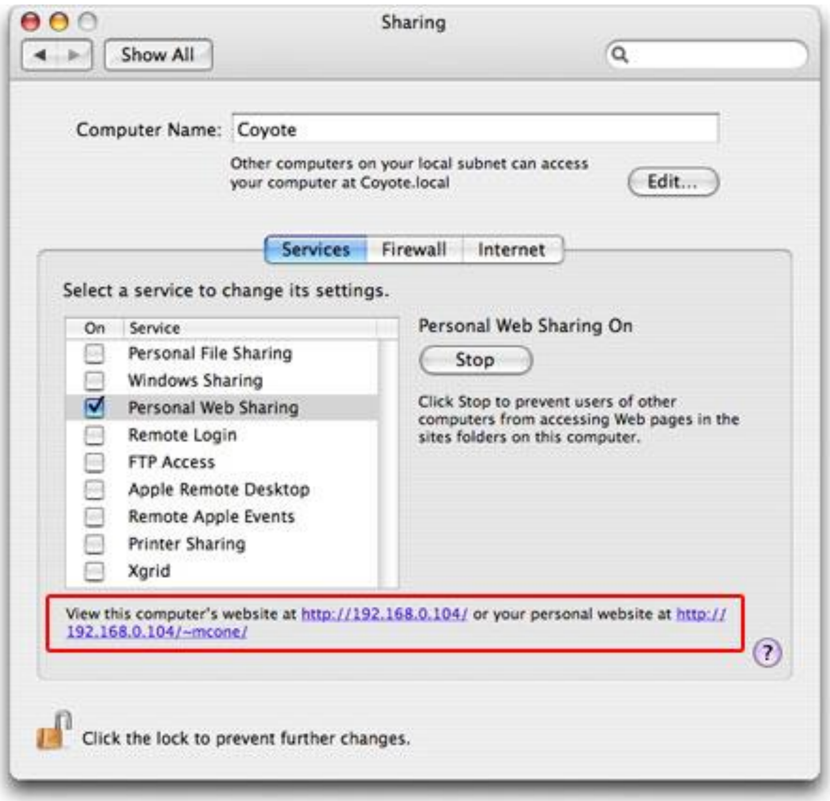
# How to Turn Your Mac into a Web Server

Mac OS X is built on Darwin -- a Unix-like, open source operating system developed by Apple and built on FreeBSD. This means that Mac users have access to free built-in server applications, like the Apache web server.

System Preferences – Services



Computer Name: Coyote

Other computers on your local subnet can access your computer at Coyote.local

Select a service to change its settings.

| On | Service |
|---|---|
| ☐ | Personal File Sharing |
| ☐ | Windows Sharing |
| ✓ | Personal Web Sharing |
| ☐ | Remote Login |
| ☐ | FTP Access |
| ☐ | Apple Remote Desktop |
| ☐ | Remote Apple Events |
| ☐ | Printer Sharing |
| ☐ | Xgrid |

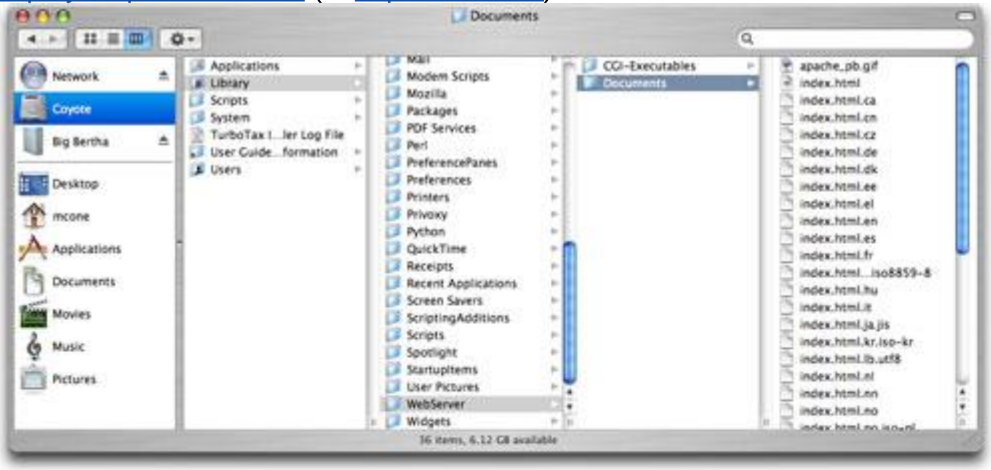
Personal Web Sharing On

Click Stop to prevent users of other computers from accessing Web pages in the sites folders on this computer.

