Preconditions and Laboratory Requirements before executing the activity

1. Locate your drive “D:” and create a folder that matches your login account for the class. (e.g. “spelec1aa03”)
2. Inside this folder create a folder named “php”. (You will use this later on when you install php).
3. Go to you drive “H:” which is you network account drive and create a folder called “phpsite”. This will be used as the DocumentRoot and <Directory> directive settings in configuring the Apache web server.
4. In the Apache server activity “C:/Apache24” or “C:/Apache24/bin” are commonly used as examples. This not applicable in your case since this will now reflect the actual location where you will install the Apache web server which will be “D:/<your login account>/Apache24” and “D:/<your login account>/Apache24/bin” respectively.
5. Likewise, In the PHP activity “C:/php” is commonly used as an example. This not applicable in your case since this will now reflect the actual location where you will install the php compiler which will be “D:/<your login account>/php”.
6. When you reach the execution of the web server for the first time this will run the Windows firewall. You are to uncheck Domain access and check only the Private accessibility setting. Once you have allowed this the administrator password is required to finalize this process. You will need the assistance of the lab assistant to key in the administrator credentials to allow the process to progress.
7. Configuring the configuration files for Apache and PHP will be depended on where these files are located. You can edit them by using Visual Studio code.
8. You do not need to execute the optional steps.
9. The recommended approach to adding PHP to apache is as a module but you are free to check out the CGI approach as well. Whichever will work for you.

Apache Web Server Installation and Configuration Guide for Windows (Without Installing as a Service)

Overview

This guide provides instructions on how to install and configure the Apache HTTP web server on Windows without installing it as a service. We will cover key configuration areas, including `ServerRoot`, `Listen`, and `DocumentRoot`.

Prerequisites

- A Windows machine with administrator privileges.
- Internet access for downloading Apache.
- Basic knowledge of using Windows Command Prompt or PowerShell.

Step 1: Download Apache Web Server

1. **Download Apache HTTP Server** from the Apache Lounge:
    - Go to https://www.apachelounge.com/download/.
    - Download the latest version of the **Apache HTTP Server (httpd)** binary for Windows (zip file).
2. **Unzip the Apache package**:
    - Extract the contents of the `.zip` file to a directory, for example, `C:\Apache24`.
    - **Addendum:** you only need the Apache24 folder inside the zip file. Copy this folder to the recommended laboratory location which will in drive "D:" and inside you login account folder.

Step 2: Configuring Apache

The main Apache configuration file is located in `C:\Apache24\conf\httpd.conf`. This file contains various directives to configure the behavior of the web server.

1. `ServerRoot`

The `ServerRoot` directive defines the top-level directory where Apache's configuration files and logs are stored. By default, this is set during installation.

Default Setting:

```
ServerRoot "C:/Apache24"
```

If you move the Apache directory, ensure that `ServerRoot` reflects the new location. Also, make sure to use forward slashes (`/`) instead of backslashes (`\`), as Apache on Windows expects Unix-style paths.

You can also set the new location in the SRVROOT variable. This will automatically map to the `ServerRoot` directive

```
Define SRVROOT "C:/Apache24"

ServerRoot "${SRVROOT}"
```

2. `Listen`

The `Listen` directive tells Apache which IP address and port to use for incoming requests. By default, Apache listens on port 80.

Default Setting:

```
Listen 80
```

To change the port or specify a particular IP address:

1. To listen on port `8080`, modify the `Listen` directive: (recommended)

   ```
   Listen 8080
   ```

2. To listen on a specific IP address and port (e.g., IP `192.168.1.100` and port `8080`):

   ```
   Listen 192.168.1.100:8080
   ```

Ensure that Windows Firewall allows traffic on the new port by creating an inbound rule.

3. `DocumentRoot`

The `DocumentRoot` directive specifies the directory from which Apache serves files to clients. By default, this points to the `htdocs` directory.

Default Setting:

```
DocumentRoot "C:/Apache24/htdocs"
```

To change the document root to another folder (e.g., a custom website folder):

1. Modify the `DocumentRoot` directive in `httpd.conf`:

   ```
   DocumentRoot "path/to/your/website"
   ```

2. Update the corresponding `<Directory>` block to ensure Apache has access to serve files from this directory:

   ```
   <Directory "path/to/your/website">
       Options Indexes FollowSymLinks
   ```

```
    AllowOverride None
    Require all granted
</Directory>
```

3. Save the changes and restart Apache (as described in the next section).

Step 3: Start and Stop Apache Manually

Since you aren't running Apache as a service, you will need to start and stop Apache manually.

To Start Apache:

1. Open **Command Prompt** as Administrator.
2. Navigate to the Apache `bin` directory:

```
cd C:\Apache24\bin
```

3. Run Apache:

```
httpd
```

Apache will continue running in this command window until manually stopped or the window is closed.

To Stop Apache:

- Simply close the Command Prompt window where Apache is running, or press `Ctrl + C` to stop the process.

Step 4: Testing Apache Configuration

Before restarting Apache, it's a good practice to test the configuration to ensure there are no errors.

1. Open **Command Prompt** as Administrator.
2. Navigate to the Apache `bin` directory:

```
cd C:\Apache24\bin
```

3. Run the following command to test the configuration:

```
httpd -t
```

If no errors are reported, you can restart Apache manually to apply any changes you made to the configuration.

