Moirai Web Builder

Video website generator

Contents

Intro

Requirements

Build Files Locally

Remote Web Building (XBMC) + Generate SSH keys

Subtitles

Tips and tricks!

Help with development?

Intro

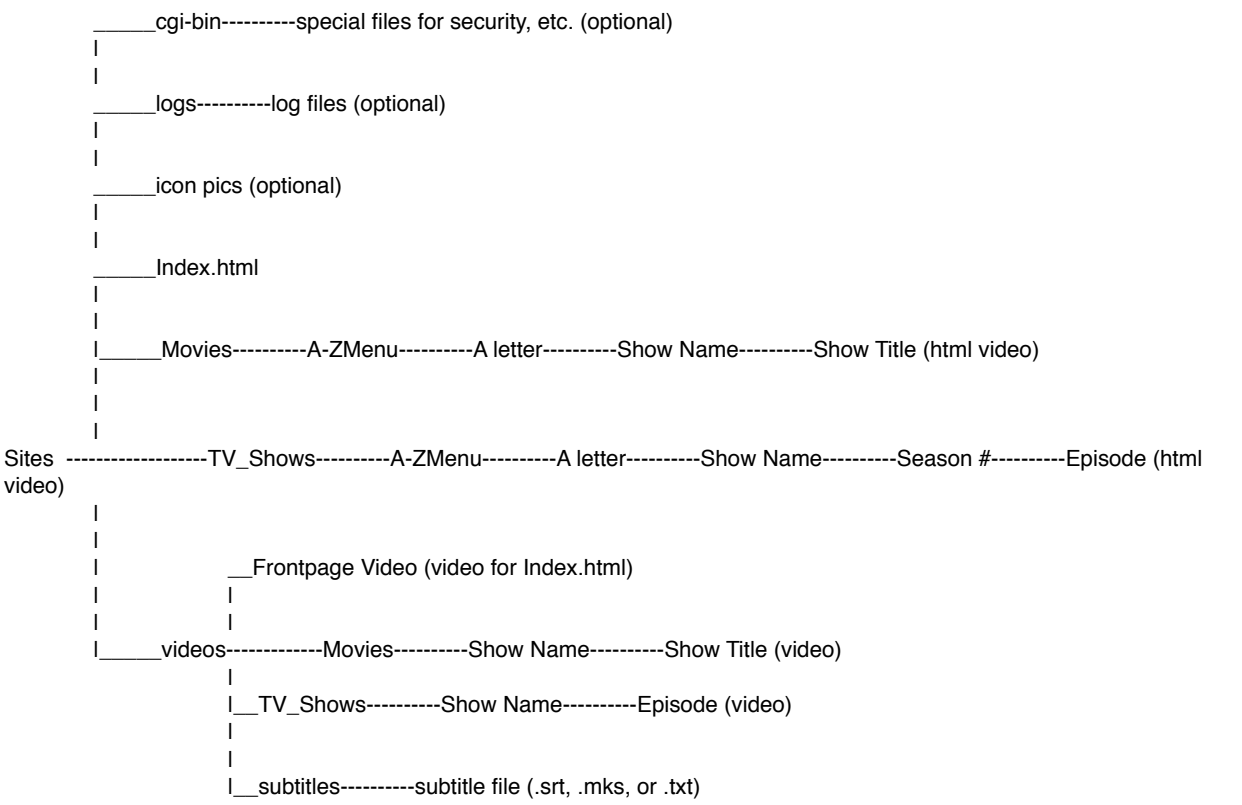
The Moirai Web Builder is an application designed to make video watching simple and fun! This scans video files and then builds directories and HTML5 web pages that are required for a basic video website. A user can choose to either build files on their local computer (the one in front of them) or on a remote computer (like a server at home). The application is named after the Greek Moirai Clotho, Lachesis, and Atropos who spun, measured, and cut the threads of life for humans (this app kind of does the same thing but for HTML5 files). Please read through all of the instructions before trying one of the methods for building your website.

