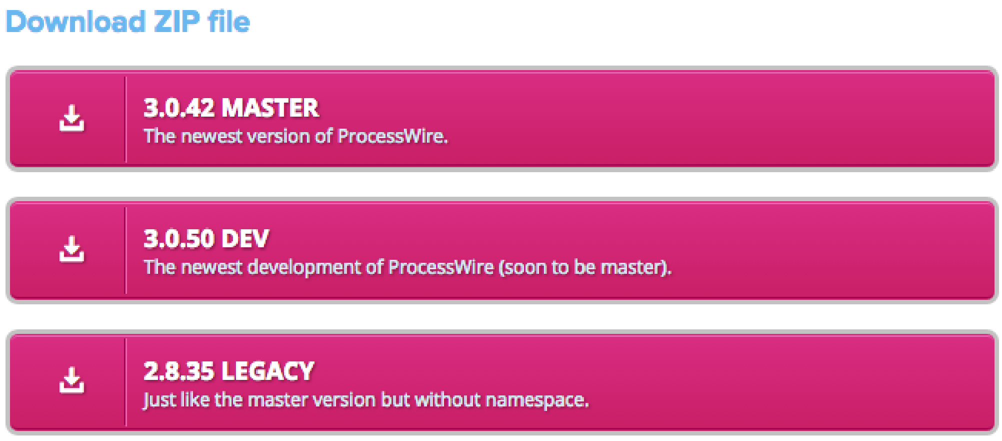
how-to-install-processwire

ProcessWire installation is quite simple. There are various ways to get the latest release like traditional downloading *git*, using https://wireshell.pw/, http://grab.pw or with https://getcomposer.org. Since the more straightforward way is the traditional downloading we will use that method.

Downloading the Latest Version

You must go to https://processwire.com/download/ and select the current version thats more suitable for your needs. The current 3.x version as the time of writing is *3.0.61*.



Download ZIP file

3.0.42 MASTER
The newest version of ProcessWire.

3.0.50 DEV
The newest development of ProcessWire (soon to be master).

2.8.35 LEGACY
Just like the master version but without namespace.

Which version should I use?

ProcessWire comes in two different versions. One with PHP namespaces and the other with no namespaces. A good number of modules where made before *ProcessWire* supported namespaces, but thankfully *ProcessWire* is smart enough and loads them without problems. Namespaces are quite handy for a more clean access to *ProcessWire* properties and methods and allow new kinds of arquitectures.

ProcessWire 3.x

Recommended when creating new sites or downloading *ProcessWire* for the first time. Namespaces enables using *composer*, so you can access the vast amount of code that *composer* brings. Also namespaces allow multiple *ProcessWire* instances (Access to another PW Installation).

```
The PHP namespace used by ProcessWire is:

<?php namespace ProcessWire; ?>
```

ProcessWire 2.8.x legacy

Recommended when you want the same functionality as 3.x but in a non-namespace environment like in 2.7. Usually when old projects needs security updates and you don’t need or want namespace support. No *composer* or multiple instances allowed in this version.

Dev or Master?

Master is the version with the most stable and tested code. Conversely *Dev* may contain new bugs because the adding of features. If you need something reliable use *Master*, if you want to try the new goodies use *Dev*. If you found a bug, need help or want to request a new feature please report them in the official github repository (https://github.com/processwire) or the forums (https://processwire.com/talk).

Server Requirements

ProcessWire needs a basic setup that is found on a great number of hosting providers.

The following are the minimum requirements:

- A Unix or Windows-based web server running Apache.
- PHP version 5.3.8 or newer with PDO database support (PHP 5.6+ preferable).
- MySQL or MariaDB, 5.0.15 or greater (5.6+ preferable).
- Apache must have mod_rewrite enabled (when applicable).
- Apache must support .htaccess files (when applicable).
- PHP's bundled GD 2 library (ImageMagick also supported).

Is Only Apache Supported?

Other web servers can be used. But the default configuration is only tailored to Apache. Users reported successfully running *ProcessWire* with https://www.nginx.com or https://caddyserver.com. Web servers besides Apache should be used at your own risk.

Are other Databases besides MySQL (MariaDB) Supported?

