

German Martinez Cagigas

DAW

Activity 3. Web architectures

Index

Content

Index.....	1
Introduction.....	2
Exercise.....	3
Apache, Tomcat, jsp and Servlet.....	3
Node.....	7
Conclusion.....	14
Bibliography.....	15

Introduction

Exercise about deploy a static web on Apache and link it through tomcat with a jsp that is related with the Servlet.










Also doing the same with static html on Apache and node js.

Exercise

Apache, Tomcat, jsp and Servlet.

Well the start doesn't have any mystery.

I created a html and added on xampp htdocs to open it as static page with apache on localhost 8081/hola.html

Este equipo > Disco local (C:) > xampp > htdocs		
Nombre		Fecha de modificac
 dashboard		27/10/2023 0:52
 img		27/10/2023 0:52
 webalizer		27/10/2023 0:51
 xampp		27/10/2023 0:52
 applications.html		15/06/2022 18:07
 bitnami.css		15/06/2022 18:07
 favicon.ico		16/07/2015 17:32
 hola.html		27/10/2023 5:44
 index.php		16/07/2015 17:32

First of all, I wanted to do something different and more related with what we are watching with Anuar in some way, the static page will be the client, the jsp the business layer and the Servlet the dd.bb(in this example a repository).

So, first I needed a button to link with the jsp.

```
<button onclick="redirigirAJSP()">Ir a JSP</button>
```

The button has a simple function that when you click on it....

```
function redirigirAJSP() {  
    window.location.href = 'http://localhost:8080/MiApp/jsp.jsp';  
}
```

It will redirect to the jsp url stored on tomcat.

The jsp its just a bridge between start and end point, so this jsp redirect to the class MyServlet, to ask for something, in this case obtain the “data of a data base”.



Here is a closest view.

```

<button onclick="obtenerListaTomcat()">Obtener Lista de Tomcat</button>
<ul id="listaTomcat"></ul>
</div>
<script>
    function obtenerListaTomcat() {
        fetch('MiServlet')
            .then(response => response.text())
            .then(data => {
                const listaTomcat = document.getElementById('listaTomcat');
                listaTomcat.innerHTML = data;
            });
    }
</script>

```

Now the Servlet is who really do the logic here, this is a simple one I created an object class and generated two objects on servlet class.

```

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
    List<Objeto> objetos = new ArrayList<>();
    objetos.add(new Objeto(1, "Objeto 1"));
    objetos.add(new Objeto(2, "Objeto 2"));

    response.setContentType("text/html");
    response.setCharacterEncoding("UTF-8");
    PrintWriter out = response.getWriter();

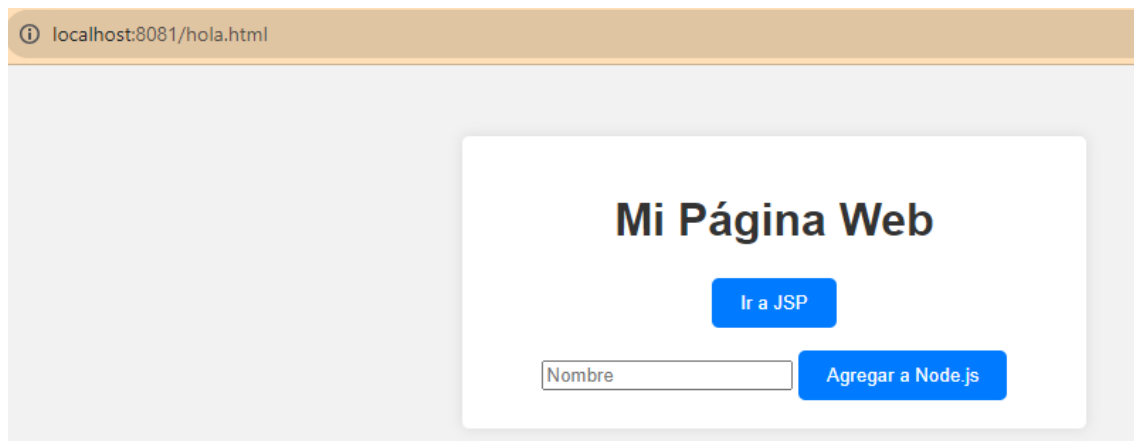
    out.println("<html>");
    out.println("<head><title>Objetos</title></head>");
    out.println("<body>");
    out.println("<h1>Lista de Objetos</h1>");
    out.println("<ul>");
    for (Objeto objeto : objetos) {
        out.println("<li>ID: " + objeto.getId() + ", Nombre: " + objeto.getNombre() + "</li>");
    }
    out.println("</ul>");
    out.println("</body>");
    out.println("</html>");
}

```

After create the objects the class gave it back on text mode (I couldn't use json, I don't know too much about it and I was getting pissed off with red text over and over).

