

Introduction to Java EE

Java Platform, Java EE, Servlets



SoftUni Team
Technical Trainers



SoftUni
Foundation



Software University

<http://softuni.bg>

Table of Contents

1. Java Platform
2. Java Enterprise Edition
3. Apache TomEE
4. Context & Dependency Injection
5. Java Servlets



sli.do

#java-web



Java Platform

The Big Picture

- The **Java Technology** is both a **programming language** and a **platform**
- The **Java programming language**:
 - Object-oriented
 - C-like syntax
- The **Java platform**:
 - Environment
 - Where Java applications run



Java - The Big Picture (2)

- There are 4 Java platforms:

Java SE
(Standard Edition)

Java ME
(Micro Edition)

Java FX
(Effects)

Java EE
(Enterprise Edition)

Java - The Big Picture (3)

- Each Java platform:
 - Provides a JVM & an API
 - Runs applications on any compatible system
 - Is completely independent
 - Provides all of the advantages of the Java programming language

Java SE

Java ME

Java FX

Java EE



Java Enterprise Edition

Diving into the depths of Java EE

- Java EE
 - Formerly (J2EE), Currently Jakarta EE
 - A set of specifications extending Java SE
- Defined by its specification
 - Defines APIs and their interactions
- Used in various contexts, including:
 - E-commerce, Accounting, Banking Systems, etc.



■ Java SE

- Provides APIs to handle collections
- JVM is the container for Applications
- The container provides lower-level services

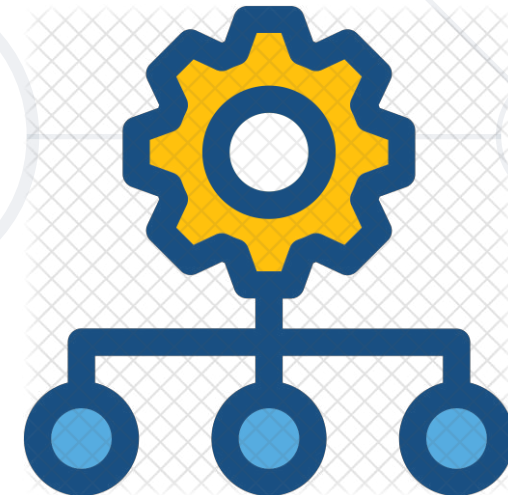
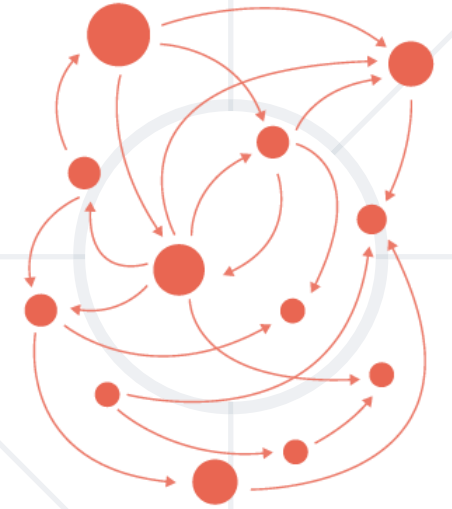