No. Other databases like *SQLite*, *Oracle*, *PostgreSQL* or *Microsoft SQL Server* are not supported. The main reason is because *ProcessWire* is highly optimized and tailored to fit the*MySQL* technology. This enables storing huge amount of data and executing complex queries in a super efficient way.

Installation

We will assume that you already have a working PHP, Apache and MySQL enviroment that meets at least the minimum *ProcessWire* requirements. Popular PHP dev apps are Mamp https://www.mamp.info/en/, Ampps http://www.ampps.com or Xampp https://www.apachefriends.org/es/index.html. We will use the command line for the MySQL configuration. Although you could use tools like *PHPMyAdmin* that comes in those applications for that task if you like. Also modern tools like*Laravel's Valet* https://laravel.com/docs/master/valet are supported too see https://processwire.com/talk/topic/13246-laravel-valet-with-processwire/

Create a ProcessWire Database

With all the prerequisites in place, we can go ahead and create a new MySQL database and user for*ProcessWire*.

First, log into the MySQL Shell:

```
$ mysql -u root -p
```

Now, create the database and user:

```
CREATE DATABASE processwire;
CREATE USER pwuser@localhost;
SET PASSWORD FOR pwuser@localhost= PASSWORD("password");
GRANT ALL PRIVILEGES ON processwire.* TO pwuser@localhost
IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
exit
```

A couple of things are going on here:

1. Create the actual `processwire` database.
2. Create the user `pwuser`.
3. Set a password for this user.

4. Grant all privileges of the `processwire` database to this user.
5. Reload the new user settings.

Feel free to name your database or user differently.

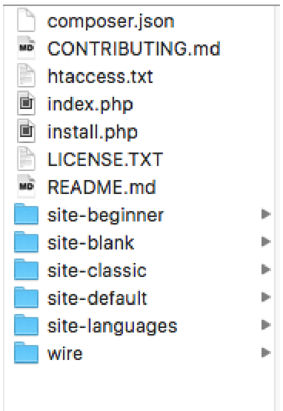
Download and Copy the Files

We will go to the download page and grab the latest masterhttps://github.com/processwire/processwire/archive/master.zip. And unzip it inside the directory used for development. If you prefer using the command line you could use (Unix compatible):

```
$ curl  -SL http://grab.pw > pw.zip && unzip pw.zip && rm pw.zip
```

If all went well you should see the following files:

```
CONTRIBUTING.md composer.json   install.php     site-classic    wire
LICENSE.TXT     htaccess.txt    site-beginner   site-default
README.md       index.php       site-blank      site-languages
```



composer.json
CONTRIBUTING.md
htaccess.txt
index.php
install.php
LICENSE.TXT
README.md
site-beginner
site-blank
site-classic
site-default
site-languages
wire

Post Installation Modifications

After the installation all the `site-*` directories will be deleted and replaced by a single `site` with the chosen profile. Here we only will use the `blank` profile.

The `htaccess.txt` will be renamed to `.htaccess` (Only valid in Apache Server).

All these modifications all done automatically by the `installer.php` file.

Important Files and Directories

The following items are important. And should not be deleted.

composer.json

This file enables the *composer* integration. Should not be deleted if you want to use that tool.

index.php

Holds the main script for *ProcessWire*. If you want to use *ProcessWire* functions and properties in another script you must include this file like this:

```
<?php
include("/path/to/processwire/index.php");
```

wire/

This directory contains all the *core* files. It should not be touched. Only for upgrading the core version.

