Java Backend Development Live-85

lecture-7

Agenda

- Maven
- Maven Plugins
- Web Services
- Web Server vs Application Server
- Rest clients: Postman, curl
- Http Server Demo
- Spring Framework (Intro)
- Spring vs Spring Boot

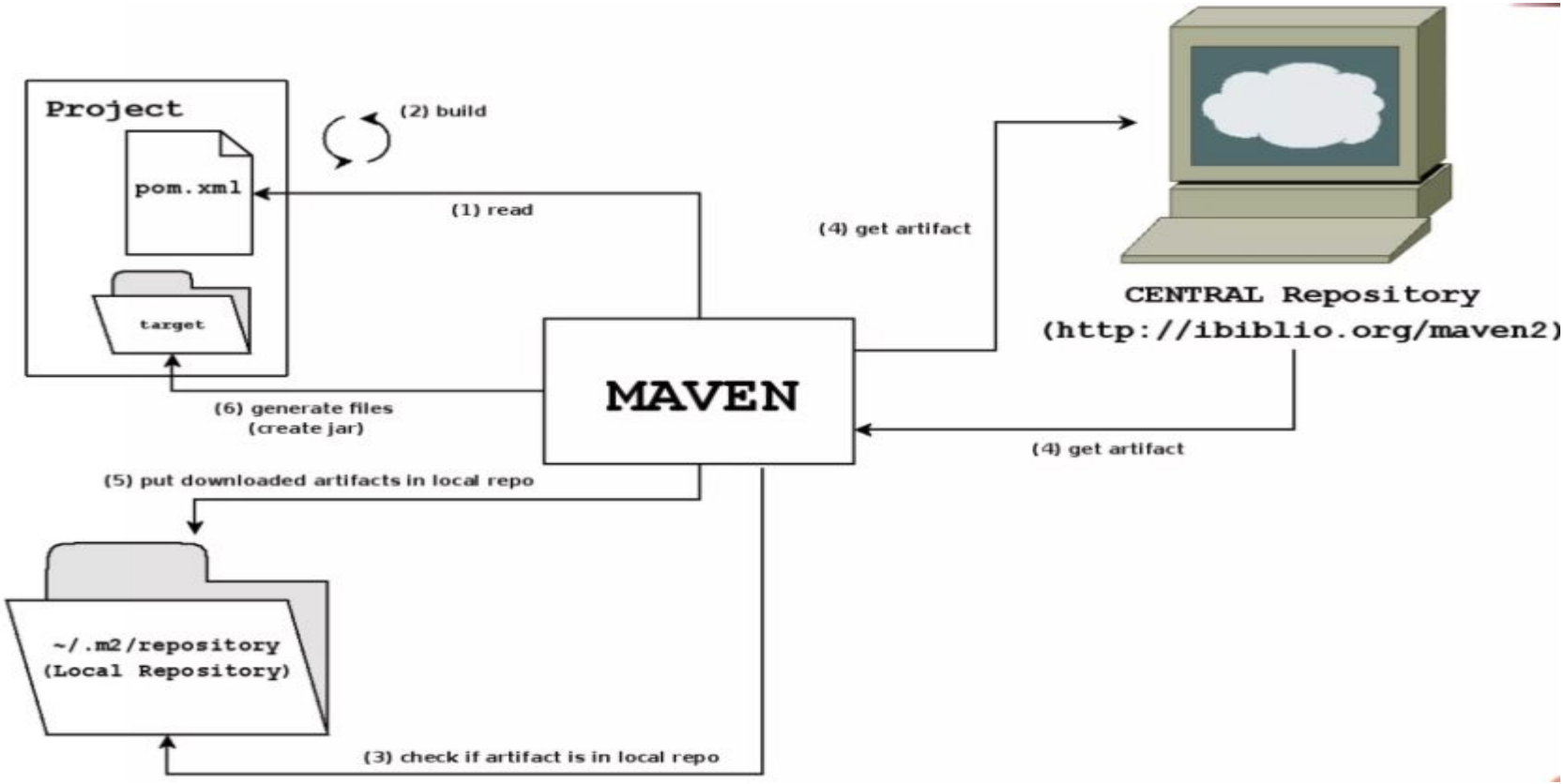
Maven

- Maven is a powerful build automation tool that is primarily used for Java-based projects.
- Maven prefers **convention over configuration.**
- Maven dynamically downloads Java libraries and Maven plug-ins from one or more repositories such as the Maven Central Repository and stores them in a local cache(.m2).
- **Project Object Model(POM)** file is an XML file that contains information related to the project and configuration information such as dependencies, source directory, plugin, goals, etc. used by Maven to build the project.

When to use Maven.

- If there are too many dependencies for the project.
- When the dependency version update frequently.
- Continuous builds, integration, and testing can be easily handled by using maven.
- When one needs an easy way to generate documentation from the source code, compiling the source code, packaging compiled code into JAR files or ZIP files, WAR.

How Maven Works ?



Maven life cycle

- default – This is the main life cycle of Maven as it is responsible for project deployment.
- clean – This life cycle is used to clean the project and remove all files generated by the previous build.
- site – The aim of this life cycle is to create the project’s site documentation.