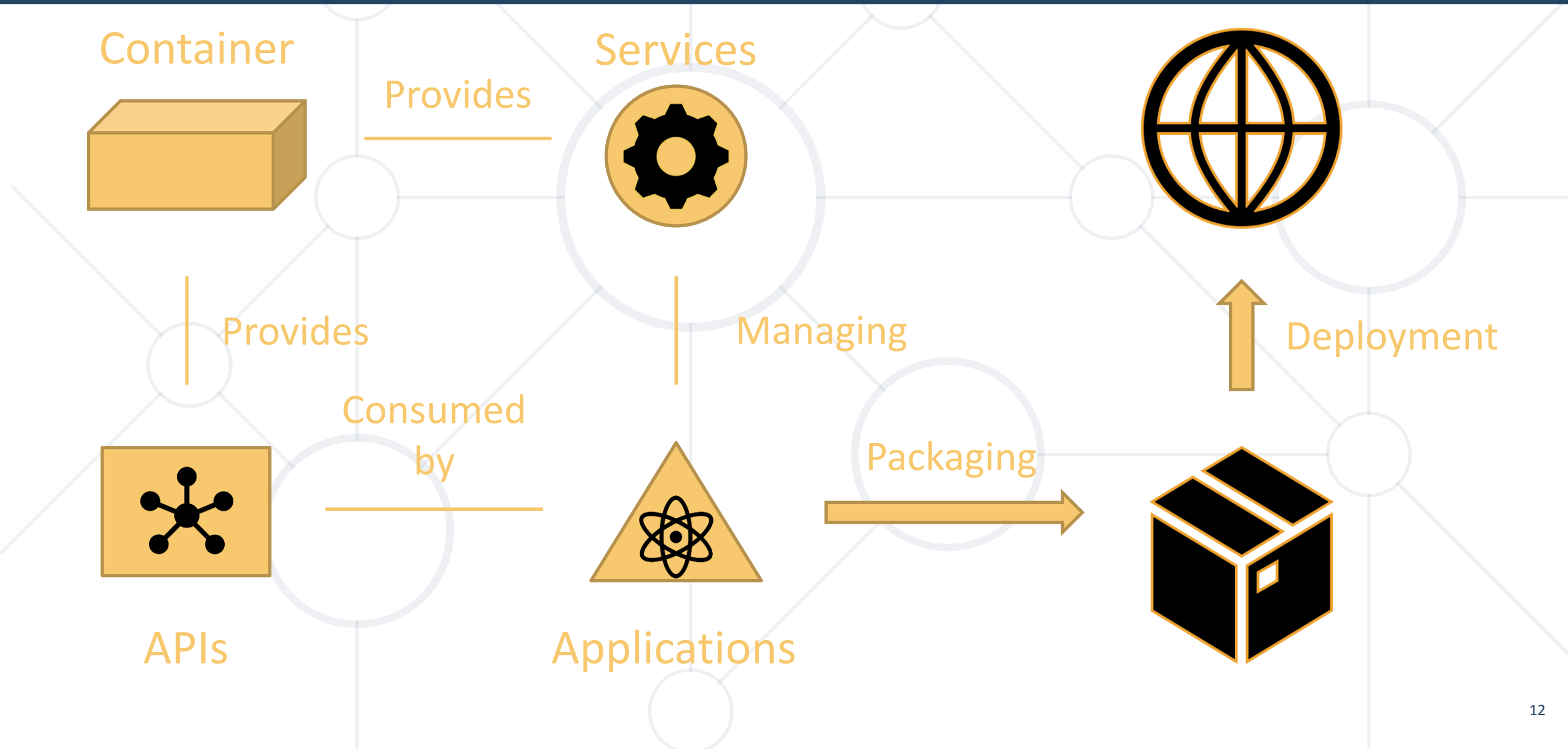
■ Java EE

- Provides APIs to handle transactions, persistence, security, messaging, etc.
- Applications are ran on a specific container
- The container provides higher-level services



- Enterprise Applications:
 - Have powerful functionalities
 - Tend to be heavy and complex
- Java EE reduces complexity
 - Provides a programming model
 - Provides a Runtime environment
 - Provides various APIs

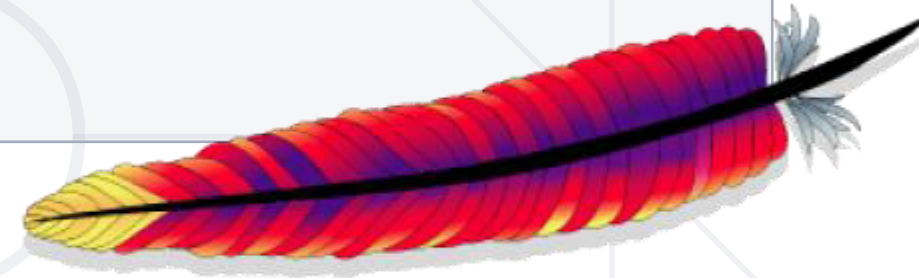




Maven Dependencies

pom.xml

```
<dependency>  
  <groupId>javax</groupId>  
  <artifactId>javaee-api</artifactId>  
  <version>8.0</version>  
</dependency>
```



- The **JCP** is the mechanism for developing **standard technical specifications** for Java technology.
 - <https://jcp.org/en/home/index>



Java
Community
Process



Apache TomEE

What is a Apache TomEE?(1)

- **Apache TomEE** (pronounced "Tommy") is the Java Enterprise Edition of Apache Tomcat (Tomcat + Java EE = TomEE) .
 - Developed by the Apache Software Foundation



What is a Apache TomEE?(2)

- Combines several Java enterprise projects including:
 - Apache OpenEJB(Enterprise JavaBeans):
 - An open source, embeddable and lightweight EJB Container System and EJB Server.
 - Has been integrated with Java EE application servers such as Geronimo and WebObjects.



