

macOS / macOS Sonoma

You can make a difference in the Apple Support Community!

When you [sign up with your Apple ID](#), you can provide valuable feedback to other community members by [upvoting helpful replies and User Tips](#).



etresoft

Level 8 (48,226 points)

Setting up a local web server on macOS 14 "Sonoma"

67

16038

Last modified Mar 15, 2024 7:41 AM

Here is my definitive guide to getting a local web server running on macOS 14 "Sonoma". This is meant to be a development platform so that you can build and test your sites locally, then deploy to an internet server. This User Tip only contains instructions for configuring the Apache server and Perl module. I have another User Tip for installing and configuring [MySQL](#) and [email](#) servers.

Note: This user tip is specific to macOS 14 "Sonoma". Pay attention to your OS version. There have been significant changes since earlier versions of macOS. Another note: These instructions apply to the client versions of OS X, not Server. Server does a few specific tricks really well and is a good choice for those. For things like [database](#), web, and [mail](#) services, I have found it easier to just setup the client OS version manually.

Important note: Over the past few years, 3rd party "package managers" like Homebrew and MacPorts have become extremely popular. This User Tip assumes that no 3rd party package manager has been installed. If you have installed Homebrew or similar, these instructions will not work. In that case, I recommend you contact Homebrew support.

Requirements:

1. Basic understanding of Terminal.app and how to run command-line programs.
2. Basic understanding of web servers.
3. Basic usage of vi. You can substitute nano if you want.

Lines in **bold** are what you will have to type in. Replace <your short user name> with your short user name.

Here goes... Enjoy!

To get started, edit the Apache configuration file as root:

```
sudo vi /etc/apache2/httpd.conf
```

Sorry, but PHP is **NOT** included in Sonoma. Apple even includes the following note at line 187 of this file:

```
#PHP was deprecated in macOS 11 and removed from macOS 12
```

Unfortunately, Perl will also not work with Apache in Sonoma. Sonoma ships with a broken version of Perl that will not work with the mod_perl module. See this bug report: <https://github.com/Perl/perl5/issues/17154>

So for now, we'll go old-school and use the ancient mod_cgi method. As an added benefit, if you install your own PHP interpreter, you can run PHP scripts this way too. Your CGI scripts will run more slowly than with the

`mod_*` versions.

If you want to run CGI scripts, you will need to do the following:

Enable `mod_cgi` by uncommenting line 174 changing:

```
#LoadModule cgi_module libexec/apache2/mod_cgi.so
```

to

```
LoadModule cgi_module libexec/apache2/mod_cgi.so
```

(Note that this is dependent on the state of the `mpm_prefork_module`, which is enabled by default. This is User Tip is meant to be as simple as possible. Don't overcomplicate things.)

Enable personal websites by uncommenting the following at line 184:

```
#LoadModule userdir_module libexec/apache2/mod_userdir.so
```

to

```
LoadModule userdir_module libexec/apache2/mod_userdir.so
```

(See below for an important new ACL change that are required in Sonoma for personal websites.)

and do the same at line 521:

```
#Include /private/etc/apache2/extra/httpd-userdir.conf
```

to

```
Include /private/etc/apache2/extra/httpd-userdir.conf
```

Now save and quit.

Open the file you just enabled above with:

```
sudo vi /etc/apache2/extra/httpd-userdir.conf
```

and uncomment the following at line 16:

```
#Include /private/etc/apache2/users/*.conf
```

to

```
Include /private/etc/apache2/users/*.conf
```

Save and exit.

Lion and later versions no longer create personal web sites by default. If you already had a Sites folder, it should still be there. To create one manually, enter the following:

```
mkdir ~/Sites
```

```
echo "<html><body><h1>My site works</h1></body></html>" > ~/Sites/index.html.en
```

While you are in `/etc/apache2`, double-check to make sure you have a user config file. It should exist at the path: `/etc/apache2/users/<your short user name>.conf`.

That file may not exist and if you upgrade from an older version, you may still not have it. If that file doesn't exist, you will need to create it with:

```
sudo vi /etc/apache2/users/<your short user name>.conf
```

Even if the file does exist, double check the content. Use the following as the content:

```
<Directory "/Users/<your short user name>/Sites/">
```

```
AddLanguage en .en
AddHandler cgi-script .cgi .pl .php
Options Indexes MultiViews FollowSymLinks ExecCGI
AllowOverride None
Require host localhost
</Directory>
```

You may need to make changes to this configuration to enable things such as additional languages, access from devices other than localhost, and .htaccess file overrides.

