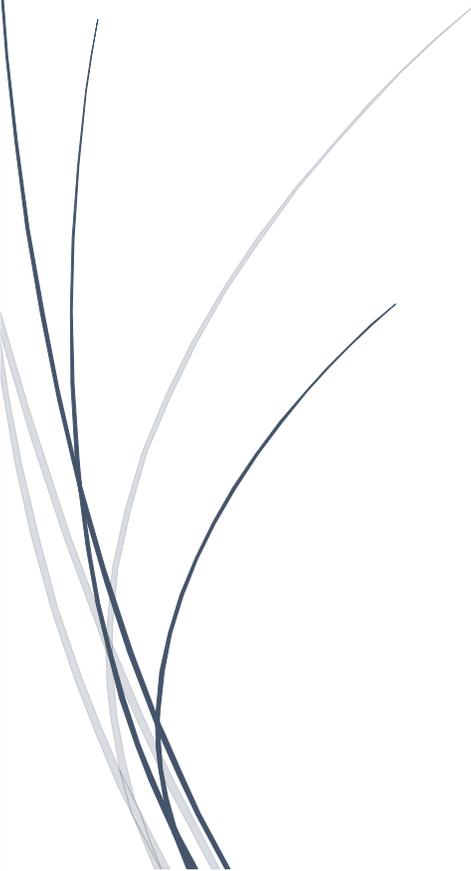




Février 2023

1st workshop

V 1.0



Dr Abdel Heq Silem
UNIVERSITE EL ARBI BEN M'HIDI

Table des matières

I.	Creating A Sign-Up java program	2
1.	Objectives	2
2.	Requirements	2
3.	Due and Assessment	2
4.	Creating A Sign-Up java program.....	2
5.	Bonus: Access your local server from the internet.....	6

I. Creating A Sign-Up java program

1. Objectives

- To teach participants how to create a sign-up java program using WAMP and MySQL.
- To introduce participants to web services and how to communicate with a WAMP server from java.
- To teach participants how to handle user input and send it to a remote server for storage and retrieval.
- Participants will learn how to connect a java to a WAMP server and send data to a MySQL database.

2. Requirements

- A computer with WAMP server installed.
- Basic knowledge of Java programming language and PHP.
- Basic understanding of MySQL database management and SQL commands.

3. Due and Assessment

Warning

- Submission deadline for workshop project is in **one week**. Late submissions will be discounted. Please prioritize completion to ensure full credit.
- Participants will be assessed through their ability to create a functioning java program that can communicate with a WAMP server and store user data in a MySQL database.
- Participants will be assessed based on their understanding and their ability to explain the steps they took to resolve any issues.

4. Creating A Sign-Up java program

Method

Step1: Setting Up the WAMP Server

1. Download and install WAMP Server from the official website: <https://www.wampserver.com/en/>
2. Once installed, start the WAMP server, and ensure that it is running.

Step2: Testing wamp server from another device within the same network

1. Find the IP address of the computer running the WAMP server: To do this, open a command prompt and type "ipconfig" (Windows) or "ifconfig" (Linux/Mac) and look for the IP address of the computer running WAMP.
2. Open a web browser on the device you want to use to access the WAMP server and enter the IP address followed by the port number used by WAMP (usually 80 or 8080). For example, "http://192.168.0.100:80".
3. If the WAMP server is working correctly, you should see the default WAMP server page.
4. If you do not see this page, check firewall settings: Make sure that the firewall on the computer running WAMP is not blocking incoming traffic on the port used by WAMP. You may need to create a new rule in your firewall settings to allow incoming traffic on this port (add *httpd.exe*). and make sure that both the device you are using to access the WAMP server and the computer running WAMP are connected to the same network. You may need to check your network settings to make sure that both devices have valid IP addresses and are able to communicate with each other.
5. If you get "403 Forbidden: You don't have permission to access/ on this server", and if you are on **WAMP 3.X**, change "**Require local**" to "**Require all granted**" in *httpd-vhosts.conf* file.
6. Restart WAMP server: Try restarting the WAMP server to see if that resolves the issue. Sometimes, restarting the server can clear up any issues that may be preventing access from other devices on the network.

If none of these steps resolve the issue, you may need to consult the WAMP documentation or seek assistance from a qualified expert.

Step 3: Creating the MySQL Database with the Corresponding Table

1. Open the *phpMyAdmin* web interface by going to <http://localhost/phpmyadmin/> in your web browser.
2. The username is "**root**" and the password is "" (blank).
3. Click on the Databases tab at the top of the page.
4. Enter a name for your database (e.g. users) and click on the Create button.
5. Select the newly created database from the list of databases on the left-hand side.
6. Click on the SQL tab at the top of the page.
7. Enter the following SQL command in the text area:

```
CREATE TABLE IF NOT EXISTS users (  
  id INT(11) NOT NULL AUTO_INCREMENT,  
  email VARCHAR(255) NOT NULL,  
  password VARCHAR(255) NOT NULL,  
  age INT(11),  
  address VARCHAR(255),  
  PRIMARY KEY (id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Step 4: Creating the java Project.

1. Open your java IDE and create a new project.
2. Set the project name and other settings as per your requirement.

Step 5: Creating the sign-up client.

1. Create a class called "http_client" in a package called "http_connexion".
2. Add the following code to create the client.