Each life cycle is made up of a sequence of phases.

Maven Phases

- **validate** – This phase checks if all information necessary for the build is available
- **compile** – This phase compiles the source code
- **test-compile** – This phase compiles the test source code
- **test** – This phase runs unit tests
- **package** – This phase packages compiled source code into the distributable format (jar, war)

- **integration-test** – This phase processes and deploys the package if needed to run integration tests
- **install** – This phase installs the package to a local repository
- **deploy** – This phase copies the package to the remote repository

Maven executes phases in a specific order. This means that if we run a specific phase using the command such as mvn <phase>, this won't only execute the specified phase but all the preceding phases as well.

Maven Plugins

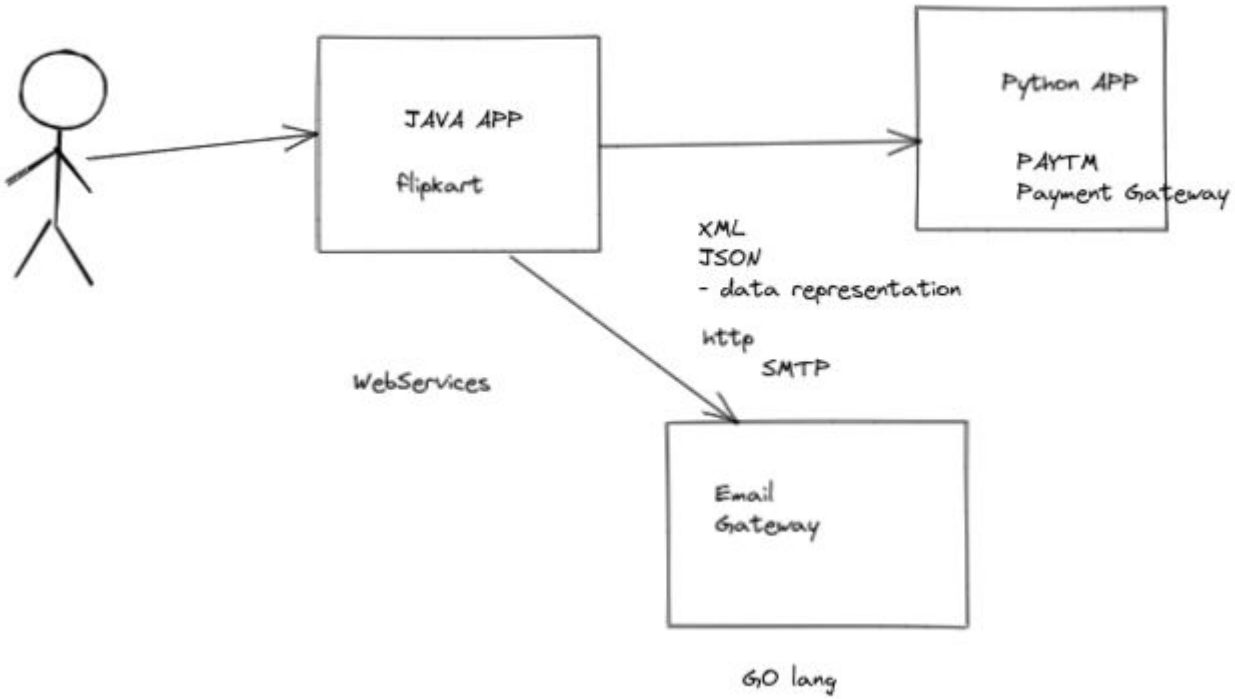
- **Definition**: Maven plugins are extensions to the Maven build tool that allow developers to perform specific tasks during the build lifecycle.
- **Purpose**: Automate repetitive tasks such as compiling code, running tests, packaging applications, and deploying artifacts.
- **Usage**: Configured in the `pom.xml` file.
- We can create custom maven plugins to address project-specific requirements.
    - Create a Maven project with `maven-plugin` packaging.
    - Extend `AbstractMojo` to define custom logic.
    - Configure and install the plugin.