There were new security defaults in macOS 13 "Ventura". By default, other users have no access to another user's home directory. This includes the special "_www" user that is running the Apache web server. Run the following command to give the Apache web server access to the Sites folder in your home directory.

```
chmod +a "_www allow execute" ~
```

This will add an ACL permission to your home directory that will allow the Apache web server access to all subdirectories inside your home directory. The permissions on those subdirectories may allow, or deny, access to other users such as the _www user. If you wanted to be extra paranoid, you could change the permissions on all subdirectories (except for the "Sites" folder, obviously) to disallow any access from other users. But this User Tip is a minimal setup guide, so we will just do the minimum here.

Now you are ready to turn on Apache itself. But first, do a sanity check. Sometimes copying and pasting from an internet forum can insert invisible, invalid characters into config files. Check your configuration by running the following command in the Terminal:

```
apachectl configtest
```

If this command returns "Syntax OK" then you are ready to go. It may also print a warning saying "httpd: Could not reliably determine the server's fully qualified domain name". You could fix this by setting the ServerName directive in /etc/apache2/httpd.conf and adding a matching entry into /etc/hosts. But for a development server, you don't need to do anything. You can just ignore that warning. You can safely ignore other warnings too.

Turn on the Apache httpd service by running the following command in the Terminal:

```
sudo launchctl load -w /System/Library/LaunchDaemons/org.apache.httpd.plist
```

This command could fail with an error: "Load failed: 37: Operation already in progress". If so, that means your web server is already running from a previous operating system version. However, you will still need to bump Apache to reload with the configuration changes you've just made. Use the following command:

```
sudo apachectl graceful
```

In Safari, navigate to your web site by entering the following into Safari's address bar:

<http://localhost/>

It should say:

It works!

Now try your user home directory by entering the following into Safari's address bar:

<http://localhost/~<your short user name>>

It should say:

My site works

To test Perl, try something similar. Create a Perl test file with:

```
vi ~/Sites/info.pl
```

Use the following as the content:

```
#!/usr/bin/perl

use CGI;

my $cgi = CGI->new;

print $cgi->header( -type => 'text/plain' );

print $ENV{SERVER_SOFTWARE};
```

And test it by entering the following into Safari's address bar:

`http://localhost/~<your short user name>/info.pl`

Make sure the file can be executed by the `_www` user by doing this:

```
chmod ugo+x ~/Sites/info.pl
```

You should see the following string:

Apache/2.4.56 (Unix)

Doing the same for PHP is left as an exercise for the reader. You will have to download and install a PHP interpreter. For extra credit, try something more modern like Python or Rust.

If you want to setup MySQL, see [my User Tip on Installing MySQL](#).

If you want to make further changes to your Apache system or user config files, you will need to restart the Apache server with:

```
sudo apachectl graceful
```

Most important note: This configuration is only useful for testing HTML/CSS layouts and some Javascript. Please don't write any new websites using Perl or PHP. That kind of web architecture is going to be painful in the modern world. Instead, look for content management systems, static site builders, or even GitHub. If you want to deploy a real web site, you'll do that on AWS. You can even use AWS Lightsail to get started with a more traditional web hosting architecture. But eventually you'll want to transition to Cloud Front. AWS SSL certificates alone are worth the effort.

(67)

This site contains user submitted content, comments and opinions and is for informational purposes only. Apple may provide or recommend responses as a possible solution based on the information provided; every potential issue may involve several factors not detailed in the conversations captured in an electronic forum and Apple can therefore provide no guarantee as to the efficacy of any proposed solutions on the community forums. Apple disclaims any and all liability for the acts, omissions and conduct of any third parties in connection with or related to your use of the site. All postings and use of the content on this site are subject to the Apple Support Community Terms of Use. See how your data is managed...

[Support](#) [Community](#)

More ways to shop: Visit an [Apple Store](#), call 1-800-MY-APPLE, or [find a reseller](#).

Copyright © 2024 Apple Inc. All rights reserved. [Privacy Policy](#) | [Terms of Use](#) | [Sales and Refunds](#) | [Legal](#) | [Site Map](#)

United States