What is a Apache TomEE?(3)

- **Apache OpenJPA:**
 - An open source implementation of the **Java Persistence API** specification.
 - It is an **object-relational mapping**(ORM) solution for the Java language, which simplifies storing objects in databases.



What is a Apache TomEE?(4)

- Apache MyFaces:

- An Apache Software Foundation project that **creates** and **maintains** an open-source **JavaServer Faces** implementation, along with several libraries of JSF components that can be deployed on the core implementation.



Tomcat Setup

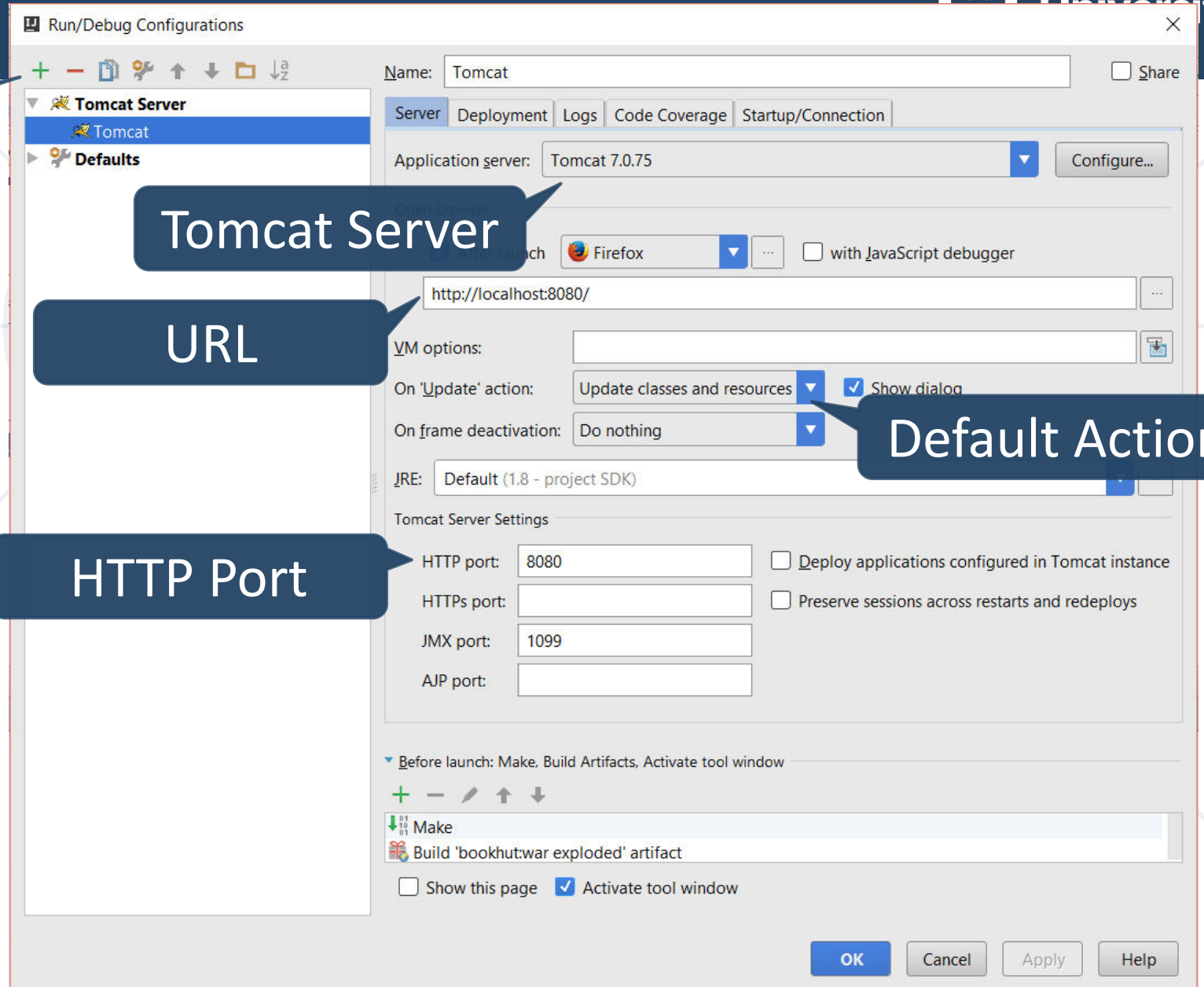
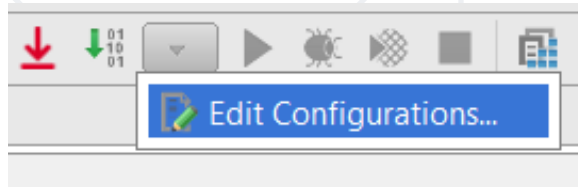
Add Tomcat

