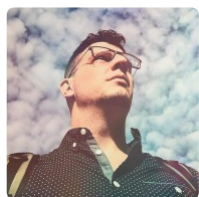Shop   Books   Blog   Contact

≡
MENU

REQUEST A QUOTE

1-855-WEBDEV-0

How to Set Up SSL with OSX Mountain Lion's Built-In Apache

May 24, 2013
by Justin Sternberg

There are many tutorials out there, but for whatever reason I couldn't get it work after a first pass. After a little help from some Twitter friends (thanks @ryandonsullivan & @andygambles!), I was able to pull together all the necessary pieces (read: remove gremlins from my setup including but not limited to typos) and get it working.

This will be very thorough, and (I hope) won't take any knowledge for granted.

References:

- Using SSL for your local development
- Apache SSL on Mac OSX Lion 10.7
- How to create a self-signed SSL Certificate

Steps:

1. Finder App -> Go -> Go to Folder (cmd-shift-g) -> type:

   ```
   /private/etc/apache2
   ```

   -> hit enter.

2. New folder (cmd-shift-n) (OSX may ask for your admin password) -> name the folder '**ssl**' -> hit enter
3. Create a backup of the '**httpd.conf**' file in that folder.
4. Open the '**httpd.conf**' file (in the apache2 folder that you're in) in a text editor. (next several steps taken from here)

   1. Make sure the SSL module is enabled if it's not. Do this by uncommenting (aka, remove the '#' symbol in front) the line that looks like:

      ```
      LoadModule ssl_module libexec/apache2/mod_ssl.so
      ```

   2. In the same file search for and uncomment (remove the #) this line:

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

   3. Save the file. (OSX may ask for your admin password)

5. Create a backup of the '**httpd-ssl.conf**' file (in the '**apache2/extra**' folder).
6. Now open the '**httpd-ssl.conf**' file in your text editor.

   1. Update the '**DocumentRoot**' line with where your site's root folder. In my case I put the path to my WordPress installation I needed to test. Something like:

      ```
      DocumentRoot /Users/YOURUSERNAME/Sites/WordPress
      ```

   2. Make sure '**ServerName**' line looks like:

      ```
      ServerName localhost
      ```

3. modify '**ServerAdmin**' line to use one of your email addresses. (I doubt this is necessary)

4. Take note of the line '**ErrorLog**'. You can use the location listed there to "Go to folder" in finder and view the log if things don't work. Or worse-case scenario, you can provide it's output to someone smart who might be able to help. :)

5. Edit '**SSLCertificateFile**' line to point to your new ssl folder and the ssl certificate file (don't worry, we'll be creating that shortly). That line should look like:

```
SSLCertificateFile "/private/etc/apache2/ssl/ssl.crt"
```

6. Do the same with the '**SSLCertificateKeyFile**' line:

```
SSLCertificateKeyFile "/private/etc/apache2/ssl/ssl.key
```

7. In the same file comment out (add a # to the beginning of the line) the '**SSLCACertificatePath**' and '**SSLCARevocationPath**' lines if they're not already.

8. Save the file. (OSX may ask for your admin password)

7. Open Terminal app -> Read below caveats, then follow instructions here

1. Each step has a singular line of dark bold text.. that is text you will paste into terminal and hit enter (copy the text exactly, not including white space or line breaks). The blue text below is an example of what terminal will output. (After pasting the line from step 2, terminal will walk you through the certificate's necessary info. Whatever you put should be fine.. just hit enter after answering)

2. On step 5, replace '**/usr/local/apache/conf/ssl.crt**' with the path to the location of the new 'ssl' folder you created in finder: **/private/etc/apache2**. So this will look like:

```
/private/etc/apache2/ssl/ssl.crt
```

Do the same with the ssl.key line. Replace:

```
/usr/local/apache/conf/ssl.key
```

with:

```
/private/etc/apache2/ssl/ssl.key
```

If you haven't noticed, these values need to line up exactly with the paths you put in the '**httpd-ssl.conf**' file.

3. Skip step 6 (it's already done), and step 7.

8. At this point, you should be able to go to https://localhost and see it working. If you're using a good browser, you'll probably be warned about the authenticity of the certificate, but unless you have trust issues with yourself, you can ignore the errors. You'll probably also notice other indications that you're not using a known certificate, but again, you don't need to worry about them.

Chrome's
notification

*Note: from Using SSL for your local development, Beware that Apache only supports one SSL domain per IP. So you need to add further IP's to your system, or use a module that allows multiple SSL VirtualHosts per IP. There is some further reading on this topic here.

*Another Note: According to Steph in the comments, "for the new OSX we have to uncomment `LoadModule socache_shmcb_module modules/mod_socache_shmcb.so` in file httpd.conf".

Share this:

          G+      
  3

Posted in **DEVELOPMENT**   **TUTORIAL** and tagged with   **APACHE**   **HTTPS**   **LOCAL DEVELOPMENT**   **LOCALHOST**   **MOUNTAIN LION**   **OSX**   **SSL**

[« PREVIOUS POST]   [NEXT POST »]

15 thoughts on

How to Set Up SSL with OSX Mountain Lion's Built-In Apache

Pingback: Configuring a Local Apache/PHP/MySQL Dev Environment in OS X – Ramblings From a Nobody – Personal site of Brian Richards

Sabine says:

July 11, 2013 at 10:07 am

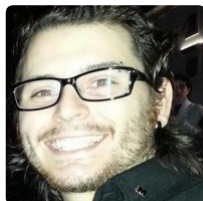Excellent article! Thank you so much, saved me a lot of time. Where is your Donate button?!?!