site/

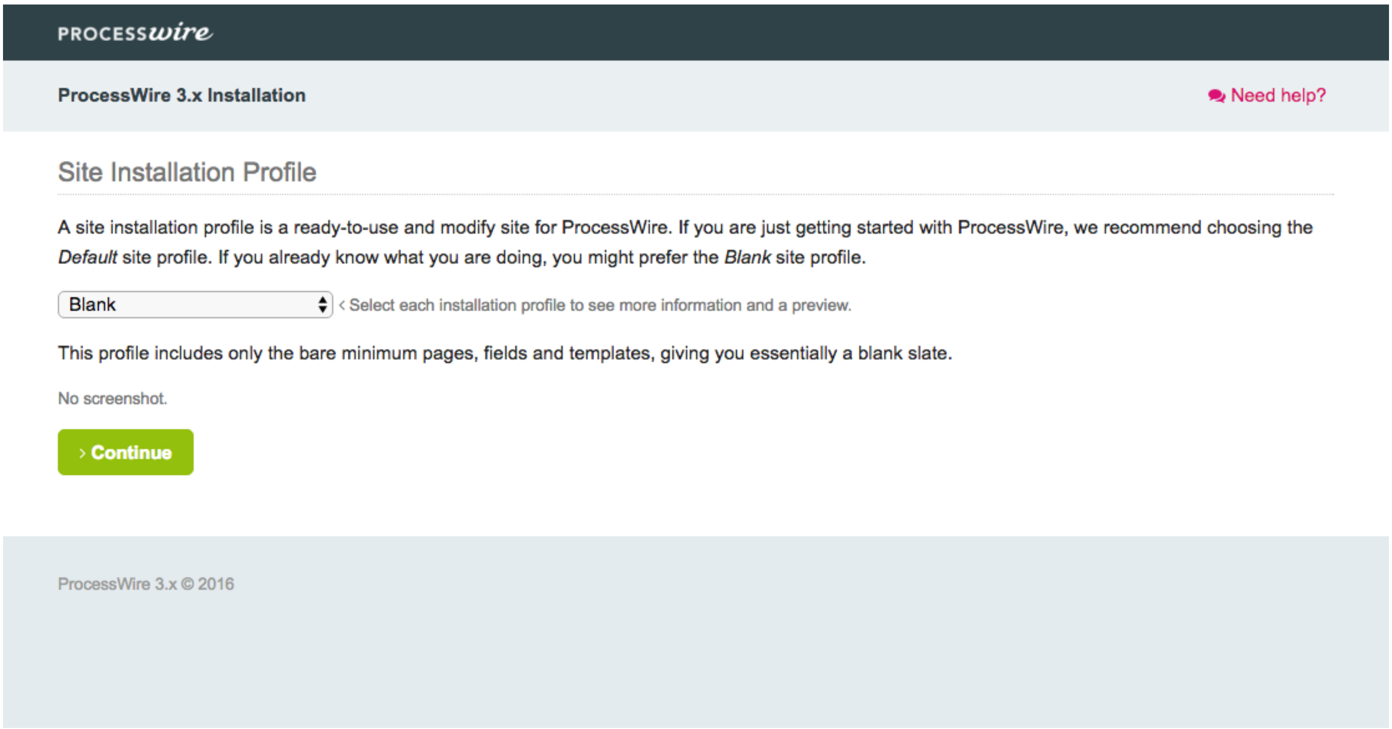
All the files related to the site (templates, modules, logs, images) are organized in this directory.

.htaccess

Defines the routes and security rules for Apache. Do not delete if you use Apache.

Begin the Installation

Open your browser and navigate to your defined development directory. You should be welcomed by the*ProcessWire* installer. The first decision to make is selecting the desired profile. These profiles contain example code that you can explore further by your own. The profile we will use across this article is the `Blank` profile.



PROCESSwire

ProcessWire 3.x Installation

Need help?

Site Installation Profile

A site installation profile is a ready-to-use and modify site for ProcessWire. If you are just getting started with ProcessWire, we recommend choosing the *Default* site profile. If you already know what you are doing, you might prefer the *Blank* site profile.

Blank ‹ Select each installation profile to see more information and a preview.

This profile includes only the bare minimum pages, fields and templates, giving you essentially a blank slate.