The object class doesn't have any mystery a constructor with name and id.

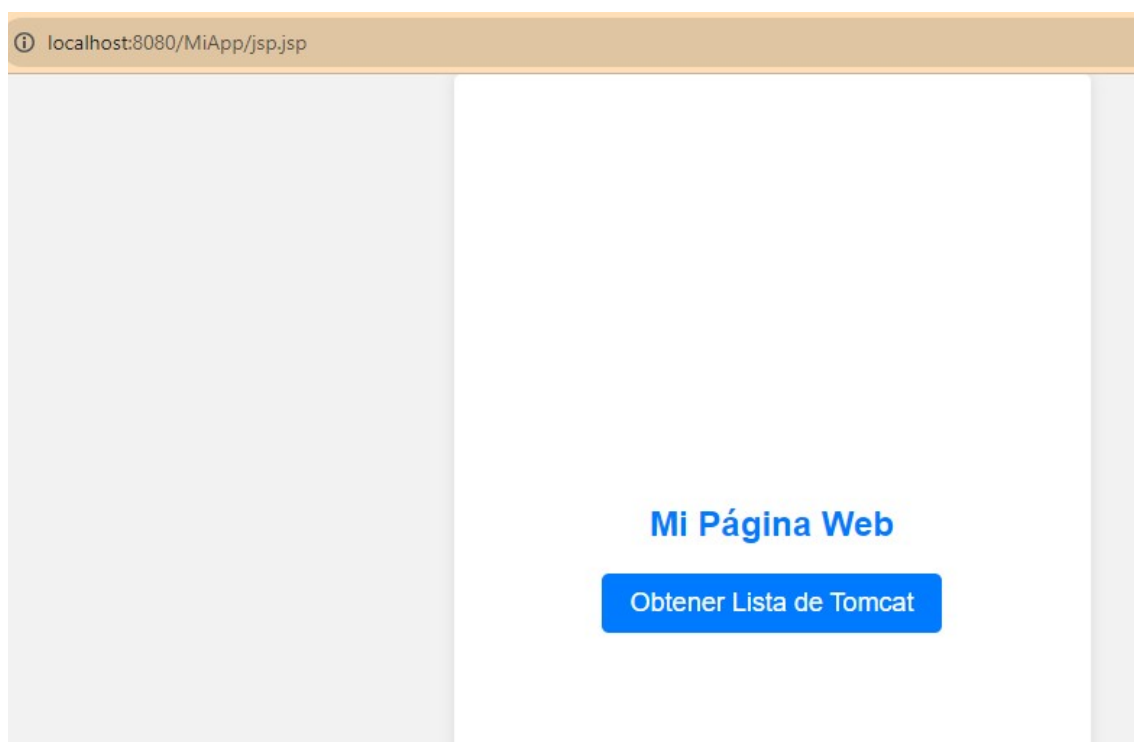
And this is how this looks.