Step 5: Additional Configuration (Optional)

Virtual Hosts

You can configure **Virtual Hosts** to host multiple websites on the same Apache instance. Here's how:

1. Open `httpd.conf` and uncomment the following line:

```
Include conf/extra/httpd-vhosts.conf
```

2. Open the virtual hosts configuration file (`C:\Apache24\conf\extra\httpd-vhosts.conf`) and add your virtual host configuration:

```
<VirtualHost *:80>
    ServerAdmin admin@example.com
    DocumentRoot "C:/path/to/your/website"
    ServerName www.example.com

    <Directory "C:/path/to/your/website">
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog "logs/example.com-error.log"
    CustomLog "logs/example.com-access.log" common
</VirtualHost>
```

3. Save the file and manually restart Apache for the changes to take effect.

Step 6: Running Apache Automatically on Startup (Optional)

If you want Apache to run automatically at startup without installing it as a service, you can create a batch file to start Apache and place it in your **Startup** folder.

1. Create a batch file (e.g., `start-apache.bat`) with the following content:

```
@echo off
cd C:\Apache24\bin
httpd
```

2. Place this batch file in the Windows startup folder:
    - Press `Win + R`, type `shell:startup`, and press Enter.
    - Copy the `start-apache.bat` file into this folder.

This will automatically start Apache when you log in to your Windows account.

Conclusion

You have successfully installed and configured the Apache HTTP web server on Windows without installing it as a service. This guide covered key settings like `ServerRoot`, `Listen`, and `DocumentRoot`, and explained how to manually start and stop the Apache server. You can now serve content from your machine and configure additional features such as virtual hosts.

Instruction Manual: Installing and Configuring PHP on Windows

This guide will walk you through installing and configuring PHP on a Windows operating system. It will include steps for configuring `extension_dir`, setting dynamic extensions, and attaching PHP to Apache both as a module and as a CGI.

Step 1: Download and Install PHP

1. **Download PHP**
   - Visit the https://windows.php.net/download and download the latest **Thread Safe** version for Windows. The Thread Safe version is required for running PHP as an Apache module.
   - Extract the contents of the downloaded file to `C:\php` (or any desired directory).
   - Addendum: You will be extracting the contents to the folder "D:/<your login account>/php".

Step 2: Configure PHP

1. **Rename `php.ini`**
   - Navigate to the folder where you extracted PHP.
   - Locate the file `php.ini-development` and rename it to `php.ini`.
2. **Edit `php.ini`**
   - Open the `php.ini` file in a text editor (e.g., Notepad).
3. **Set `extension_dir`**
   - Locate the line that specifies `extension_dir` (around line 741). By default, it looks like this:

     ```
     ;extension_dir = "ext"
     ```

   - Remove the semicolon (`;`) to uncomment the line and modify it to point to the extensions folder:

     ```
     extension_dir = "ext"
     ```

   - This tells PHP where to find its extensions.
4. **Enable Dynamic Extensions**
   - In the `php.ini` file, find the section that lists the available extensions. It will look like this:

     ```
     ;extension=curl
     ;extension=mbstring
     ;extension=mysqli
     ```

   - Remove the semicolon (`;`) in front of the extensions you want to enable. For example:

     ```
     extension=curl
     ```

```
extension=mbstring
extension=mysqli
```

- Recommended extensions for you to enable are:
  1. curl
  2. fileinfo
  3. gd
  4. intl
  5. mbstring
  6. openssl
  7. pdo_mysql
  8. pdo_slite
  9. zip
- Save and close the `php.ini` file.

Step 3: Attach PHP to Apache

Option 1: Running PHP as an Apache Module

1. **Configure Apache to Use PHP as a Module**
   - Open the Apache configuration file `httpd.conf` (located in you Apache Server install directory).
   - Add the following lines to the end of the `httpd.conf` file to tell Apache to use PHP:

```
LoadModule php_module "C:/php/php8apache2_4.dll"
AddType application/x-httpd-php .php
PHPIniDir "C:/php"

<Directory "C:/php">
    Require all granted
</Directory>
```

   - Save and close `httpd.conf`.
2. **Restart Apache**
   - Open the Command Prompt as an administrator and restart Apache:

```
httpd
```

Option 2: Running PHP as a CGI

1. **Configure Apache to Use PHP as CGI**
   - Open the `httpd.conf` file again.
   - Add the following lines to enable PHP as a CGI script:

```
SetEnv PHPRC C:/php
```

```
ScriptAlias /php/ "C:/php/"
AddType application/x-httpd-php .php
Action application/x-httpd-php "/php/php-cgi.exe"

<Directory "C:/php">
     Require all granted
</Directory>
```

- Save and close `httpd.conf`.

2. **Restart Apache**
   - Open the Command Prompt and restart Apache again:

     ```
     httpd
     ```

Step 4: Test the PHP Installation

1. **Create a Test PHP File**
   - Navigate to your Apache web root (usually `C:\Apache24\htdocs\`).
   - Create a new file called `info.php` with the following content:

     ```
     <?php
        phpinfo();
     ```

2. **Access the PHP File**
   - Open your web browser and navigate to `http://localhost/info.php`.
   - If everything is set up correctly, you will see a PHP information page detailing your PHP configuration.

Additional Notes:

- **Verifying Extension Configuration**
  To ensure the correct loading of PHP extensions, scroll down on the `phpinfo()` page and verify the loaded extensions.
- **Thread Safe vs Non-Thread Safe PHP**
  When running PHP as an Apache module, always use the Thread Safe version of PHP, which is designed for multi-threaded environments like Apache. For CGI, the Non-Thread Safe version may be used, though using Thread Safe is generally recommended for compatibility.

By following this manual, you have successfully installed and configured PHP on Windows, attached it to Apache, and tested the setup.