

# Set up Apache for local development/testing?

Asked 16 years, 3 months ago   Modified 3 years ago

Viewed 22k times    Part of [PHP](#) Collective



14



I've been impressed by the screencasts for Rails that demonstrate the built-in web server, and database to allow development and testing to occur on the local machine. How can I get an instance of Apache to execute a project directory as its DocumentRoot, and maybe serve up the files on port 8080 (or something similar)?

The reason why I'm asking is that I'm going to be trying out CodeIgniter, and I would like to use it for multiple projects. I would rather not clutter up my machine's DocumentRoot with each one. Suggestions on how to do database migrations are also welcome.

Thank you for your responses so far. I should clarify that I'm on Mac OS X. It looks like WAMP is Windows-only. Also, XAMPP looks like a great way to install Apache and many other web tools, but I don't see a way of loading up an instance to serve up a project directory. Mac OS X has both Apache and PHP installed - I'm just looking for a way to get it to serve up a project on a non-standard port.

I just found [MAMP Pro](#) which does what I want, but a more minimalist approach would be better if it's possible.

Does anyone have a `httpd.conf` file that can be edited and dropped into a project directory?

Also, sorry that I just threw in that database migration question. What I'm hoping to find is something that will enable me to push schema changes onto a live server without losing the existing data. I suspect that this is difficult and highly dependent on environmental factors.

PHP

php

codeigniter

apache

Share

Improve this question

Follow

edited Dec 8, 2021 at 8:17



Abdulla Nilam

38.5k ● 18 ● 68 ● 94

asked Sep 23, 2008 at 0:31



Kyle Cronin

79k ● 45 ● 151 ● 167

8 Answers

Sorted by:

Highest score (default)



28

Your Mac comes with both an Apache Web Server and a build of PHP. It's one of the big reasons the platform is well loved by web developers.



Since you're using Code Igniter, you'll want PHP 5, which is the default version of PHP shipped with 10.5. If you're on a previous version of the OS hop on over to [entropy.ch](http://entropy.ch) and install the provided PHP5 package.



Next, you'll want to turn Apache on. In the sharing preferences panel, turn on personal web sharing. This will start up apache on your local machine.

Next, you'll want to setup some fake development URLs to use for your sites. Long standing tradition was that we'd use the fake TLD `.dev` for this (ex. `stackoverflow.dev`). However, `.dev` is now an actual TLD so you probably don't want to do this -- `.localhost` seems like an emerging defacto standard. Edit your `/etc/hosts` file and add the following lines

```
127.0.0.1    www.example.localhost
127.0.0.1    example.localhost
```

This points the above URLs at your local machine. The last step is configuring apache. Specifically, enabling named virtual hosting, enabling PHP and setting up a few virtual hosts. If you used the entropy PHP package, enabling PHP will already be done. If not, you'll need to edit your `http.conf` file as described [here](#). Basically, you're uncommenting the lines that will load the PHP module.

Whenever you make a change to your apache config, you'll need to restart apache for the changes to take effect. At a terminal window, type the following command

```
sudo apachectl graceful
```

This will gracefully restart apache. If you've made a syntax error in the config file apache won't restart. You

can highlight config problems with

```
sudo apachectl configtest
```

So, with PHP enabled, you'll want to turn on `NamedVirtualHosts`. This will let apache respond to multiple URLs. Look for the following (or similar) line in your `http.conf` file and uncomment it.

```
#NameVirtualHost *
```

Finally, you'll need to tell apache where it should look for the files for your new virtual hosts. You can do so by adding the following to your `http.conf` file. NOTE: I find it's a good best practice to break out config rules like this into a separate file and use the `include` directive to include your changes. This will stop any automatic updates from wiping out your changes.

```
<VirtualHost *>
    DocumentRoot /Users/username/Sites/example.localho
    ServerName example.localhost
    ServerAlias www.example.localhost
</VirtualHost>
```

You can specify any folder as the `DocumentRoot`, but I find it convenient to use your personal `Sites` folder, as it's already been configured with the correct permissions to include files.

Follow

answered Sep 23, 2008 at 1:44



**Alana Storm**

166k ● 95 ● 415 ● 618

---

I think this is the best approach because you don't have to mess with DNS, worry about your dynamic IP address, or anything. You can run your local webserver and type in your 'dev' URLs just like they are real websites. – [JayTee](#) Sep 23, 2008 at 4:39

---

This is the simplest approach, and the one I recommend heartily as well :) – [Aeon](#) Sep 23, 2008 at 20:10

---

Great answer and still relevant almost a decade later. Thanks Alan. – [Leonard H. Martin](#) Nov 24, 2017 at 22:47

---

Please note that the [.dev TLD](#) is not fake anymore. Pick another one or make sure your rules dont interfere with any website you might want to visit. – [Morgan](#) Feb 27, 2019 at 16:19

- 
- 1 @flugga Thanks! As much as .dev being an actual thing now fills me with eye rolling, you're right. I've updated the question to reflect the current reality -- 10 years?! after the original answer. – [Alana Storm](#) Feb 27, 2019 at 16:52
- 