To run any of the scripts press "⌘ + r" or click the green run button near the top left: 

Requirements

- The application requires all scripts to be run on an OS X computer. However, the remote web builder can build files for an OS X, Linux, *BSD, or Windows computer
- For the remote builder, the multimedia application XBMC is necessary on the server and the host computer (should be changed in the future so only the server needs XBMC).
- The remote builder also requires the user to set up an SSH keygen for their connection to their server prior to running the remote update script (check the remote builder section below for more details).
- Movies must have the title as their name with a file extension after that (i.e. “Foo.mkv” for the movie Foo)
- TV Shows must have the format: the show title, period, season number (s #), period, episode number (e #), period, file extension. For example: “Foo.s02.e05.webm” for episode five of season two of Foo, “Bar.s1.e14.mov” for episode fourteen of season one of Bar

- Video files must be ".mp4", ".webm", or ".ogg". If you're not sure about whether a video will work, try it out and see what happens.
- Similar to the video files, all subtitle files must follow a specific naming format: “video file name” + “language name” + “name extension”. For a movie Foo.English.srt would tell the computer to use English subtitles for the movie Foo. Alternatively, Bar.s1.e14.Macedonian.txt would mean display Macedonian subtitles for episode fourteen of season one of Bar. Use the language chart below for all the different language names. Make sure to drop all the SRT files in a folder named subtitles.
- Viewing the website requires one of these browsers: Safari, Google Chrome, or Chromium. Opera will not work (sorry!), Firefox acts a little glitchy with the videos, and I'm not sure about how Internet Explorer behaves (please switch your browser if your still using IE, it's terrible).
- Subtitles can only have .srt name extensions and the subtitles can only be viewed with Safari web browser (more support will come when webvtt subtitles are adopted into HTML5, sorry)
- Use this folder structure for your website as a starting point (I highly suggest doing this):



As you can see the videos section is where you should store the actual video files and the directory structure complies with the standard XBMC format (you'll have to try switching your format if you're not currently using this, sorry). I suggest using the sites folder in your home directory on OS X for this. In OS X the corresponding httpd.conf file that controls the apache2 site is located at /etc/apache2/httpd.conf (it's optional to change the config file but I think it's helpful to test on OS X before going to Linux, *BSD, or Windows). You will also probably have to edit the videos > files section in XBMC to point the application to your new videos.

Build Files Locally

This option allows the user to build all their files on their local computer. To run, double-click the Local install script and answer the prompts that are displayed. The application will then erase old web pages and folders and build either the TV or movie section of the website automatically in the specified location. After that, just repeat the same process to build more sections or update the existing web pages.

Build Files Remotely

Doing the remote build is a little more complicated than the local build but it's helpful if you don't have an OS X server or want to update your computer while you're out of the house. Double-click the install script and provide it with your information to create a customized remote-update script. The update script uses password prompts for all SSH connections and you'll have about fifteen seconds to type in each password. If you'd like to skip these password prompts, then complete the SSH key pair step (look below, I highly recommend using the key pair method). The newly built update script will build sections of your website for all the TV shows and movies that you currently have stored in your XBMC video library so your website is in sync with XBMC. If there are new videos in XBMC, new website sections will be made. Similarly, if there are old videos that were deleted from XBMC but are still on your website, then these deprecated parts of your website will be deleted. Don't forget to assign your server a static ip address too if you're not going to leave it on 24/7.

Generating an SSH Key Pair

To set up key-based SSH, you must generate the keys the two computers will use to establish and validate the identity of each other. To do this run, the following commands in Terminal.

1. Check to see whether a .ssh folder exists in your home directory by running the command ls -a. If .ssh is listed in the output, move to step 3. If .ssh is not listed in the output, run mkdir ~/.ssh; chmod 0700 .ssh and continue to step 2.
2. Run: ssh-keygen

   After entering this command, the computer should ask where to save the keys. The default location is usually a good location so hit return after the first prompt. It should look like this:

   Generating public/private rsa key pair.