Tomcat Server

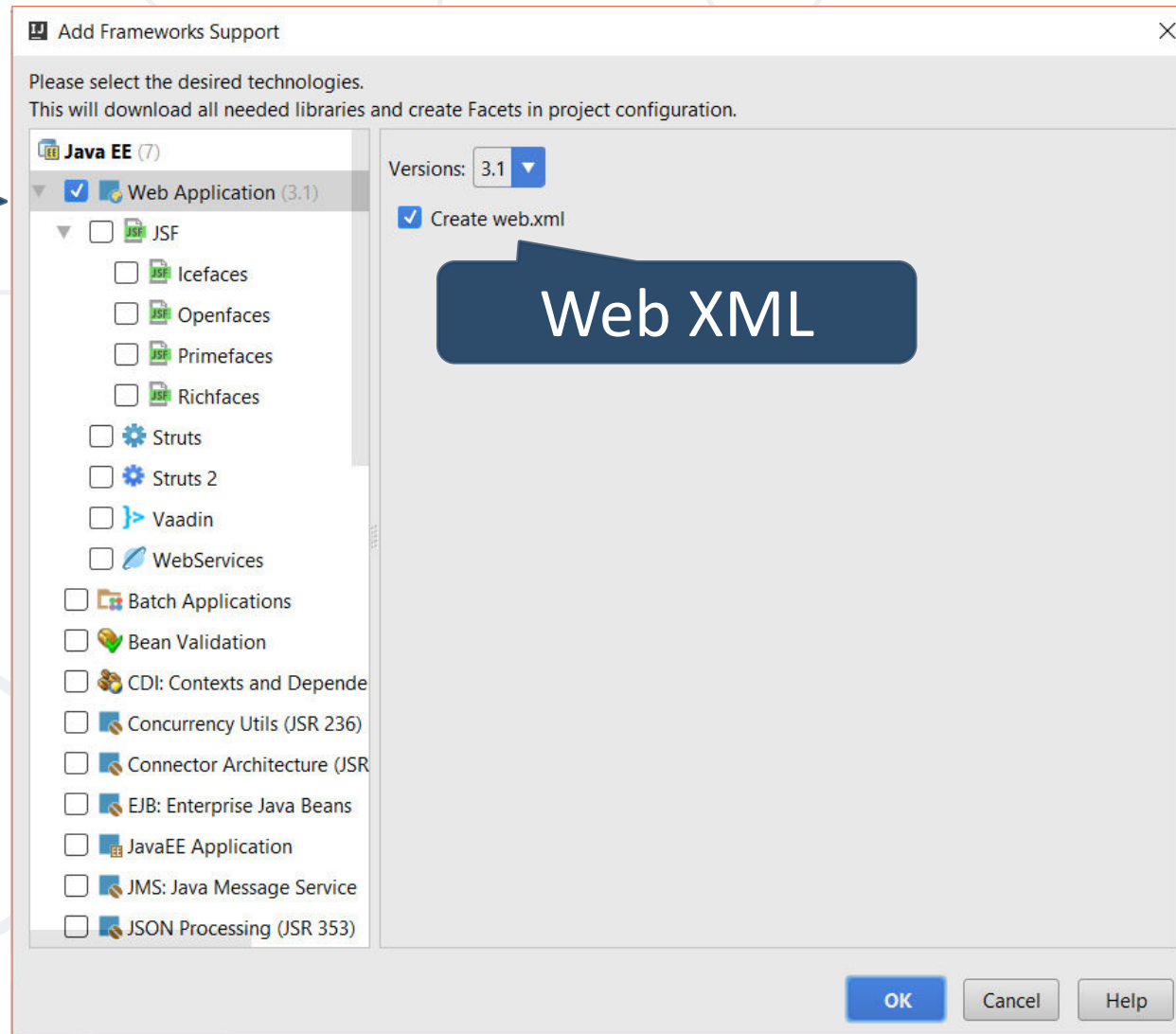
URL

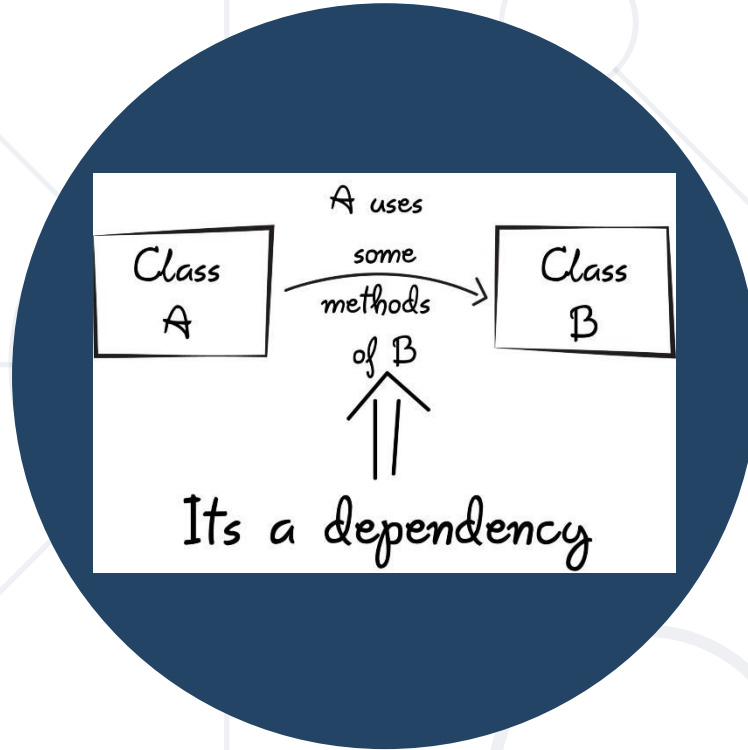
Default Action

HTTP Port



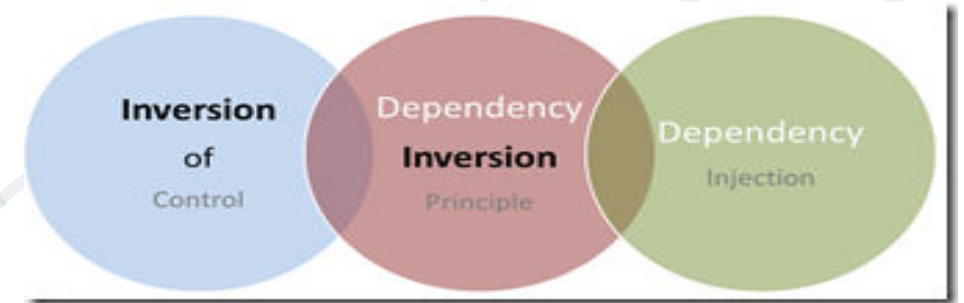
Web Support



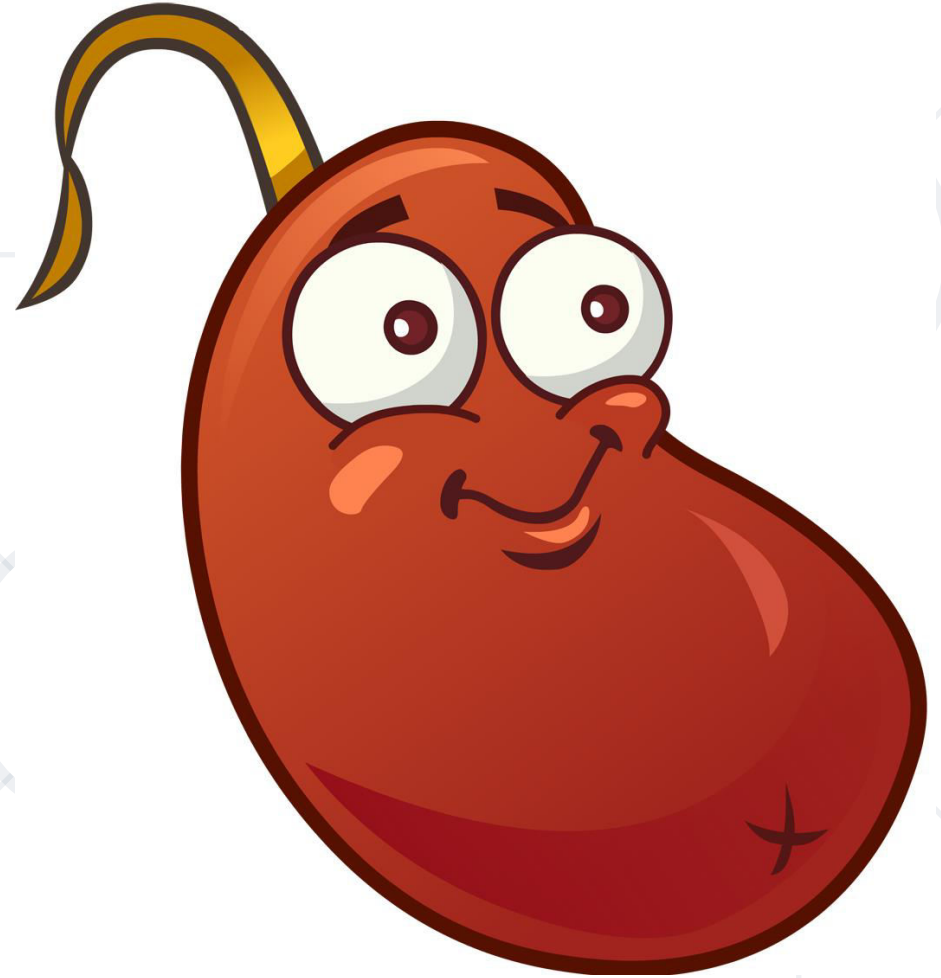


Context & Dependency Injection

- **CDI, or Context & Dependency Injection:**
 - Key Java EE specification
 - Triggered when components are packaged
- **Bean Manager:**
 - Discovers these components
 - Spots the dependencies between them
 - Injects the right reference into the right beans



- After deployment:
 - All beans are connected
- CDI Bean:
 - Just an object
 - Managed by the bean manager





Java Servlets

- **Programs** that run on a **Web/Application** server
 - **Extend** the capabilities of a Server
 - Able to respond to any type of request



Servlet Example

URL pattern

Servlet.java

```
@WebServlet("/home")
```

```
public class Servlet extends HttpServlet {  
    protected void doPost(HttpServletRequest request,  
                           HttpServletResponse response)  
        throws ServletException, IOException {  
    }  
}
```