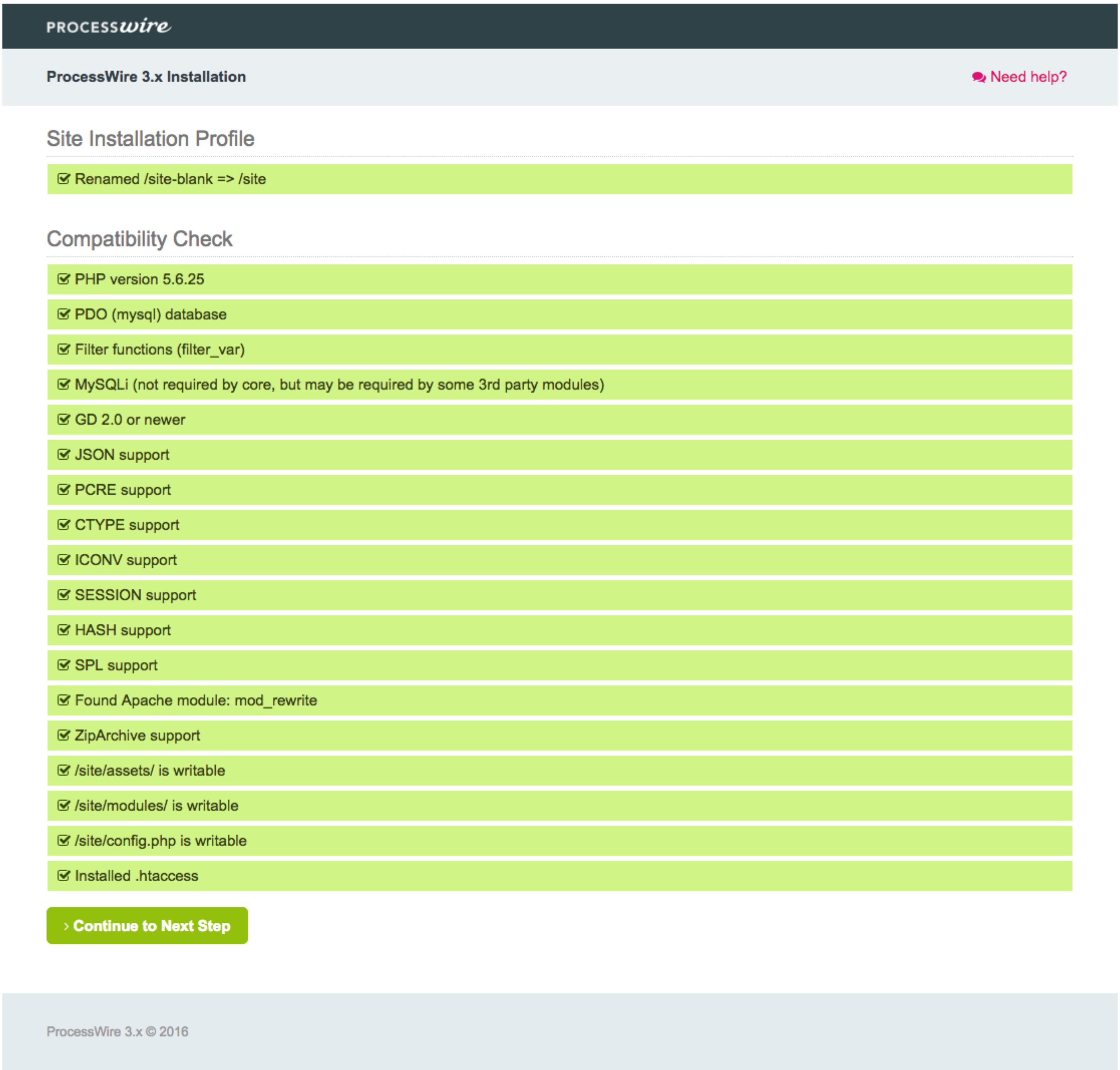
No screenshot.

› Continue

ProcessWire 3.x © 2016

Server Check

After selecting the desired profile,*ProcessWire* will check if your server is compatible with the minimum requirements. If something is missing you should check your server configuration and try again.



ProcessWire 3.x Installation

Site Installation Profile

☑ Renamed /site-blank => /site

Compatibility Check

☑ PHP version 5.6.25

☑ PDO (mysql) database

☑ Filter functions (filter_var)

☑ MySQLi (not required by core, but may be required by some 3rd party modules)