   Enter file in which to save the key (/home/your_user_name/.ssh/id_rsa):

   Next, the computer will ask for a passphrase so just enter whatever passphrase you'd like to choose. Remember, this is an important passphrase so make it long and complicated.
3. Run: ssh serverUsername@0.0.0.0

   Make sure to SSH into the server at least once so that location is recorded in the known_hosts file on the local machine.
4. Run: scp .ssh/id_rsa.pub serverUsername@0.0.0.0:.ssh/authorized_keys2

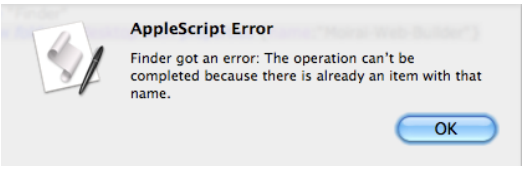
   Substitute serverUsername for the actual username on your sever and 0.0.0.0 for the actual ip address of the server. If you want to find your ip address run the command netstat and look for the address near the top of the data

(or you could try googling "what's my ip address?").

5. If a Finder window pops up and asks for your ssh passphrase, then type in the passphrase for the local computer (that was the one set for ssh-keygen in step 2).

Finder error

There is one error that seems to pop up quite a bit when using the remote build script. The script may report an error that looks like this:



If this ever comes up, try going to Users/username/library/application support/Moirai_web_builder/Sites and deleting everything in the Sites folder. The script should work well again after that.

Remote Build for Windows

If you're using a Windows server the remote build will work a bit differently because it will use Samba filesharing instead of SSH and rsync. When the remote build script runs, it will mount the drive from your Windows computer and ask for your username and password. If you'd like to automate the build script so you never see this password prompt again, try checking the box next to "Remember this password in my keychain".

Let me know here if application still has glitches for Windows and I'll try to fix everything right away (I don't have a Windows computer so I still haven't tested this too much :p).

Subtitles

Subtitles for videos can be displayed as long as you have a subtitle file with a proper title. A .srt, .mks, or .txt subtitle file is basically just a text file that lists the subtitle text and the times the text should be displayed in the video. There are many sites which offer free SRT downloads so try googling something like "show title, language, srt". Here's a list of languages that are currently supported (image taken from VideoJS website):

ab	Abkhazian	fr	French	lv	Latvian (Lettish)	si	Sinhalese
aa	Afar	fy	Frisian	li	Limburgish (Limburger)	ss	Siswati
af	Afrikaans	gl	Galician	ln	Lingala	sk	Slovak
sq	Albanian	gd	Gaelic (Scottish)	lt	Lithuanian	sl	Slovenian
am	Amharic	gv	Gaelic (Manx)	mk	Macedonian	so	Somali
ar	Arabic	ka	Georgian	mg	Malagasy	es	Spanish
an	Aragonese	de	German	ms	Malay	su	Sundanese
hy	Armenian	el	Greek	ml	Malayalam	sw	Swahili (Kiswahili)
as	Assamese	kl	Greenlandic	mt	Maltese	sv	Swedish
ay	Aymara	gn	Guarani	mi	Maori	tl	Tagalog
az	Azerbaijani	gu	Gujarati	mr	Marathi	tg	Tajik
ba	Bashkir	ht	Haitian Creole	mo	Moldavian	ta	Tamil
eu	Basque	ha	Hausa	mn	Mongolian	tt	Tatar
bn	Bengali (Bangla)	he	Hebrew	na	Nauru	te	Telugu
dz	Bhutani	hi	Hindi	ne	Nepali	th	Thai
bh	Bihari	hu	Hungarian	no	Norwegian	bo	Tibetan
bi	Bislama	is	Icelandic	oc	Occitan	ti	Tigrinya
br	Breton	io	Ido	or	Oriya	to	Tonga
bg	Bulgarian	id	Indonesian	om	Oromo (Afan, Galla)	ts	Tsonga
my	Burmese	ia	Interlingua	ps	Pashto (Pushto)	tr	Turkish
be	Byelorussian (Belarusian)	ie	Interlingue	pl	Polish	tk	Turkmen
km	Cambodian	iu	Inuktitut	pt	Portuguese	tw	Twi
ca	Catalan	ik	Inupiak	pa	Punjabi	ug	Uighur
zh	Chinese (Simplified)	ga	Irish	qu	Quechua	uk	Ukrainian
zh	Chinese (Traditional)	it	Italian	rm	Rhaeto-Romance	ur	Urdu
co	Corsican	ja	Japanese	ro	Romanian	uz	Uzbek
hr	Croatian	jv	Javanese	ru	Russian	vi	Vietnamese
cs	Czech	kn	Kannada	sm	Samoan	vo	Volapük
da	Danish	ks	Kashmiri	sg	Sangro	wa	Wallon
nl	Dutch	kk	Kazakh	sa	Sanskrit	cy	Welsh
en	English	rw	Kinyarwanda (Ruanda)	sr	Serbian	wo	Wolof
eo	Esperanto	ky	Kirghiz	sh	Serbo-Croatian	xh	Xhosa
et	Estonian	rn	Kirundi (Rundi)	st	Sesotho	yi	Yiddish
fo	Faeroese	ko	Korean	tn	Setswana	yo	Yoruba
fa	Farsi	ku	Kurdish	sn	Shona	zu	Zulu
fj	Fiji	lo	Laothian	ii	Sichuan Yi		
fi	Finnish	la	Latin	sd	Sindhi		