```
package http_connexion;

import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class http_client {

    public static void main(String[] args) {
        try {
            URL url = new URL("http://localhost/signup.php");
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setRequestMethod("POST");
            conn.setDoOutput(true);

            String data = "email=??&password=??&adress=??&age=??";
            DataOutputStream out = new DataOutputStream(conn.getOutputStream());
            out.writeBytes(data);
            out.flush();
            out.close();

            BufferedReader in = new BufferedReader(new InputStreamReader
                (conn.getInputStream()));

            String inputLine;
            StringBuffer response = new StringBuffer();
            while ((inputLine = in.readLine()) != null) {
                response.append(inputLine);
            }
            in.close();

            System.out.println("response = " + response.toString());
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Step 6: Sending the Sign Up Data to the WAMP Server

1. Create a new PHP file named "***signup.php***" in the "***www/***" directory of the WAMP server.

2. Add the following code to receive the sign-up data and insert it into a MySQL database.

```
<?php
if($_SERVER['REQUEST_METHOD']=='POST'){

    $email = $_POST['email'];
    $password = $_POST['password'];
    $age = $_POST['age'];
    $address = $_POST['address'];

    require_once('db_config.php');

    $sql = "INSERT INTO users (email, password, age, address) VALUES ('$email', '$password', '
$age', '$address')";

    if(mysqli_query($con, $sql)){
        echo "sign-up was successful";
    }else{
        echo "error in sign-up";
    }

    mysqli_close($con);
}
?>
```

3. Create a new PHP file named db_config.php in the same directory with the following code to connect to the MySQL database:

```
<?php

define('HOST','localhost');
define('USER','root');
define('PASS','');
define('DB','users');
$con = mysqli_connect(HOST,USER,PASS,DB) or die('Unable to Connect');
?>
```

Step 7: Run the application and test the sign-up functionality.

1. Execute the java file and test the sign-up functionality.

Hint ⓘ

Use Notepad++ for editing PHP files as it is a powerful and versatile text editor that can improve your coding efficiency and accuracy. You can download Notepad++ for free from their official website: <https://notepad-plus-plus.org/downloads/>.

5. Bonus: Access your local server from the internet

Method

Step1: Attribute an address to your Dynamic Ip

1. Create an account on the *No-IP website*.
2. Login to your account, in the dashboard go to *Dynamic DNS > No-IP hostnames* and click in *Create hostname*.
3. Choose the *name of your hostname* and select the *domain name* and click on *create hostname*.
4. Go to *Dynamic DNS > Dynamic Update Client* and download the *DUC program* to keep your Ip address updated.
5. Install the DUC program and select *your hostname* to be updated (your Ip will be updated every 5 min).

Step2: Open the port in your firewall and router to receive outside requests.

1. Search for *firewall* in windows and access *advanced security*.
2. Add a *new rule* in *InboundRules* from *Action > New rule* with the following information:
 - a. *Rule Type: Port*
 - b. *Protocol and Port: TCP with Port 80*
 - c. *Action: Allow all connection.*
 - d. *Keep Profile as it is.*
 - e. *Name: choose a name for your rule (Wamp will be great).*
3. Now go to your *router login* interface by entering the following address in your browser *192.168.1.1*.
4. Enter your *username* and *password* (usually *admin, admin* and can be found behind the router).
5. Search for *Port Forwarding* and fill the required fields as follow:
 - a. *Name: Wamp.*
 - b. *Protocol: TCP*
 - c. *Leave Wan host start and end Ip blank.*
 - d. *Wan start and end ports: 80.*
 - e. *Lan host Ip address: enter the Ip address of your server (your pc and can be found using ipconfig)*
 - f. *Lan host start and end ports: 80.*
6. Enable this forwarding (check the box of enable) and click on the add button to confirm the port forwarding.

Step 3: add a html page into your server.

1. Create an html file "**success.html**" in Wamp "**www/**" folder that contains a simple display "you have successfully configured your server to receive external requests."
2. Override the **index.php** file with the following code to **redirect requests** to your server to the **success.html page**.

```
<?php
header ("Location:success.html",true,301);
exit();
?>
```

3. Finally test your server from another device that is connected to another network (from your phone connected with 4G as example).
4. If you get "403 Forbidden: You don't have permission to access/ on this server", and if you are on WAMP 3.X, change "Require local" to "Require all granted" in httpd-vhosts.conf file.