☑ GD 2.0 or newer

☑ JSON support

☑ PCRE support

☑ CTYPE support

☑ ICONV support

☑ SESSION support

☑ HASH support

☑ SPL support

☑ Found Apache module: mod_rewrite

☑ ZipArchive support

☑ /site/assets/ is writable

☑ /site/modules/ is writable

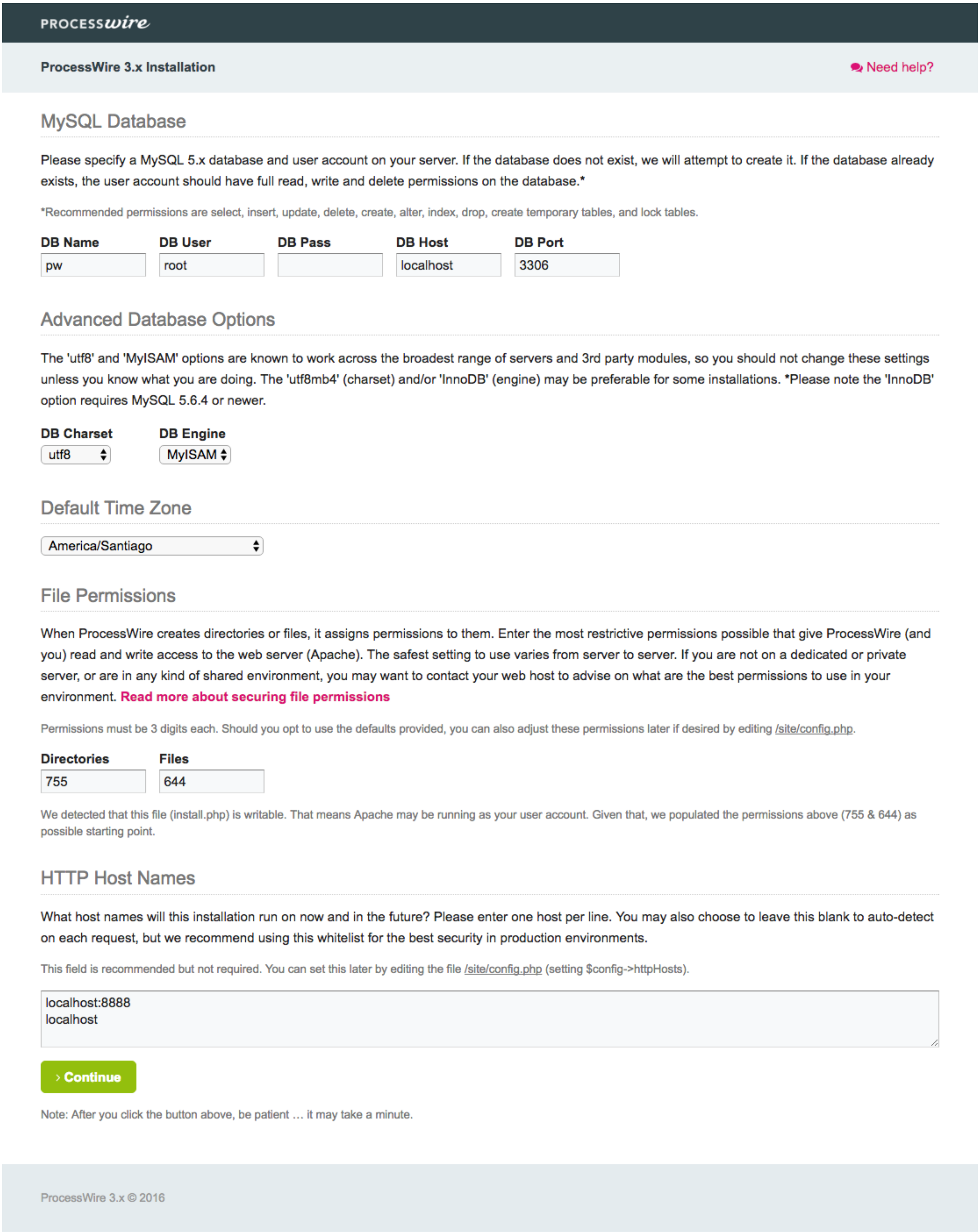
☑ /site/config.php is writable

☑ Installed .htaccess

ProcessWire 3.x © 2016

Database Config

If all the server requirements were met, now you will need to tell*ProcessWire* the database credentials (*user*, *password*, *server address*, *server port*) and optionally, the database charset and engine.



ProcessWire 3.x Installation

MySQL Database

Please specify a MySQL 5.x database and user account on your server. If the database does not exist, we will attempt to create it. If the database already exists, the user account should have full read, write and delete permissions on the database.*

*Recommended permissions are select, insert, update, delete, create, alter, index, drop, create temporary tables, and lock tables.