6



Sorry Kyle, I don't have enough cred to respond directly to your comment. But if you want to have each project be served on a different port, try setting up your virtual host config exactly like Kelly's above (minus the DNS stuff) except instead of 80, give each virtual host its own port



number, assuming that you've added this port to your `ports.conf` file.



```
NameVirtualHost *

<virtualhost *:80>
DocumentRoot /site1/documentroot
</virtualhost>

<virtualhost *:81>
DocumentRoot /site2/documentroot
</virtualhost>

<virtualhost *:82>
DocumentRoot /site3/documentroot
</virtualhost>

<virtualhost *:83>
DocumentRoot /site4/documentroot
</virtualhost>
```

Hope that helps

Share Improve this answer

Follow

edited Dec 8, 2021 at 8:26



[Abdulla Nilam](#)

38.5k ● 18 ● 68 ● 94

answered Sep 23, 2008 at 1:23



[Jurassic\\_C](#)

2,411 ● 3 ● 23 ● 19

---

Not exactly what I was hoping for, but it's good enough. I think I'll use it when my trial of MAMP Pro runs out. P.S. you should now have enough rep to comment :) – [Kyle Cronin](#)

Sep 23, 2008 at 1:36

---



2

I also download the latest binaries for each and set them up manually. I've found it to be a painless process, as long as you know a little bit about configuring Apache.



On my development machine, I have apache setup with [name-based virtual hosting](#). I also have a [dyndns.org](#)



account which maps my development domain to my local machine. [DynDNS](#) provides a wildcard domain and therefore using name-based virtual hosts I can easily create as many sites (with separate document roots) for as many development domains as I want, all running off the one Apache instance.



e.g. Apache config for the virtual hosts might be something like

```
NameVirtualHost *:80

<virtualhost *:80>
ServerName site1.mydyndns.dyndns.org
DocumentRoot /site1/documentroot
</virtualhost>

<virtualhost *:80>
ServerName site2.mydyndns.dyndns.org
DocumentRoot /site2/documentroot
</virtualhost>
```

This has been the quickest and easiest way for me to easily maintain many development sites on my local machine.

I hope that makes sense.

Share Improve this answer

edited Dec 8, 2021 at 8:28

Follow



Abdulla Nilam

38.5k ● 18 ● 68 ● 94

answered Sep 23, 2008 at 0:59



Bazman

1,229 ● 9 ● 10

---

That's a good idea and does make sense; however, my development machine is a laptop and switches IPs too often for DNS to keep up. However, it might be possible to set up my own DNS server on my machine and use that instead.

– [Kyle Cronin](#) Sep 23, 2008 at 1:02

---



0

You could use a low up front setup package such as [XAMPP](#) and run it as a separate instance. There are many other similar projects as well.



Share Improve this answer

answered Sep 23, 2008 at 0:38

Follow



phloopy

5,583 ● 4 ● 29 ● 37



0

For PHP you have several high-quality packages for deploying Apache+Mysql+PHP, such as [WAMP](#) and [XAMPP](#). Personally, I download the latest binaries of each and install manually to have more fine grained control.



There are plenty of online tutorials on how to handle that.







Database migrations should be straightforward - dump the database on the server, either at the command line or through an interface such as [PHPMyAdmin](#), and export it back in similar ways (PHPMyAdmin is recommended if you have no experience with the Mysql command line).

Share Improve this answer

answered Sep 23, 2008 at 0:42

Follow



[Eran Galperin](#)

86.8k ● 24 ● 118 ● 132



0

You can use MAMP pro but the free version is a very good choice as well. Get it here:

<http://www.mamp.info/en/mamp.html>



Share Improve this answer

answered Sep 23, 2008 at 2:24

Follow



[rg88](#)

21k ● 18 ● 79 ● 110



0

I don't use macOS, but I do use Apache. In my apache configuration file (on Linux it's usually at `/etc/apache2/apache2.conf`), look for a reference to a file called `ports.conf`. Find this file and add the line

```
Listen 8080
```



Then restart the apache process. After that, you should be in business. I apologize in advance if this doesn't work on a mac :)

Share Improve this answer

Follow

edited Dec 8, 2021 at 8:28



Abdulla Nilam

38.5k ● 18 ● 68 ● 94

answered Sep 23, 2008 at 1:07



Jurassic\_C

2,411 ● 3 ● 23 ● 19

---

That would work on Mac OS X as well, but it would only change the port that the default httpd process serves at. My goal is to have a separate port for each project. I suspect that will involve a separate httpd.conf for each project as well.

– [Kyle Cronin](#) Sep 23, 2008 at 1:09

---



0

I might recommend using a separate LAMP virtual appliance for each development environment you wish to experiment with.



Run them on VMware Server or VirtualBox.



Share Improve this answer

Follow



edited Dec 8, 2021 at 8:28



Abdulla Nilam

38.5k ● 18 ● 68 ● 94

answered Sep 23, 2008 at 16:25



EmmEff

7,921 ● 2 ● 19 ● 19

---