



WiFinder

Launch Instructions

Contents

1	Setting up your server	2
1.1	Apache2	2
1.2	MySQL	2
1.3	PHP	2
1.4	phpMyAdmin	3
2	Installing WiFinder	5
2.1	Downloading WiFinder from Github	5
2.2	Setting up WiFinder	5
2.3	Integrating Google log in	5
3	Customizing and Maintaining WiFinder	6
3.1	API Details	6
3.2	Binary data upload	6
3.3	Customize Logos	6

1 Setting up your server

WiFinder is designed to run on a LAMP(Linux, Apache2, MySQL, PHP) server, but there is no reason why it can't run on another OS apart from Linux Debian 8. However, the design do require Apache2, MySQL and PHP.

1.1 Apache2

Apache2 is the worlds most used web server and is completely free to use.

For Linux

Command: `sudo apt-get install apache2`

For Windows:

Download Apache2 from:

<https://httpd.apache.org/download.cgi>
and install it.

1.2 MySQL

MySQL is a relational database, it will store all your server side information such as users.

For Linux

Command: `sudo apt-get install mysql-server`
and set up your database user and password

For Windows:

Download MySQL from:

<https://www.mysql.com/downloads/>
and install it. Set up your database user and password

1.3 PHP

PHP is a script language that bridges the communication gap between user and server

For Linux

Command: `sudo apt-get install php5 libapache2-mod-php5`
Restart your server

For Windows:

Download PHP from:

<http://windows.php.net/download/>
and install it. Restart your server.

1.4 phpMyAdmin

phpMyAdmin is an interface for interacting with your database and is a useful admin tool. It does require some set up after installation

For Linux

Command: `sudo apt-get install phpmyadmin php-mbstring php-gettext`

For the server selection, choose apache2.

Select yes when asked whether to use 'dbconfig-common' to set up the database.

You will be prompted for your database administrator's password.

You will then be asked to choose and confirm a password for the phpMyAdmin application itself.

Command: `sudo phpenmod mcrypt`

Command: `sudo phpenmod mbstring`

Command: `sudo systemctl restart apache2`

Command: `sudo nano /etc/apache2/conf-available/phpmyadmin.conf`

Add 'AllowOverride All' under

Directory /usr/share/phpmyadmin

Save the file and again, issue the command

Command: `sudo systemctl restart apache2`

Command: `sudo nano /usr/share/phpmyadmin/.htaccess`

In the file, add the following text and save the file

AuthType Basic

AuthName "Restricted Files"

AuthUserFile /etc/phpmyadmin/.htpasswd

Require valid-user

Command: `sudo apt-get install apache2-utils`

Command: `sudo htpasswd -c /etc/phpmyadmin/.htpasswd yourdesiredusername`

Choose a password and phpMyAdmin will be operational and found at

https://domain_name_or_IP/phpmyadmin

For Windows:

Download phpMyAdmin from:

<https://www.phpmyadmin.net/downloads/>

and install it. Then extract and place phpMyAdmin into localhost's DocumentRoot folder generally found at C:\Program Files\Apache Software Foundation\Apache2.4\htdocs\

Then open

C:\Program Files\Apache Software Foundation\Apache2.4\htdocs\phpmyadmin\config.inc.php

Look for:

```
$cfg['Servers'][$i]['controlhost'] = 'localhost';
```

```
$cfg['Servers'][$i]['controluser'] = 'pma';
```

```
$cfg['Servers'][$i]['controlpass'] = 'pmapass';
```

Enter your user name instead of 'pma' and your password instead of 'pmapass', save the file and phpMyAdmin will be operational and found at

https://domain_name_or_IP/phpmyadmin

2 Installing WiFinder

WiFinder requires downloading, a set up and creating a google API user.

2.1 Downloading WiFinder from Github

Go to the following link and download the WiFinder app

<https://github.com/NWretborn/WiFinder/blob/master/WiFinder.rar>

Unzip the file in the location of your web site folder generally found at

C:\Program Files\Apache Software Foundation\Apache2.4\htdocs\

for windows or

/var/www




for Linux.

2.2 Setting up WiFinder



Go to

https://domain_name_or_IP/phpmyadmin

and log in. Create a new database and name it e.g "WiFinder". In the database, create a table called marker and one called user. Below is a picture of the structure of the tables, design them with the same names and structure as the picture.

				id	user	karma	name	address	quality	lat	lng	type	picurl
<input type="checkbox"/>				1	Admin	0	Root			999.000000	999.000000		

The above image is a picture of the marker table. Create it in the same manner and add a marker with the same values as shown in the picture.

				id	name	password	mail
<input type="checkbox"/>				1	admin	\$2y\$10\$OwKTIEOzGnQo.Z5O1g.P2uvAQzRySxtsmpWotAPx7ol...	admin@wifinder.com

The above image is a picture of the user table. Create it in the same manner. The user is just an example. Next, go into the WiFinder folder, open the file named
phpsqlajax_dbinfo.php

In this file, enter the information of your database.

2.3 Integrating Google log in

Create a google API console project and receive a client ID from

<https://developers.google.com/identity/sign-in/web/devconsole-project>

Once you have your client ID, you need to replace

meta name="google-signin-client_id" content="YOUR_CLIENT_ID.apps.googleusercontent.com"
in the following files

index.php

map.php

loginerror.php

with your own client ID.

3 Customizing and Maintaining WiFinder

The project repository can be found at Github along with updates and patches <https://github.com/NWretborn/WiFinder>

3.1 API Details

The Rest API is written in PHP and is documented at https://domain_name_or_IP/api_documentation.php

The frontend uses ajax POST requests to send data to the API. The API then issues a SQL query to the backend.

3.2 Binary data upload

When users upload pictures with their network, the images are saved in the 'uploaded_files' folder. They are saved with their file name pointing to the location they belong to with latitude and longitude coordinates. The application uses 'upload.processor.php' to upload the file to the folder.

3.3 Customize Logos

In the '\WiFinder\img' folder, you will find all the logos used in the application. If you wish to use different icons, name them with the same name and file typ in order to avoid having to replace file names in different .php and .js documents.