DB Name	DB User	DB Pass	DB Host	DB Port
pw	root		localhost	3306

Advanced Database Options

The 'utf8' and 'MyISAM' options are known to work across the broadest range of servers and 3rd party modules, so you should not change these settings unless you know what you are doing. The 'utf8mb4' (charset) and/or 'InnoDB' (engine) may be preferable for some installations. *Please note the 'InnoDB' option requires MySQL 5.6.4 or newer.

DB Charset	DB Engine
utf8	MyISAM

Default Time Zone

America/Santiago

File Permissions

When ProcessWire creates directories or files, it assigns permissions to them. Enter the most restrictive permissions possible that give ProcessWire (and you) read and write access to the web server (Apache). The safest setting to use varies from server to server. If you are not on a dedicated or private server, or are in any kind of shared environment, you may want to contact your web host to advise on what are the best permissions to use in your environment. **Read more about securing file permissions**

Permissions must be 3 digits each. Should you opt to use the defaults provided, you can also adjust these permissions later if desired by editing /site/config.php.

Directories	Files
755	644

We detected that this file (install.php) is writable. That means Apache may be running as your user account. Given that, we populated the permissions above (755 & 644) as possible starting point.

HTTP Host Names

What host names will this installation run on now and in the future? Please enter one host per line. You may also choose to leave this blank to auto-detect on each request, but we recommend using this whitelist for the best security in production environments.

This field is recommended but not required. You can set this later by editing the file /site/config.php (setting $config->httpHosts).

```
localhost:8888
localhost
```

Note: After you click the button above, be patient … it may take a minute.

ProcessWire 3.x © 2016

Which engine is better, MyISAM or InnoDB?

The only reason *ProcessWire* doesn't default to InnoDB is because *PW* makes significant use of fulltext indexes, and InnoDB didn't support them until MySQL 5.6.4. If you want to use InnoDB just make sure that both your development and production environments support it before choosing the option.

While there are many benefits to InnoDB relative to MyISAM, it is admittedly rare in the *ProcessWire* world that we experience the relative drawbacks of MyISAM. But you'll likely notice real benefits from using InnoDB if working with a high traffic site that needs to perform a lot of saving of pages (whether automated, or from a team of people making edits at once). You might also see significant benefits on sites that need to do regular automated imports of data that update a lot of pages. Why? InnoDB doesn't need to lock the entire table in the way MyISAM does to perform such operations, so there's real potential for improved performance in these installations. For this reason, *ProcessWire* has always used InnoDB on certain tables (like those for user sessions). But now you can specify that *ProcessWire* should use InnoDB for all tables.

Be careful with the InnoDB selection because it could be a real problem if your development server is running MySQL 5.6.4 or newer, while your production server isn't. In such a case, the production server wouldn't be able to import the development server database.

Which charset is better, utf8 or utf8mb4?

To use *utf8mb4* charset, you'll need MySQL 5.5.3 or newer. When in use, it enables*ProcessWire* to store 4-byte characters rather than just 3-byte characters as in *utf8*.

Why would you need *utf8mb4* relative to *utf8*? Most probably don't, but there's been demand for it specific to some languages as *utf8mb4* greatly expands the number of characters that can be represented. Outside of language needs, it's what would enable you to use *emoji* in *ProcessWire*, for instance.

Because *utf8mb4* uses more bytes per character, it places new limits on the length of indexes used by*ProcessWire*. *ProcessWire* has several 255-character index lengths, and the maximum allowed by*utf8mb4* are 250 with MyISAM and 191 using InnoDB (MySQL Prefixes can be up to 1000 bytes long with MyISAM and 767 bytes for InnoDB tables. Since *utf8bm4* is up to 4 bytes the key length should be up to 250 chars `4 * 250 = 1000 bytes` in MyISAM, and 191 chars in InnoDB `4 * 191 = 764 bytes`). For this reason, all of the core*Fieldtypes* use no more than 250 length indexes in MyISAM or 191 length indexes in InnoDB in order to support *utf8mb4*. However, it's possible that 3rd party modules might be using index lengths that aren't compatible with *utf8mb4*, so this is something to keep in mind.