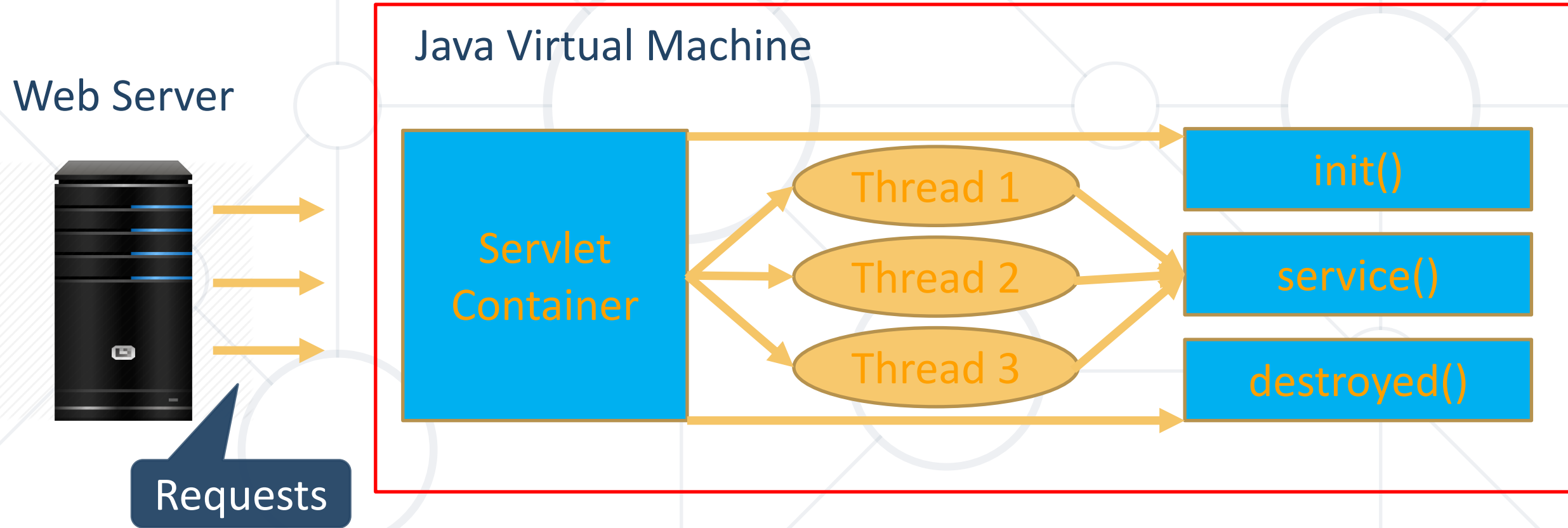
Handle
Post requests

```
protected void doGet(HttpServletRequest request,  
                     HttpServletResponse response)  
    throws ServletException, IOException {  
}
```

Handle
Get requests

```
}
```

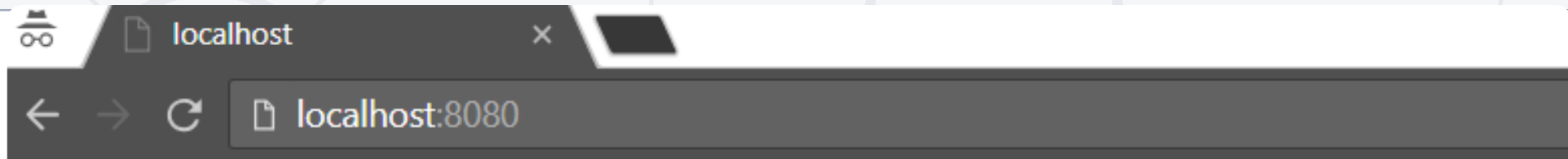
Servlet Lifecycle



Hello World!

Servlet.java

```
@WebServlet("/")  
public class Servlet extends HttpServlet {  
    protected void doGet(...) {  
        PrintWriter out = response.getWriter();  
        out.println("Hello World!");  
    }  
}
```



Hello World!

Read Parameter

Servlet.java

```
@WebServlet("/param")
public class Servlet extends HttpServlet {
    protected void doGet(...) {
        PrintWriter out = response.getWriter();
        String name = request.getParameter("name");
        out.println("Param:" + name);
    }
}
```

Param name

Param name

http://localho...etsAreAwesome X



localhost:8080/param?name=ServletsAreAwesome

Param:ServletsAreAwesome

Redirect

Servlet.java

```
@WebServlet("/form")
public class Servlet extends HttpServlet {
    protected void doGet(...) {
        response.sendRedirect("/home");
    }
}
```

Redirect to /home

Status	Method	File	Domain	Cause
▲ 302	GET	form	localhost:8080	JS document
● 200	GET	home	localhost:8080	document
○ 200	GET	home.css	localhost:8080	stylesheet

Inject Beans(1)