What Are Web Services?

Web services are client and server applications that communicate over the World Wide Web's (WWW) with HyperText Transfer Protocol (HTTP).

As described by the World Wide Web Consortium (W3C), web services provide a standard means of **interoperating** between software applications running on a **variety of platforms and frameworks.**

Communication in web services



Web server

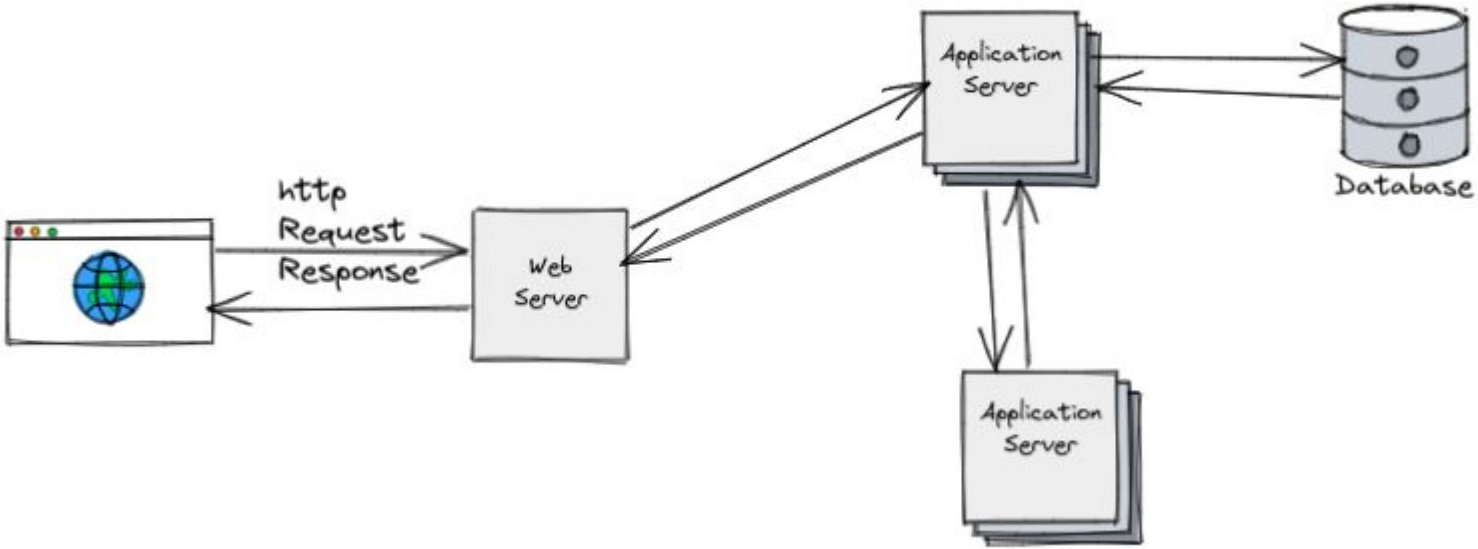
A web server accepts and fulfills requests from clients for static content (i.e., **HTML pages, files, images, and videos**) from a website. Web servers handle HTTP requests and responses only.