Like with the InnoDB selection, before using this option, you'll need to make sure that your MySQL versions are new enough between your development and production servers. Though since the requirements are for MySQL 5.5.3 or newer, chances are most will be okay here. However, because not many are currently using *utf8mb4* support in *ProcessWire*, you should consider using it only if needed.

Timezone

Modules and time related fields like `$page->created` use this timezone info for displaying date and time correctly. You should select the one that makes more sense for your need. The default value displayed in the installation depends on the *php.ini* setting value. Example `date.timezone = "America/Santiago"`.

File Permissions

When *ProcessWire* creates directories or files, it assigns permissions to them. When you install*ProcessWire*, it performs a check to see if the *install.php* file is writable. If it is, then there's a good chance (though not a guarantee) that Apache runs as your user account. If it detects this, it will recommend the 755 permission for writable directories and 644 permission for writable files, as a starting point. This translates to directories and files that are writable only to you, but readable to everyone else.

If the installer populates 777 and 666 permissions, this translates to directories and files that are readable and writable to everyone, which is not a good scenario in shared environments. But without knowing more about the hosting environment, they may be the only permissions that we know for certain will enable *ProcessWire* to run. In either case, you should read the file permissions docs for more details. https://processwire.com/docs/security/file-permissions/.

Usually we only will need the default recommended values that are 755 permission for directories and 644 permission for files.