beans.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<beans
```

```
  xmlns="http://xmlns.jcp.org/xml/ns/javaee"
```

```
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
```

```
  http://xmlns.jcp.org/xml/ns/javaee/beans_1_1.xsd"
```

```
  bean-discovery-mode="all">
```

```
</beans>
```


Inject Beans(2)

Servlet.java

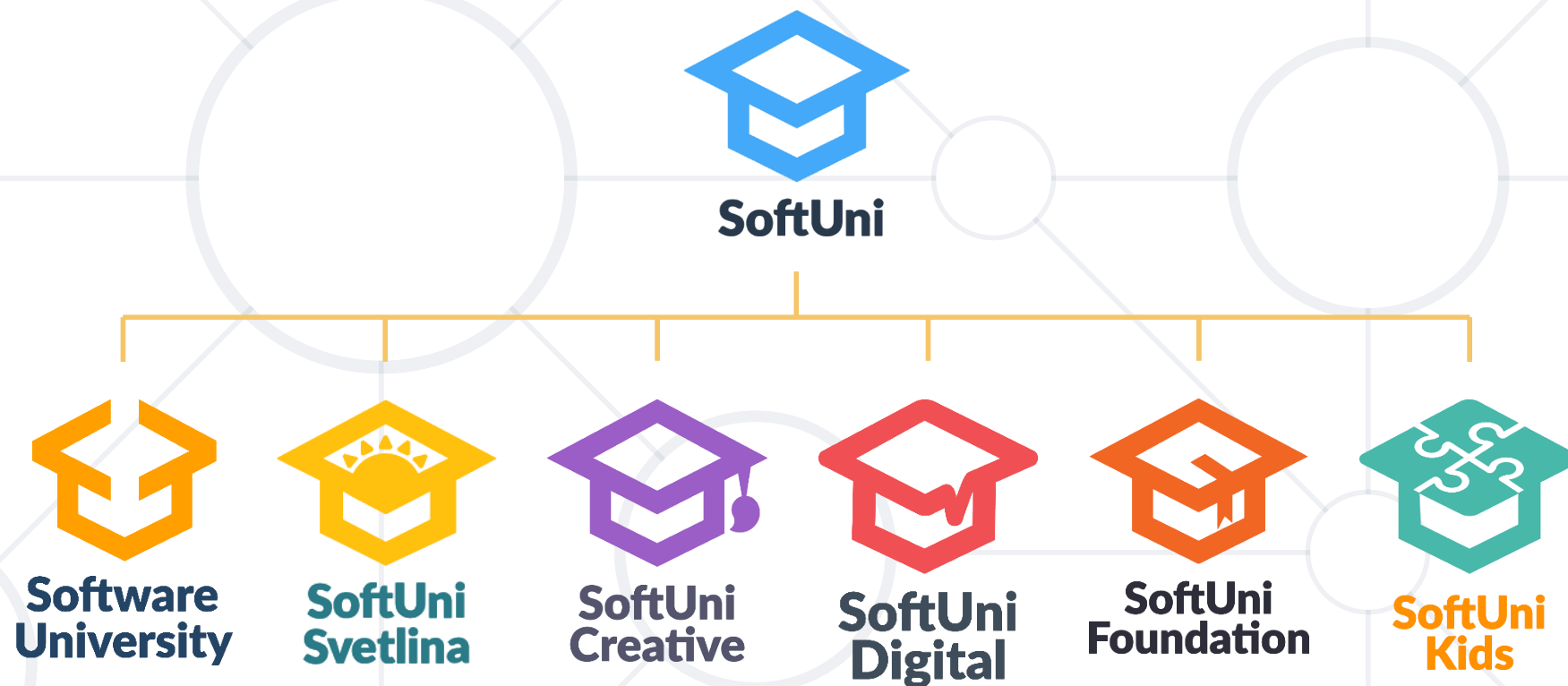
```
@WebServlet("/")
public class Servlet extends HttpServlet {
    @Inject
    private UserService userService;

    protected void doGet(...) {
        PrintWriter out = response.getWriter();
        User user = this.userService
            .getUserByUsername(request.getParameter("name"));
        out.println("User:" + user.getName());
    }
}
```

- The **Java Technology**:
 - A **platform** and a **programming language**
- **Java EE** – The Big Picture:
 - The Java EE Architecture
 - What is Apache TomEE?
 - What is CDI?
 - What are Servlets?



Questions?



SoftUni Diamond Partners



XSsoftware



SBTech



telenor



SoftwareGroup
doing it right

NETPEAK



SmartIT



æternity

**SUPER
HOSTING
.BG**

INDEAVR

Serving the high achievers



INFRAGISTICS®

LIEBHERR



Postbank

Решения за твоето утре

SoftUni Organizational Partners



OneBit
SOFTWARE



WORLD
OF
MYTHS

- Software University – High-Quality Education and Employment Opportunities
 - softuni.bg
- Software University Foundation
 - <http://softuni.foundation/>
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University Forums
 - forum.softuni.bg



- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license