April Williams says:

July 12, 2013 at 10:31 am

Glad we could help!

Andres Monge says:

November 13, 2013 at 9:47 am

Thank you very much!

Christian says:

December 4, 2013 at 8:43 pm

Perfect.
Clear instructions, and it worked :)

Nitin says:

April 22, 2015 at 2:37 pm

This is THE best and THE best guide. Worked flawlessly and so so so simple to understand. Thanks sooo much for this :)

jithin says:

September 3, 2015 at 5:54 am

Super good and clear.Closest thing to an ideal guide!!

Brandon Reynolds says:

September 14, 2015 at 12:43 pm

Regular apache works but I get this error when trying to connect to the SSL site.

AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using Brandons-MacBook-Pro.local. Set the 'ServerName' directive globally to suppress this message

Steph says:

September 28, 2015 at 11:47 am

for the new OSX we have to uncomment : LoadModule socache_shmcb_module modules/mod_socache_shmcb.so in file httpd.conf

Rene says:

October 6, 2015 at 6:14 pm

Hi, thank you for the time you took to write this guide, so far I've followed everything in it, but after everything it's done, I'm still unable to connect into my https://localhost. I'm working on "El Capitan", and already uncomment the line Steph mentioned; so far I'm receiving a "ERR_CONNECTION_REFUSED". I would really appreciate if someone could help me. I'm pretty much a beginner. Thanks in advance for your time to read this.

Rene says:

October 6, 2015 at 6:46 pm

I've just redo it once again and it's finally working. I really appreciate all the efforts you people do to make this guides.

Gordy says:

October 22, 2015 at 4:45 pm

Superb. Thanks for that. Steph's comment was also useful (well, required).

Cheers.

Stanislav says:

October 28, 2015 at 7:13 pm

Hi, Thank You for your simple and clear instructions!

In my case after all steps I had to run in terminal:

sudo apachectl restart

Kellan says:

December 12, 2015 at 11:33 am

Just a heads up–In httpd.conf, I had to enable

LoadModule socache_shmcb_module
libexec/apache2/mod_socache_shmcb.so
LoadModule socache_dbm_module
libexec/apache2/mod_socache_dbm.so

Craig R Morton says:

July 7, 2016 at 5:56 am

Thanks, this worked great for me on El Capitan. I did have to enable
this though http://stackoverflow.com/a/34729432/691505

Have a comment?

Your email address will not be published.
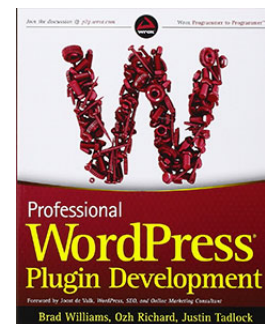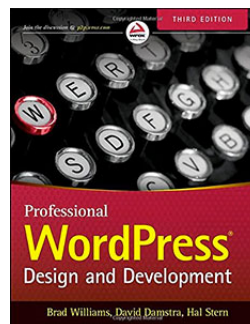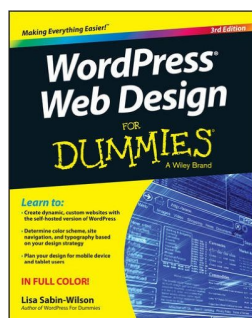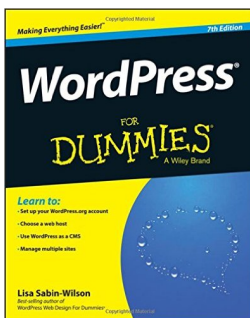
Be friendly, and let's talk.

Name (required)

Email (required)

Website (optional)

POST COMMENT

☐ Notify me of follow-up comments by email.

☐ Notify me of new posts by email.

WebDevStudios wrote the books on WordPress

VIEW OUR BOOKS ▶

We Build Great Things with Great Clients

VIEW OUR WORK ▶   Or, become a client

project scoping and management, all the way through development, to knowledge transfer and support, WebDevStudios delivered excellent service! Their team has excellent technical skills and solid communications practices. They never missed a deadline. WDS made our vision to create the INPUT iCenter a reality.

k you t
project was
were alread
group of pe

Randi Haran // Information Services History Company

Latest from the WDS Blog

DESIGN VS. ART: WHAT'S THE DIFFERENCE AND DOES IT MATTER?

Design is a solution to a problem. Art is a question to a problem. —John Maeda Have you ever called your designer "a creative," or asked them to just "use their imagination" when discussing new designs? Maybe you even came to them with a solution to a …

READ MORE

A CHANGE IN FIVE FOR THE FUTURE AT WEBDEVSTUDIOS

For almost two years now, WebDevStudios has been participating in the Five for the Future movement. During that time we have contributed thousands of hours to the WordPress community across various projects. Contributing to the WordPress community, and open source in general, is at the …

READ MORE

CONTACT US TODAY   1-855-WEBDEV-0

Proudly powered by WordPress. Hosted by WP Engine.
WebDevStudios is a **WD3** Party