**Title:** Getting Started with Jakarta EE  
**Julio Pochet**  
Bellevue University  
CSD 430: Server-Side Development  
Instructor: David Ostrowski  
August 14, 2025

**Getting Started with Jakarta EE**

**What**

Jakarta EE (Jakarta Enterprise Edition) is an open-source platform for building big, complex applications in Java—especially the kind businesses run behind the scenes. It’s the next chapter of what used to be Java EE, which was managed by Sun Microsystems, then Oracle, before moving to the Eclipse Foundation in 2019 (Eclipse Foundation, 2024). Jakarta EE isn’t a single program—it’s a collection of rules, APIs, and tools that help developers handle things like connecting to databases, running secure logins, managing web pages, and sending messages between systems.

**How**

I think Jakarta EE matters because it gives developers a predictable way to build projects without having to reinvent the wheel each time. Whether you’re running it on Tomcat, Payara, or WildFly, the code works pretty much the same. That’s a lifesaver when you’re working with a team or trying to keep an app running long-term. From what I’ve learned, it’s especially good for projects where front-end JSP pages need to talk to back-end Java code and databases through JDBC. Setting this up in Eclipse is pretty smooth once Tomcat and the right plugins are installed.

**Why**

Companies use Jakarta EE because it’s reliable, secure, and backed by an active community that keeps it up to date. It also helps avoid getting “locked in” to a single vendor. For me as a student, learning Jakarta EE means I’m building skills that can transfer directly into real-world enterprise projects. Plus, knowing it makes me more adaptable if I need to work on anything from a simple web app to a large-scale business platform (Jakarta EE, 2024).

**Code Example**

// HelloServlet.java

import jakarta.servlet.\*;

import jakarta.servlet.http.\*;

import java.io.IOException;

public class HelloServlet extends HttpServlet {

@Override

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

response.setContentType("text/html");

response.getWriter().println("<h1>Hello from Jakarta EE!</h1>");

}

}

**web.xml Mapping:**

<servlet>

<servlet-name>HelloServlet</servlet-name>

<servlet-class>HelloServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>HelloServlet</servlet-name>

<url-pattern>/hello</url-pattern>

</servlet-mapping>

**References**

Eclipse Foundation. (2024). *Jakarta EE releases*. <https://jakarta.ee/release/>  
Jakarta EE. (2024). *What is Jakarta EE?*. <https://jakarta.ee/>