Watch folder/files change with bash script

No dependencies!

By **Ibrahim Diallo**

Published Jun 15 2016 ~ 5 minutes read

One of the greatest thing about web development is instant gratification. If you make any changes to your HTML, CSS, or JavaScript all you have to do is refresh the browser page and your new changes will show up. There is no time wasted compiling the code. However, recently I have been working on different projects and some of them make use of SASS or Less CSS. These are programming tools that compile into CSS. The main benefit is that you can use it modularize your css but the price is that you have to compile the CSS every time you make change.

There are multiple tools that you can use to automatically compile when you save a file. One that I have used in the past was grunt. However, for this project I didn't want to install nodejs and a bunch of other dependencies just so I can watch a folder.

Here is the command to compile the CSS with SASS:

```
> scss input.scss output.css
```

So every time I make a change in a `.scss` file, I have to run this command. Because I also have to create a minified version of the css, I have to run this command too:

```
> scss input.scss output.min.css --style compressed;
```

```
# ./compilesass.sh
scss input.scss output.css
scss input.scss output.min.css --style compressed;
```

This makes me slightly happier but I still have to run the command manually every single time. The solution is to create a batch script that will watch all the files in my `/scss` folder and compile the moment it detects any change. So I created `sasswatch.sh`.

I identified the path of my folders and first hard-coded them in the script:

```
# Main files and folders
cssfolder='public/assets/css'
scssfolder="$cssfolder/scss"
sourcecss="$scssfolder/main.scss"
targetcss="$cssfolder/main.css"
targetcssmin="$cssfolder/main.min.css"
```

It is easy to upgrade this script to take parameters instead of hard coding the values. For example you can assign a 'source', 'target', 'watchfolder' and pass it to the script:

```
> ./sasswatch.sh public/assets/css/scss/main.scss public/assets/css/main.css p
            $0                  $1                                $2
```

Just replace it with the dollar sign values on your script.

Alright back to watching the folder.

A simple way to detect if the files have change is to get a list of all of them with the last modified date.

```
> find $scssfolder -type f -printf "%T@ %p\n"
```

Running this command will give us the last modified timestamps in milliseconds and the name of each file.

```
1465925599.3579026000 public/assets/css/scss/base/_colors.scss
1465926289.1712187000 public/assets/css/scss/base/_fonts.scss
```

```
1466014639.5718180000 public/assets/css/scss/layouts/_about-responsive.scss
1466017941.9916993000 public/assets/css/scss/layouts/_about.scss
1465926289.2073878000 public/assets/css/scss/layouts/_footer.scss
1465926345.8030610000 public/assets/css/scss/main.scss
1466013665.5203733000 public/assets/css/scss/_bundled.scss
```

Now we can take all this information and hash it with md5. In fact we can combine finding the files and md5ing it.

```
> find $scssfolder -type f -printf "%T@ %p\n" | md5sum | cut -d " " -f 1
c6c57757085183a5d081dedcd401d178
```

Now we have a unique string that represent the current files and their last modified date. Note that I added the command `cut` to remove everything after the space. We can save this hash, run the script at a time interval and if the hash changes, we will run our sass commands. Just like that we have all the components to build our script.

Here is the full code:

```
# ./sasswatch.sh

# colors
GREEN='\033[0;32m'
NC='\033[0m'

# run every interval in seconds
timeinterval=2;

# Main files and folders
cssfolder='public/assets/css'
scssfolder="$cssfolder/scss"
sourcecss="$scssfolder/main.scss"
targetcss="$cssfolder/main.css"
targetcssmin="$cssfolder/main.min.css"

echo "Watching SCSS files for changes"
echo "Folder=\"$scssfolder\""

chksum1=""
while [[ true ]]; do
    chksum2=`find $scssfolder -type f -printf "%T@ %p\n" | md5sum | cut -d " "
    if [[ $chksum1 != $chksum2 ]] ; then
        printf "Compiling Sass CSS\n";
        sass $sourcecss $targetcss;
        sass $sourcecss $targetcssmin --style compressed;
        printf "${GREEN}CSS compilation complete.${NC}\n";
```

I've added the script to get the hash of the file list and last modified timestamps inside a while loop that runs every 2 seconds. I feel 2 seconds is seamless enough to seem like your code compiles every time your save.

Conclusion

You can always download some tools to do the task for you. But the problem is you will bloat your code and add some unnecessary dependencies. Instead a simple bash script that is around 30 lines of code can do the job.

Happy Code Watching!

Did you like this article? You can buy me a coffee.
Share your insightful comments here.

Sign up for the Newsletter.

Name:

Enter your name

Email:

Enter your email

Subscribe

Follow me on Twitter, RSS Feed

Previous: Installing software: Ruby

Next: What is success?

On a related note, here are some interesting articles.

How to build a website from scratch series

In this series, we will start from the very

No more external CSS

One day I woke up and decided to remove all external CSS from my websites. And every website I build

Removing External JavaScript

My goal was to remove all external

View all articles

Comments

There are no comments added yet.

Let's hear your thoughts

```
Enter your insightful comments here ...
```

Your Name (Required)

Your Email (Required) For my eyes only

Your Website

Would you like to sign up to the news letter? □ **← Click here**

Post Comment

First of all, Wow! You scrolled all the way down. That means you want to know more about me. Well, here is a summary of who I am and what I do. I started this blog because ... wait I have a link for that too.

Hey! Have you heard about humans.txt? Well I kinda liked the idea and did my own version.

You can find me on:

Twitter YouTube Soundcloud

Don't hesitate to say hi, it's what keeps me going :)

Designed by Yours truly.