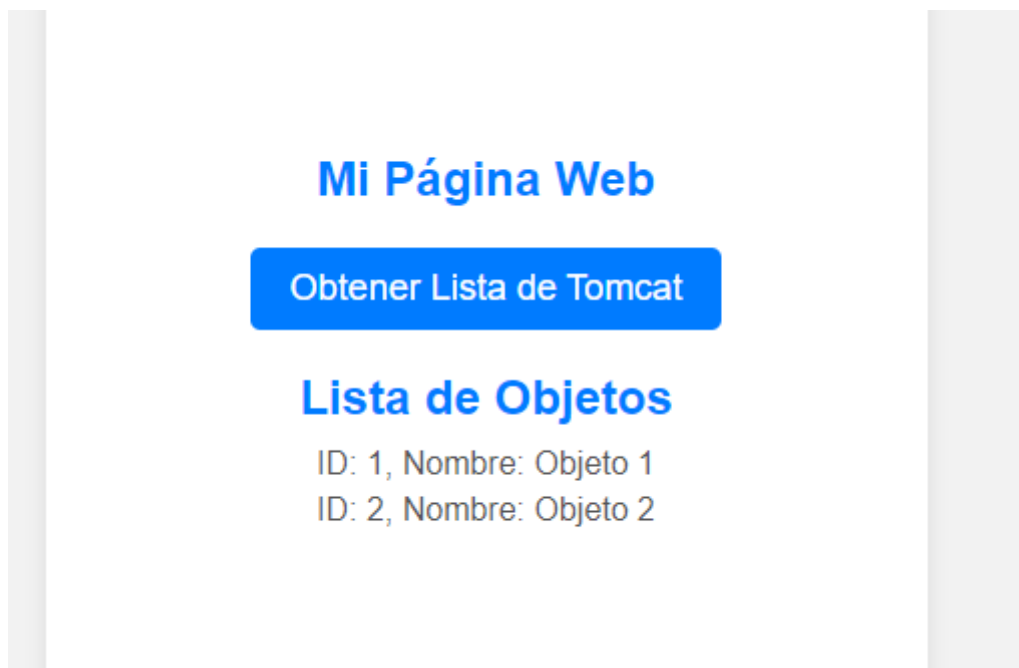


Go to JSP...

Now I realized that I did this bad, this should be in the shadows without redirect here, just see the result on the static page... I fixed it on the node method.

Anyways on this jsp now we ask for all id.





And the Servlet do the logic and give us the answer in text.

Node

Well, as always, new folder, npm init, file.js that will be the server, install express and node + app to run it.

I created my file server.js on my node folder for the proyect.

> Escritorio > adios > miProyectoNodeJS				
Nombre	Fecha de modificación	Tipo	Tamaño	
node_modules	27/10/2023 5:40	Carpeta de archivos		
package.json	27/10/2023 5:40	Archivo de origen ...	1 KB	
package-lock.json	27/10/2023 5:40	Archivo de origen ...	25 KB	
server.js	27/10/2023 5:40	Archivo de origen ...	1 KB	

Because my static page is on 8081 port and node server is in 3000 I had to enable cors because otherwise I can't send anything due to HTTP will block it to prevent malicious attacks.

The rest is just a classic install.

```
PS C:\Users\abuse\Desktop\adios\miProyectoNodeJS> npm install express --save
added 62 packages in 9s

11 packages are looking for funding
  run 'npm fund' for details
PS C:\Users\abuse\Desktop\adios\miProyectoNodeJS> node server.js
Servidor escuchando en el puerto 3000
PS C:\Users\abuse\Desktop\adios\miProyectoNodeJS> npm install cors
added 2 packages, and audited 65 packages in 4s

11 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities
PS C:\Users\abuse\Desktop\adios\miProyectoNodeJS> node server.js
Servidor escuchando en el puerto 3000
```

Ok, what will do the node server...

The client will send from static page names through method post, this name will be stored on the server.

The static page will have also a get method to obtain this added names.

So I did the easiest thing, two buttons, one for post and one for get each one related with the onclick function.

The get will be viewed on a table.

```
<form id="formularioNodeJS">
  <input type="text" id="nombre" placeholder="Nombre">
  <button type="button" onclick="enviarNombre()">Agregar a Node.js</button>
</form>

<button type="button" onclick="obtenerNombres()">Obtener Nombres</button>
<table id="tablaNombres">
  <thead>
    <tr>
      <th>Nombres</th>
    </tr>
  </thead>
  <tbody></tbody>
</table>
```

Post method.

```
fetch(url, {
  method: 'POST',
  headers: {
    'Content-Type': 'application/x-www-form-urlencoded',
  },
  body: 'nombre=' + nombre,
})
.then(response => response.text())
.then(data => {
  alert(data);
  // Limpia el campo de entrada
  nombreInput.value = '';
});
```

Get method, I can explain more on this.

The function get the answer through a json file, and with DOM taking id, creating elements of the table and adding them to the table cell, this function is generating the table with all the data.