The subtitles are enabled through the VideoJS HTML5 Video Player (look here for more info). As you can see, this supports a bunch of languages (144 total) so you're free to watch all you new shows with Javanese, Nauru, or Limburgish subtitles.

The greatest drawback to this method is that you'll need Safari to view the subtitles. If you're using Linux or *BSD you may have trouble getting Safari installed. I don't really know what to suggest (I'm sorry). Hopefully subtitle support will be adopted into HTML5 and all the major web browsers soon but until then I think VideoJS is a decent solution. Let me know if you have other ideas for a subtitle player that's more universal.

Tips and Tricks

To help anyone that's never run their own website before, here are a few ideas that will hopefully help keep your website safe and powerful!

Create an Index.html page automatically

I included a script for creating a nice looking index.html page for your website on your local computer. Just point it to the location of your front page video file and tell it where to build the HTML file. If you don't know what the index.html file is, don't worry. It's basically just the front page of the website. Make sure to keep the front page video file in the videos directory or there will be no video for the web page to display.

If you're using the remote script, the index.html will build automatically. Remember to keep your actual front-page video file in the videos folder of the Apache directory.

Use custom error pages

Error pages are a pretty common feature for websites and they're pretty easy to set up. If you want to make your own, just make an HTML file with the correct error code (check out the 404.html I included) and edit your httpd.conf file to point to this file. Check out this tutorial on how to do this:

http://httpd.apache.org/docs/2.0/custom-error.html

A huge thanks goes out to the HTML5 boilerplate team for their neat 404.html page. This webpage has an "Unlicense" attached to it so it was provided as public domain. If you want more ideas for cool HTML5 things, check out the boilerplate project!

Assign a static ip address

For the remote update script to work well you should have a static ip address on your server (otherwise you'll need to make a new remote update script each time the ip address changes). Here's how to get a static ip address on OSX and many Linux distributions:

On OSX try going to System Preferences >> Networking >> Advanced, then click the TCP/IP tab. Highlight the current line of IPv4 Address and copy that. Then switch the Configure IPv4 line to Using DHCP with manual address. Finally, just paste the old ip address into the IPv4 address line.

If your server is *BSD or Linux (this may not work on all distros) then enter this into your /etc/rc.conf using the actual ip address you'd like instead of zeros:

```
ifconfig_em0="inet 0.0.0.0 netmask 255.255.255.0."
defaultrouter="0.0.0.0"
```