Examples: Apache, Nginx, httpd

Application server

An application server exposes business logic to the clients, which generates **dynamic content**. It is a software framework that transforms data to provide the specialized functionality offered by a business, service, or application. Application servers enhance the interactive parts of a website that can appear differently depending on the context of the request.

Examples: Tomcat, Jetty, Node.js


Application
Server
Database
http
Request
Response
Web
Server
Application
Server

Common in Web Server & Application Server

Web container is a part of Web Server and the Web Server is a part of Application Server.

Web Server is composed of web container, while **Application Server is composed of web container as well as EJB container**.

Most of the times these terms Web Server and Application server are used **interchangeably**.

REST Client

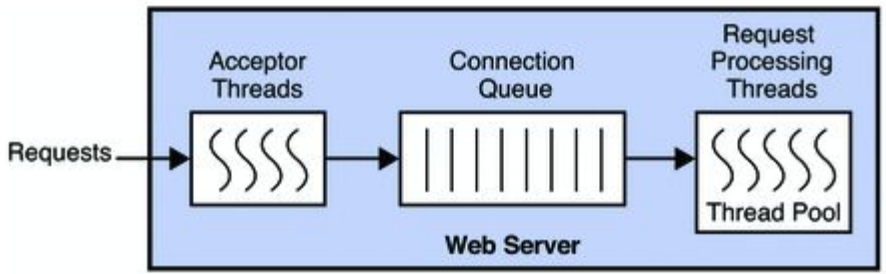
CURL

POSTMAN

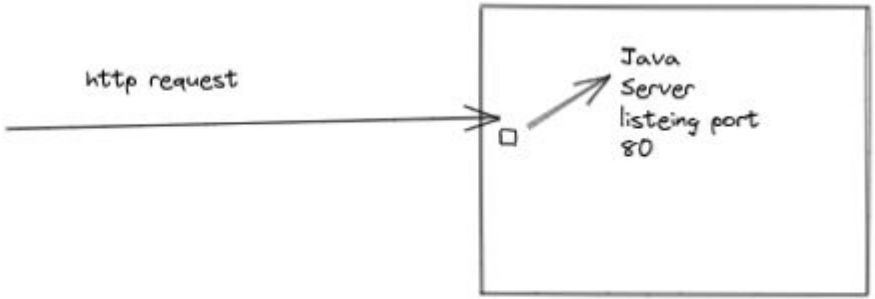
HTTP Server Demo

Servers are designed to run 24/7 and to never be turned off.

How to write such code/program ?

