📘Complete Spring Web / Servlet Note

This note explains: - What is a **Servlet** - How to **set up a Java Servlet project** using Apache Tomcat - What is **Apache Tomcat** - What is an **Embedded Server** - How Embedded Tomcat works in **Spring Boot** - How to create a **REST API using Servlets**

🧰1. What is a Servlet?

A **Servlet** is a Java class that handles **HTTP requests** and sends **HTTP responses**.

It runs inside a **Servlet Container** (like Tomcat).

🔧Responsibilities of a Servlet:

- Receive request from browser
- Process the request
- Generate dynamic response
- Send the response back to the client

🔧Servlet Lifecycle

1. **init()** → Servlet created
2. **service()** → Called for every request
3. **destroy()** → Servlet is destroyed

🧰2. What is Apache Tomcat?

Apache Tomcat is: - A **web server** - A **Servlet container** - A **JSP engine**

It provides the environment where Servlets run.

🔧Tomcat does:

- Start a server on port (default **8080**)
- Load your web application
- Create and manage servlets
- Handle HTTP requests/responses

🧰3. Setting Up Apache Tomcat

Step ✂️— Download Tomcat

Download from: tomcat.apache.org

Step 🖋️— Extract zip folder

Step 🛏️— Add to IntelliJ / Eclipse

IntelliJ

- Go to **File → Settings → Build Tools → Application Servers**
- Add Tomcat by selecting the folder

Eclipse

- Go to **Servers** tab → Add new server → Select Apache Tomcat

---

🧰4. Creating a Servlet Project (WAR Application)

Folder Structure

```
myapp/
 ├── src/main/java/
 │     └── com.example
 │           └── HelloServlet.java
 ├── src/main/webapp/
 │     └── WEB-INF/
 │           └── web.xml
```

🧰5. web.xml — Servlet Configuration

This file tells Tomcat which servlet to load.

```
<web-app>
    <servlet>
        <servlet-name>HelloServlet</servlet-name>
        <servlet-class>com.example.HelloServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>HelloServlet</servlet-name>
        <url-pattern>/hello</url-pattern>
    </servlet-mapping>
</web-app>
```

🧰6. Creating a Servlet

```java
package com.example;

import jakarta.servlet.*;
import jakarta.servlet.http.*;
import java.io.IOException;

public class HelloServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
throws IOException {
        resp.setContentType("text/plain");
        resp.getWriter().write("Hello from Servlet!");
    }
}
```

Run Tomcat → Visit:

```
http://localhost:8080/myapp/hello
```

🧰7. What Is an Embedded Server?

Spring Boot does NOT require external Tomcat.

It includes **embedded Tomcat inside the JAR file**.

🔧How Embedded Tomcat works

- When you run a Spring Boot app, Tomcat starts inside Java process
- Your controllers become “Servlets” internally
- No need for web.xml

Embedded Tomcat comes from this dependency:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

🧰8. Difference: External Tomcat vs Embedded Tomcat

Feature	External Tomcat	Embedded Tomcat
Deployment	WAR file	JAR file
Setup	Manual	Auto with Spring Boot
Config	web.xml	Annotations (@RestController)
Run	Tomcat must be installed	`java -jar app.jar`
Port	Config in Tomcat	`application.properties`

🧰9. REST API Using Servlets (Without Spring)

Here's a simple **TODO REST API** using Servlets.

🔧GET → Fetch All Todos

🔧POST → Add New Todo

🕠Step 1: Todo Model

```
public class Todo {
    private int id;
    private String task;

    // getters + setters + constructor
}
```

🕠Step 2: TodoServlet (REST API using Servlet)

```
@WebServlet("/todo")
public class TodoServlet extends HttpServlet {

    private List<Todo> todos = new ArrayList<>();
    private int idCounter = 1;

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
throws IOException {
        resp.setContentType("application/json");

        StringBuilder json = new StringBuilder("[");
        for (int i = 0; i < todos.size(); i++) {
            Todo t = todos.get(i);
            json.append(String.format("{\"id\":%d, \"task\":\"%s\"}",
t.getId(), t.getTask()));
            if (i < todos.size() - 1) json.append(",");
        }
        json.append("]");

        resp.getWriter().write(json.toString());
    }


    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
throws IOException {
        String task = req.getParameter("task");

        Todo todo = new Todo(idCounter++, task);
        todos.add(todo);

        resp.getWriter().write("Todo added!");
    }
}
```

🧰10. How to Test API

➤ Add Todo (POST)

```
http://localhost:8080/myapp/todo?task=Learn Java
```

➤ Get Todos (GET)

```
http://localhost:8080/myapp/todo
```

🕗Final Summary

Topic	Meaning
Servlet	Java class that handles HTTP requests
Tomcat	Server + Servlet Container
External Tomcat	You install + deploy WAR
Embedded Tomcat	Comes inside Spring Boot JAR
REST API using Servlet	Manual JSON + manual routing

If you want, I can now add: - Diagrams (Servlet Lifecycle, Request Flow) - Servlet Filters - Session Management - JSP integration - Convert this note into a PDF

Just tell me!