View this computer's website at http://192.168.0.104/ or your personal website at http://192.168.0.104/~mcone/

Save your files into one of the following directories:

- **Parent Directory:** Library > WebServer > Documents
  [http://youripaddresshere/](http://youripaddresshere/) (or [http://localhost/](http://localhost/))



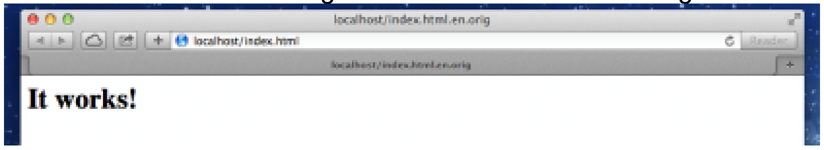
- **User Directory:** User's Home Directory > Sites
  [http://youripaddresshere/~username](http://youripaddresshere/~username)

In **OS X Mountain Lion**, Apple has removed the option for personal Web sharing; however, this feature can be enabled if needed.

- Open the OS X Terminal utility
- Create and edit an Apache user configuration file named after your user account by running the following command.
  ```
  sudo pico /etc/apache2/users/`whoami`.conf
  ```
- Modify the following text into the Terminal editor that opens:
  ```
  <Directory "/Users/username/Sites/">
          Options Indexes MultiViews
          AllowOverride All
          Order allow,deny
          Allow from all
  </Directory>
  ```
- After this action is performed, create a folder called "Sites" in your home directory if it is not already present, and then place your Web pages within it.
- To enable the Web server you have two options.
  - The first is to temporarily start it using the following terminal command (to disable the server, repeat the command with "stop" instead of "start"):
    ```
    sudo apachectl start
    ```
  - To enable the server even after subsequent reboots, then you will need to enable the launch daemon for the server, which can be done with the following command:
    ```
    sudo defaults write
    /System/Library/LaunchDaemons/org.apache.httpd Disabled -bool
    false
    ```

In **Mavericks**, load the Apache Web server by opening up the Terminal app. Type the command:
sudo -s launchctl load -w /System/Library/LaunchDaemons/org.apache.httpd.plist
(To unload the web server, use the same launchctl command by typing the following command in the Terminal app:
sudo -s launchctl unload -w /System/Library/LaunchDaemons/org.apache.httpd.plist )

Once you've loaded the Apache Web server software, you can check if it is now running in the background by opening the Safari web browser on the Mac and typing *http://localhost/* in the address box. You should get a result similar to the image below.



It works!

## Using the bundled PHP

PHP has come standard with Macs since OS X version 10.0.0. Enabling PHP with the default web server requires uncommenting a few lines in the Apache configuration file httpd.conf

- Locate and open the Apache configuration file.
  ```
  sudo pico /private/etc/apache2/httpd.conf
  ```
- uncomment the lines (by removing the #) that look similar to the following (these two lines are often not together, locate them both in the file):
  # LoadModule php5_module libexec/httpd/libphp5.so
- Create a phpinfo() file; the phpinfo() function will display information about PHP. Consider creating a file in the DocumentRoot with the following PHP code:

```
<?php phpinfo(); ?>
```

- Restart Apache, and load the PHP file created above. To restart, either execute `sudo apachectl graceful` in the shell or stop/start the "Personal Web Server" option in the OS X System Preferences.
- Example: Listing all information about the users browser

```
<?php
echo $_SERVER['HTTP_USER_AGENT'] . "\n\n";
$browser = get_browser(null, true);
print_r($browser);
?>
```