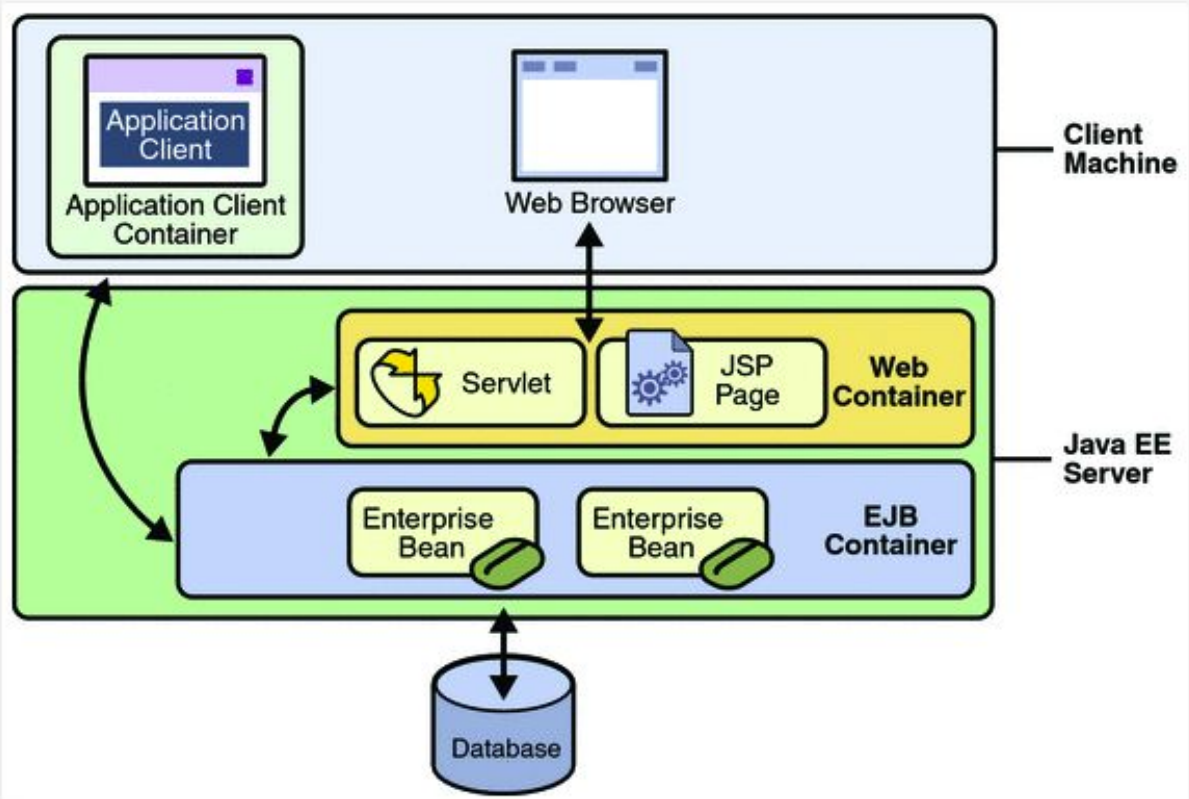


Reading Request and Writing Response via Socket



Sockets are equivalent to files.
InputStreams : Read data from socket
OutputStream: write data in socket

Java EE Server



Tomcat

Tomcat is a **servlet** and JSP container.

A Java servlet encapsulates **code and business logic** and defines how requests and responses should be handled in a Java server.

JSP is a server-side view rendering technology.

As a developer, you write the servlet or JSP page, then let **Tomcat handle the routing**.

Catalina is Tomcat's servlet container. Catalina implements Sun Microsystems' specifications for servlet and JavaServer Pages (JSP).

Spring Framework

The framework in a broader sense can be defined as a structure using which you can solve many technical problems.

Spring is a powerful lightweight application development framework used for Java Enterprise Edition (JEE).

In a way, it is a **framework of frameworks** because it provides support to various frameworks such as Struts, Hibernate, EJB, SpringSecurity, etc.

Spring vs Spring Boot

- **Spring**: A comprehensive framework for enterprise Java development that provides a wide range of features for building robust applications. It focuses on IoC, AOP, and modularity.
- **Spring Boot:** An extension of the Spring Framework that simplifies development with auto-configuration, embedded servers, and production-ready features. It aims to reduce boilerplate code and configuration, making it easier to get started with Spring-based applications.
- By using Spring Boot, you can leverage the power of the Spring Framework with greater ease and productivity, making it a popular choice for modern Java applications.