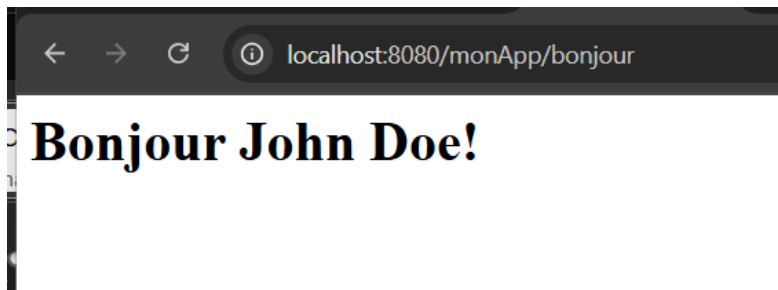
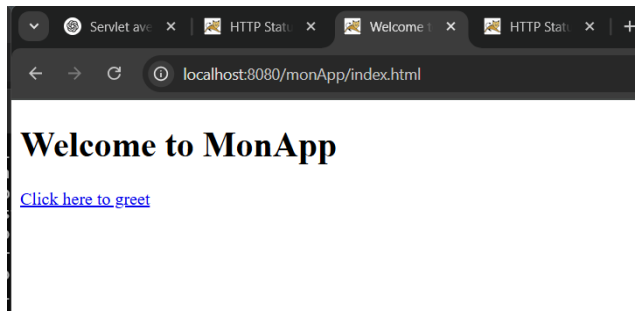


Matricule : 222178

[illegible]



## 1. The BonjourServlet.java Code

### Why We Used It:

The BonjourServlet is the heart of your application. It's a piece of code that runs on the server and handles requests from the browser. We use it because it lets us display personalized messages like "Bonjour [your name]". Without this, there would be no logic behind the server's response.

### How We Used It:

We created a class called BonjourServlet that extends HttpServlet, which is a special type of class that allows handling HTTP requests (requests from the browser). Inside this class, we wrote the logic to say "Bonjour" to the user by overriding a method called doGet. This method is automatically called when the browser sends a GET request (like typing a URL in the address bar).

### Why It's Necessary:

Without this servlet, Tomcat wouldn't know what to do when someone asks for `http://localhost:8080/monApp/bonjour`. The servlet processes the request and sends back the appropriate response, like showing your personalized

greeting. The servlet is necessary because it bridges the gap between the request (from the browser) and the response (the content shown to the user).

### **How It Works with the Other Components and Apache Tomcat:**

Tomcat, the web server, runs the servlet. When a user visits `http://localhost:8080/monApp/bonjour`, Apache Tomcat looks at the URL, finds the matching servlet (`BonjourServlet`), and runs the code inside it. The servlet then prepares the response (the greeting) and sends it back to the browser.

---

## **2. web.xml File**

### **Why We Used It:**

The `web.xml` file is like a map or guidebook for Tomcat. It tells Tomcat where to find your servlet and how to connect it with URLs. Without `web.xml`, Tomcat wouldn't know which servlet should handle which request.

### **How We Used It:**

We created entries in `web.xml` to map the servlet to a URL. For example, we told Tomcat that whenever a user visits `/bonjour` on our website, it should run the `BonjourServlet` class. We also gave the servlet a name (e.g., `bonjourServlet`) for identification.

### **Why It's Necessary:**

`web.xml` is necessary because it's how Tomcat knows what to do with different types of requests. For example, it links the URL `/bonjour` to the `BonjourServlet`, and Tomcat uses this information to direct users to the correct code.

### **How It Works with the Other Components and Apache Tomcat:**

Tomcat reads the `web.xml` file when the server starts. It then uses the information from this file to figure out which servlet handles which URL. When a user tries to visit `/bonjour`, Tomcat uses the information in `web.xml` to know that it needs to run the `BonjourServlet`.

---

### **3. The index.html File**

#### **Why We Used It:**

The index.html file is a basic webpage. It's like a welcoming front door to your application. We use it because it's the first thing users see when they visit your website. In this case, it contains a link to the servlet that says "Bonjour."

#### **How We Used It:**

We added a simple hyperlink in index.html that points to /bonjour. When users click this link, they are redirected to the servlet, which shows them the greeting.

#### **Why It's Necessary:**

Without index.html, users wouldn't have a way to access the servlet easily. It's the starting point, like a homepage, for users to interact with the application.

#### **How It Works with the Other Components and Apache Tomcat:**

The index.html file sits in the root of your web application (monApp). When a user visits the URL `http://localhost:8080/monApp/`, Tomcat serves this index.html page. If the user clicks on the "Bonjour" link, they are redirected to the servlet (`/bonjour`), which processes the request and responds with the greeting.

---

### **4. Apache Tomcat and the Role of the Server**

#### **Why We Used It:**

Apache Tomcat is the web server and servlet container that runs our code. It's necessary because it provides the environment in which our servlet code runs. Tomcat receives the HTTP requests from users, passes them to the right servlet, and sends back the HTTP responses.

### **How We Used It:**

We installed Apache Tomcat, and placed our web application (monApp) inside the webapps directory. Tomcat takes care of processing incoming requests for our app, finding the correct servlet (like BonjourServlet), running it, and sending back a response.

### **Why It's Necessary:**

Tomcat is the engine that powers your servlet. Without Tomcat, your servlet code wouldn't be able to run in response to a browser's requests. It handles the technical details of serving web content.

### **How It Works with the Other Components:**

Tomcat interacts with all of your code. It receives requests, runs the BonjourServlet, processes index.html, and sends the responses back to the browser. It also uses the web.xml to figure out what to do with each request (like directing /bonjour to BonjourServlet).

---

## **5. The classes/ Folder and Compiling Your Servlet**

### **Why We Used It:**

The classes/ folder is where we put the compiled version of our servlet (BonjourServlet.class). Java code needs to be compiled (turned into a format the computer can understand), and Tomcat runs the compiled code from this folder.

### **How We Used It:**

We compiled the BonjourServlet.java file with the javac command, and placed the resulting .class file in the classes folder, under WEB-INF/. This is where Tomcat looks for the servlet code to execute when a request is made.

### **Why It's Necessary:**

Without compiling the servlet, there would be no .class file to run, and Tomcat wouldn't know how to handle requests for /bonjour. The classes/ folder is necessary because it contains the executable code that Tomcat runs.

## How It Works with the Other Components and Apache Tomcat:

Tomcat checks the `classes/` folder for the compiled servlet code when it needs to run a servlet. When the browser requests `/bonjour`, Tomcat finds the servlet class, runs it, and sends the response back to the browser.

---

### In Summary:

- The **BonjourServlet.java** is the code that tells Tomcat how to process a request and generate a response.
- The **web.xml** is the configuration file that tells Tomcat which URL corresponds to which servlet.
- The **index.html** file provides an entry point for users to interact with your servlet.
- **Apache Tomcat** serves as the platform that runs everything, processing requests, executing code, and sending back responses.
- The **classes/ folder** contains compiled versions of the servlet, allowing Tomcat to run the servlet when requested.