## Detecting iPhone Traffic & Redirection

How do you redirect visitors from the iPhone to your web app? We can do this with a simple PHP script placed at the top of your page before any header output. This will search the visitor's user agent string for the word "iPhone". If detected, they will be redirected to the URL you specify.

```
<?php
$iphone = strpos($_SERVER['HTTP_USER_AGENT'],"iPhone");
if ($iphone == true){ header("Location: mywebapp.html");  }
else { header("Location: myregular.html");  }
?>
```

## Setting Up Your Basic Page Framework

- Scaling Your Page to Fit the Viewport Window

  The iPhone viewport dimensions are 320x480 pixels. For your content to be readable without zooming your page needs to be scaled to fit the viewport width. To do this we are going to add the following meta tags between the <head></head> tags of the HTML.

```
<meta name="viewport" content="width=device-width; initial-
scale=1.0; maximum-scale=1.0; user-scalable=0;" />
```

  This will detect the visitor's device width and scale your page appropriately, whether in portrait or in landscape mode.

- Adding also the following meta tags will enable your web application to run in "Full Screen Mode", which means that when launched from the home screen the address and status bars will vanish, making your web app appear just as a native iPhone app would.

  To launch a web app in full screen mode it must first be added to home screen. This is as easy as tapping the action icon in the status bar, as if to add as a bookmark, but instead tap the button "Add to Home Screen". An icon will be added to your home screen that can be launched just like a native iPhone app.

- Defining An Icon For Your Web App

You can define a custom icon for your web app by including the following code between the <head></head> tags of your HTML.

```
<link rel="apple-touch-icon" href="icon.png" />
```

Your icon must be 57x57 pixels. By default, the iPhone will round the corners of your icon and overlay it with the famous glossy gradient, making it look like a three dimensional button.

- Defining A Startup Splash Image

  When launching your application from the Home Screen you can opt to display a startup splash image of your choice. Include the following code between the <head> tags of your HTML.

  ```
  <link rel="apple-touch-startup-image" href="startup.png" />
  ```

  The startup image must fit the dimensions of the viewport, minus the height of the status bar, which would be 320x460 pixels.

- Change Stylesheet, Show/Hide Content on Orientation Change

  Sometimes you may wish to have separate CSS styles for portrait and landscape mode. If that is the case, you can easily switch between the two with a bit of JavaScript. Place the following code between the <head></head> tags in your HTML.

  ```
  <script type="application/x-javascript">
  window.addEventListener('load', setOrientation, false);
  window.addEventListener('orientationchange', setOrientation,
  false);
  function setOrientation() {
        var orient = Math.abs(window.orientation) === 90 ?
        'landscape' : 'portrait';
        var cl = document.body.className;
        cl = cl.replace(/portrait|landscape/, orient);
        document.body.className = cl;
       }
   </script>
  ```

  Add the class name "portrait" to your <body> tag.

  ```
  <body class="portrait">
  ```

  Lastly, customize your CSS stylesheet.

  ```
  <style type="text/css">
  body.portrait {
     background-color: black; //page background will be black
  }
  body.landscape {
  ```

```
   background-color: white; //page background will be white
}
</style>
```

If you have a <div> in your page with the ID "hidden", the content within that <div> will appear only when the phone is in landscape mode (or vice versa, depending on how you customize the CSS). This is an invaluable way of displaying content that requires varying widths. For example, portrait mode could display chart data for the previous 7 days; landscape mode could display chart data for the previous 30 days. Or you could modify this to display a thumbnail image in portrait mode and a full sized image in landspace mode. The possibilities are endless.

```
body.portrait #hidden {
   visibility: hidden;
   height: 0;
   width: 0;
   padding: 0;
   margin: 0;
   display: none;
}

body.landscape #hidden {
   visibility: visible;
   display: block;
}
```

## Offline iPhone Web Apps

One of the several features outlined in the HTML 5 specification is the support for Web applications that continue to work while they are offline. This feature is very useful for Web applications because a Web application can be loaded just once and then run offline without needing a persistent Internet connection, making it behave just like a locally installed native application.