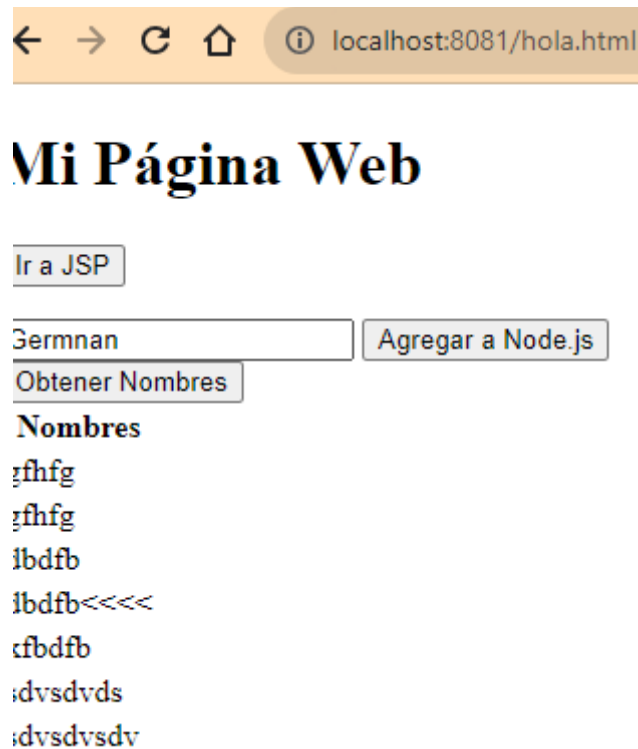
```
function obtenerNombres() {  
  fetch('http://localhost:3000/obtenerNombres')  
  .then(response => response.json())  
  .then(nombres => {  
    const tablaNombres = document.querySelector('#tablaNombres tbody');  
    tablaNombres.innerHTML = '';  
  
    nombres.forEach(nombre => {  
      const fila = document.createElement('tr');  
      const celda = document.createElement('td');  
      celda.textContent = nombre;  
      fila.appendChild(celda);  
      tablaNombres.appendChild(fila);  
    });  
  });  
}
```

This is how it looks, the css changed because I was getting really frustrated and I went for something simple.

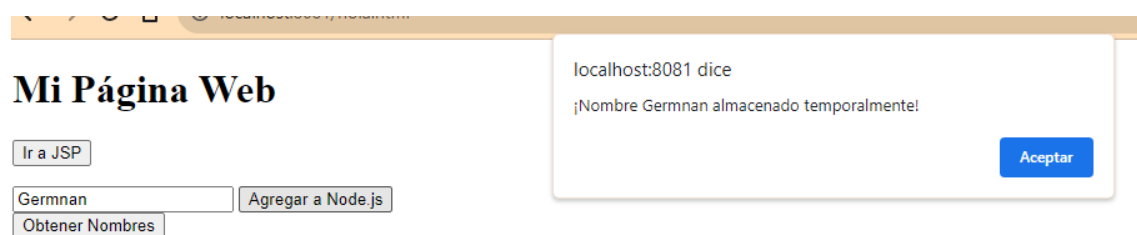


Obviously I was testing that everything was going correct....

So, now I add my name.



Send it.



There is an alert that says everything was correct.

And now, after do the post I did the get obtained also my recently added name.

Nombres
gfhfg
gfhfg
dbdfb
dbdfb<<<<
xfbdfb
sdvsdvds
sdvsdvsv
Germnan

And here is the code on node server with the url mapped on /MyServlet for post method and /obtnrNombres for get method.

All names are being stored on an Array, the alert is done by a res.send and the get method gave the answer directly on a json without convert it, that thing is more easy than in Java.

```
const express = require('express');
const cors = require('cors'); // Agrega la importacion de cors
const app = express();
const port = 3000;

app.use(cors()); // Habilita CORS

app.use(express.urlencoded({ extended: true }));

let nombres = [];

app.post('/MyServlet', (req, res) => {
  const nombre = req.body.nombre;
  nombres.push(nombre);
  res.send(`Nombre ${nombre} almacenado temporalmente!`);
});

app.get('/obtenerNombres', (req, res) => {
  res.json(nombres);
});
```

And here I forget to remove my trail of cumbs to see that im going step by step doing the things correctly.

```
app.get('/obtenerNombres', (req, res) => {  
  res.json(nombres);  
});  
  
app.listen(port, () => {  
  console.log(`Servidor escuchando en el puerto ${port}`);  
});
```

Conclusion

I had to use ChatGpt on this exercise because I was to lost, no sense the HTTP protocol that was blocking the requests, I was getting crazy, I mean I was following guides, videos, but I was having problems everywhere, maybe I wanted to do something more complicated that the activity was planed to be, but I don't know, I was too missed to do something different that we did on the last activity.

Bibliography

<https://developer.mozilla.org/es/docs/Web/HTTP/CORS>

<https://chat.openai.com/>

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch

<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/nodejs-para-una-pagina-web-con-apache-en-ubuntu/>

<https://es.stackoverflow.com/questions/470653/como-redireccionar-mi-dominio-a-mi-sitio-web-en-nodejs-en-mi-servidor-vps-hosti>

<https://es.stackoverflow.com/questions/32741/redirigir-con-servlet-y-jsp>

<https://programandojava.webnode.es/products/a13-la-redireccion-de-peticiones-con-servlets/>