>
- La versione 9 richiede Java SE 8+ e supporta Java EE 8
 - **Java Servlet** 4.0
 - **Java Server Pages** 2.3
 - **Java Expression Language** 3.0
 - Java Web Socket 1.1
 - Download <https://tomcat.apache.org/download-90.cgi> (zip file)
- *(La versione 10 è basata su Jakarta → Servlet 5, JSP 3, EL 4, WebSocket 2)*
- Dalla shell, folder bin, eseguire lo script *startup* (set JAVA_HOME)
 - Porta 8080, configurabile in server.xml, elemento Connector

Red Hat JBoss EAP – WildFly

- **Application Server** (EJB Container / “Full”) → implementa tutte le specifiche Java EE
 - <https://developer.jboss.org/>
 - <https://wildfly.org/>
- Basato sul Web Server Undertow
- Richiede Java SE 8+ (consiglia 11) e supporta Java / Jakarta EE 8:
 - **Java Servlet 4.0**
 - **Java Server Pages 2.3**
 - **Java Expression Language 3.0**
 - **Enterprise Java Bean 3.2**
 - ...
- Dalla shell, folder bin, eseguire lo script *standalone* (set JAVA_HOME)
 - Porta **8080**, configurabile in standalone/configuration/standalone.xml
 - socket-binding-group → socket-binding – http

Eclipse per Java EE

- Conviene avere Eclipse edizione JEE, ma si può installare il plugin a parte
- Help → Eclipse Marketplace (WTP – Web Tools Platform)
 - Eclipse Enterprise Java Developer Tools
 - Eclipse JST Server Adapters
 - Eclipse Web Developer Tools
 - Eclipse XML Editors and Tools
 - Maven (Java EE) integration for Eclipse WTP
- Il progetto va visto via **Java EE Perspective**, gestione web server in uso
 - Window, Show View, Servers
 - Servers View → New → Server
- Matching tra Maven ed Eclipse del progetto: **Project Properties – Project Facets**
 - **Dynamic Web Module** 4.0; Java 11
 - **Runtimes**: Apache Tomcat / JBoss / ...

JBoss / WildFly in Eclipse

- Red Hat CodeReady Studio <https://tools.jboss.org/downloads/devstudio/>
 - Versione specifica di Eclipse per JBoss / WildFly
- Altrimenti, nella prospettiva Java EE
 - Servers → New Server: Red Hat JBoss Middleware → JBoss AS, WildFly ...
 - È necessario un restart, dopodiché i server JBoss sono disponibili
 - Servers → New Server: JBoss Community / Middleware
 - Ora è possibile accedere da Eclipse alla configurazione del server
 - XML Configuration
 - Server Details
 - Filesets
 - JMX

Eclipse Dynamic Web Project

- Approccio nativo Eclipse, alternativo a Maven
- Principali setting nel wizard
 - Target runtime
 - Tomcat, JBoss, ...
 - Dynamic Web module version: 4
 - Configuration: Default
 - Generate web.xml DD (tick)
- Project Explorer
 - WebContent: HTML e JSP
 - vs. Deployed Resources → webapp
 - Java Resources: Servlet
- Generazione del WAR
 - Export, WAR file
 - vs. Run as, Maven Install

Request – Response

- Il client manda una request al Web Server per una specifica risorsa
 - Via HTML link, form, AJAX su XMLHttpRequest, ...
- Il web server genera una response
 - File HTML, JSON, XML
 - Immagine, PDF, ...
 - Errore (404 not found, ...)
- Si comunica con il protocollo HTTP, metodi GET e POST
 - GET: eventuali parametri sono passati come parte della request URL
 - POST: i parametri sono passati come message body (o *payload*)
- Associazione tra request e un nuovo thread di esecuzione della servlet

Legame debole tra backend e frontend