- In the index.html page, set the manifest attribute of the html element to "OfflineApp.manifest":

  ```
  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
  <html manifest="OfflineApp.manifest">
  <head>
  ```

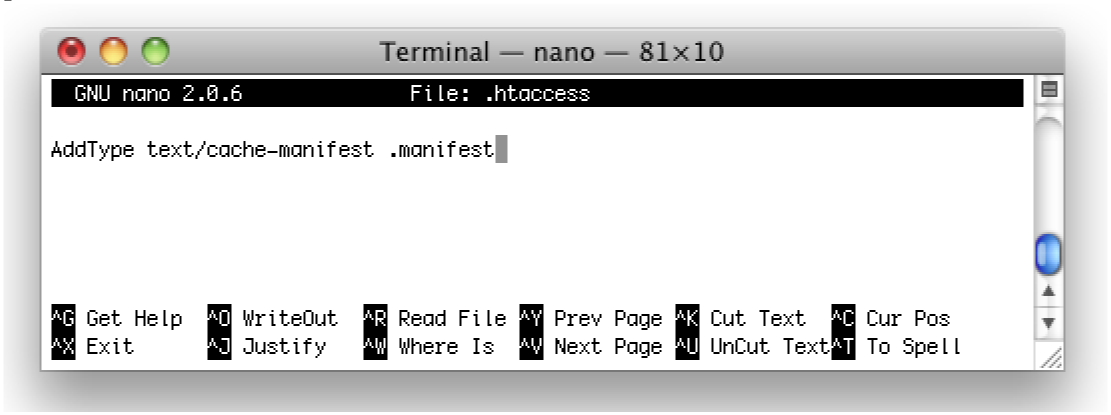
  This offline manifest file will contain the list of files that the browser needs to cache in order to allow the application to run in offline mode.

- Configuring Apache for Web Publishing

  In order for your web server to serve the manifest file correctly, you need to configure your web server (which is Apache in Mac OS X) to add the correct MIME type.

  Launch the Terminal application. Type the following

```
cd Sites
pico .htaccess
```



```
Terminal — nano — 81×10
 GNU nano 2.0.6              File: .htaccess

AddType text/cache-manifest .manifest




^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Save the file by pressing Ctrl-X. Type Y to go ahead and save the file. This will enable Apache to serve the manifest file correctly with the text/cache-manifest MIME type.

## Detecting Online and Offline Modes

Code the following statements to be called with `load` addEventListener:

```
function load()
{
    // Indicates whether the browser is online or offline
    var online = window.navigator.onLine;

    if (!online) {
        // Handle the case when the browser is offline
        alert("We are offline!");
    } else {
        alert("We are online!");
    }
}
```

manifest sample

```
CACHE MANIFEST

# Version 3

mypage.css
index.html
mypage.js

# offline icons
startup.png
icon.png
```

## Sources

- How to Turn Your Mac Into a Web Server http://www.macinstruct.com/node/112
- How to enable Web Sharing in OS X Mountain Lion http://reviews.cnet.com/8301-13727_7-57481978-263/how-to-enable-web-sharing-in-os-x-mountain-lion/
- How to enable Web Sharing in OS X Mavericks http://igerry.com/desktop/apple-os/enable-apache-web-server-os-x-mavericks.html
- Set up localhost on macOS High Sierra https://websitebeaver.com/set-up-localhost-on-macos-high-sierra-apache-mysql-and-php-7-with-sslhttps
- Using the bundled PHP http://www.php.net/manual/en/install.macosx.bundled.php
- PHP user agent http://php.net/manual/en/function.get-browser.php
- Step-by-step guide to building an iPhone web app http://www.jamiebutler.com/tutorials/iphone_web.php
- Optimizing Mobile Web Apps for iOS http://blog.teamtreehouse.com/optimizing-mobile-web-apps-ios
- How to Make an HTML5 iPhone App http://sixrevisions.com/web-development/html5-iphone-app/
- Offline iPhone Web Apps http://mobiforge.com/developing/story/offline-iphone-web-apps