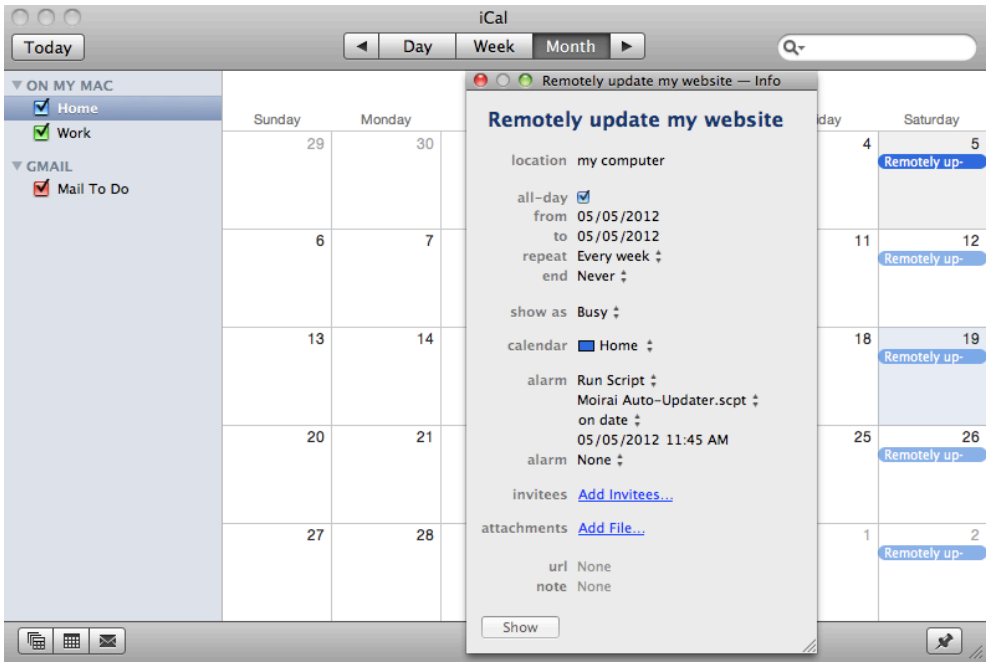
For a Windows server, try following these instructions to get a static ip address:

http://www.howtogeek.com/howto/19249/how-to-assign-a-static-ip-address-in-xp-vista-or-windows-7/

Automatically run an auto-updater script without a Cron

Ok, this title is pretty ridiculous but the geniuses that put together OS X included a handy resource for automating script running with iCal. If you're not familiar with Crons, this will basically just run the given script whenever at the designated times (your computer should probably be on through).

1. Open iCal and create an event
2. Set an alarm with parameters like this:



Password protecting your website!

This section is entirely optional because chances are that nobody except you will ever log onto your website but here are some ideas just incase.

http://httpd.apache.org/docs/2.0/howto/auth.html

Giving a static IP address to your computers and then issuing an *Allow from address* command under *Deny from all* in the httpd.conf seems like a good security measure if you'll only use your own computers to access your website (look at the bottom of the linked web page for how to do this). Just password protecting your website is also good but the password can still be intercepted by a packet sniffer. If you need to use computers other than just your own, try setting up a SSL/TLS encrypted connection. This will encrypt all of the information as it passes between the server and a user on a remote computer. Complete the password setup described above and then try creating SSL/TSL with this:

http://httpd.apache.org/docs/2.0/ssl/

Hopefully people just mind their own business and you don't have to worry about anyone messing with your stuff (your site is probably pretty difficult to find anyway). If you're still worried, you could set up logs to get a synopsis of who is connecting and when. This may help ease your fears about any hacking attempts on your site. Check out how to set up logs here:

http://httpd.apache.org/docs/2.0/logs.html

Play your videos on an iPad, iPhone, and maybe other mobile devices

Woohoo! This is a nice part of having a personal cloud video player. You should be able to watch any of your videos whenever you like as long as they're in the proper video format. Here's how to convert a video using Quicktime on an OS X computer:

1. Open your video with Quicktime
2. Hit: *Shift* + ⌘+E

3. Select the appropriate export type and destination

For batch conversions try using a batch converter for videos (I suggest trying Media Converter). This should be able to convert all the videos within a folder with a simple drag and drop. The HTML code I wrote should support iPads and iPhones but I'm not 100% sure and I have no way of testing because I don't have any kind of mobile device to play videos on. If you try watching videos on a mobile device please let me know how it goes!

Sync any files with your server anytime

* If you have a normal, residential internet account you may want to skip doing this because your ISP (internet service provider) could spas out.*

I put together a package with instructions for how to set up a cloud computer. It uses Samba mounted over WAN through an encryption tunnel, has a VM for public computer use, and can establish a VPN or proxy with sshuttle. This works best with a low-power ARM computer like the CuBox, which only uses ~2-3 watts of power.

Check it out here: http://code.google.com/p/cloud-computer-package

Help with development?

If you have any ideas for what could help make this application better feel free to suggest them in the development group or in the XBMC forums (they don't even need to be in code, articulating awesome ideas in plain English is also appreciated). If you really like this application, make sure to spread the joy by telling other people about it!

Hope you enjoy :-)