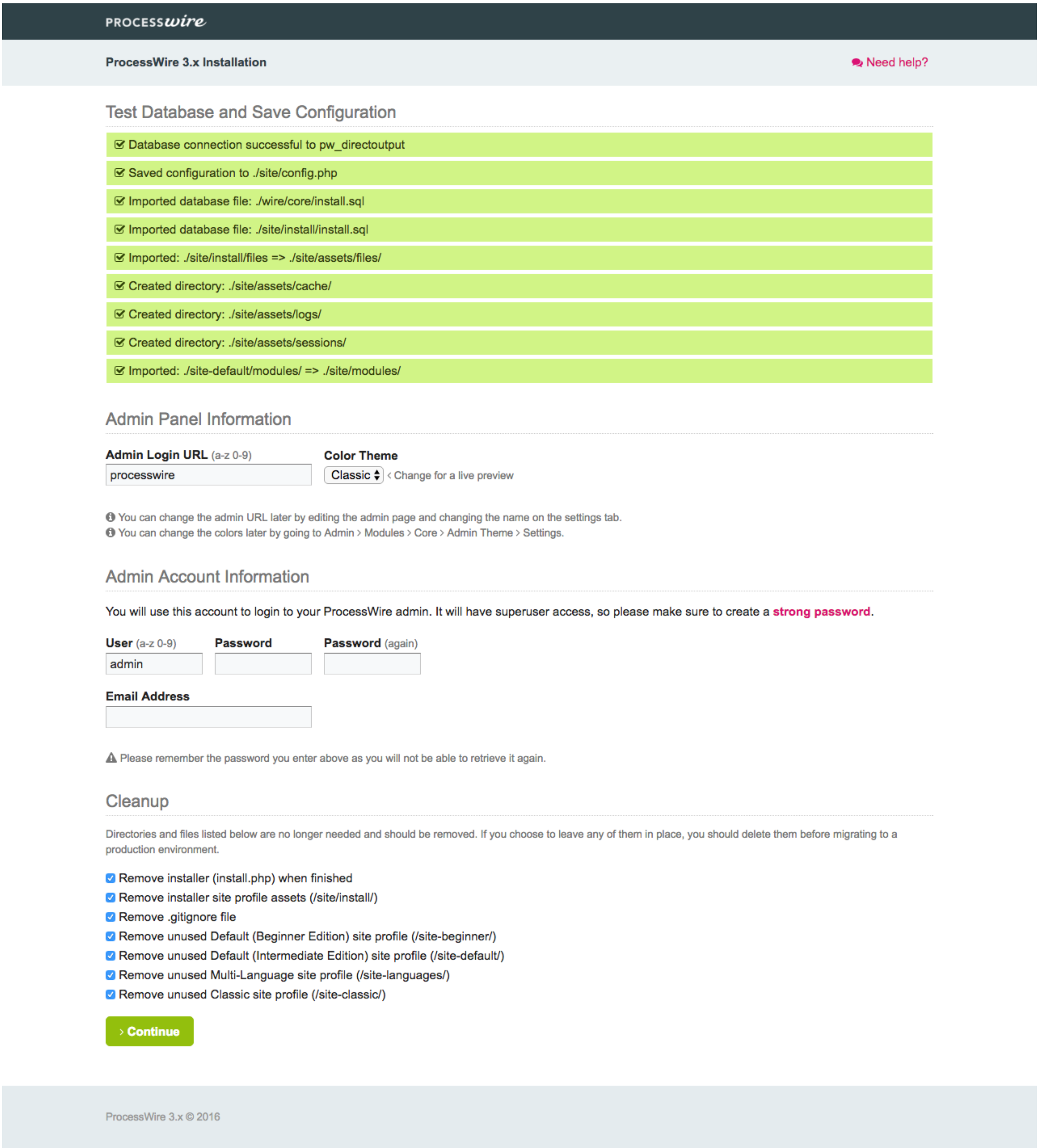
Host Names

What host names will this installation run on now and in the future?. You may also choose to leave this blank to auto-detect on each request, but we recommend using this whitelist for the best security in production environments.

The recommended choice is leaving it blank on development enviroments and only fill it when executing on production. If you leave values used in development (example `localhost`) in this list, some properties like `$page->httpUrl` could contain localhost addresses even when your are using your production enviroment. Also recommended is that you fill all the possible subdomains (`example.com, www.example.com`) or you will get a warning message when you log in the admin panel.

Last Step

Finally the only configuration left is the administration url and admin user config. For additional security it's recommended that the *Admin Login URL* should be other than*admin* or *processwire* and more related to your site context. For example in a food related site the admin page could be called `kitchen`. Hiding your admin URL is a good practice. But if strong passwords are used, as they should be, there's no security problem with having a known admin URL either. It's recommended that you remove all the installation files and unused profiles.



PROCESSwire

ProcessWire 3.x Installation

Need help?

Test Database and Save Configuration

☑ Database connection successful to pw_directoutput

☑ Saved configuration to ./site/config.php

☑ Imported database file: ./wire/core/install.sql

☑ Imported database file: ./site/install/install.sql

☑ Imported: ./site/install/files => ./site/assets/files/

☑ Created directory: ./site/assets/cache/

☑ Created directory: ./site/assets/logs/

☑ Created directory: ./site/assets/sessions/

☑ Imported: ./site-default/modules/ => ./site/modules/

Admin Panel Information

Admin Login URL (a-z 0-9): processwire

Color Theme: Classic < Change for a live preview

You can change the admin URL later by editing the admin page and changing the name on the settings tab.
You can change the colors later by going to Admin > Modules > Core > Admin Theme > Settings.

Admin Account Information

You will use this account to login to your ProcessWire admin. It will have superuser access, so please make sure to create a **strong password**.

User (a-z 0-9): admin

Password

Password (again)

Email Address

Please remember the password you enter above as you will not be able to retrieve it again.

Cleanup

Directories and files listed below are no longer needed and should be removed. If you choose to leave any of them in place, you should delete them before migrating to a production environment.

☑ Remove installer (install.php) when finished
☑ Remove installer site profile assets (/site/install/)
☑ Remove .gitignore file
☑ Remove unused Default (Beginner Edition) site profile (/site-beginner/)
☑ Remove unused Default (Intermediate Edition) site profile (/site-default/)
☑ Remove unused Multi-Language site profile (/site-languages/)
☑ Remove unused Classic site profile (/site-classic/)

ProcessWire 3.x © 2016

Reset Admin URL or Password

If you forgot your admin url or user you can set new values by putting a script inside a template or the`site/ready.php` file.

Reset Admin Password

```php
<?php
$u = $users->get('admin'); // or whatever your username is
$u->of(false);
$u->pass = 'your-new-password';
$u->save();
```

Show Admin URL

```php
<?php
echo wire('config')->urls->admin;
```

If you want to use a separated file instead. Create a file named `reset.php` in the same directory of `index.php`.

```php
<?php

require "index.php";
$admin = wire('users')->get('admin');
$admin->setOutputFormatting(false);
$admin->set('pass', 'your-new-password');
$admin->save('pass');

?>
```

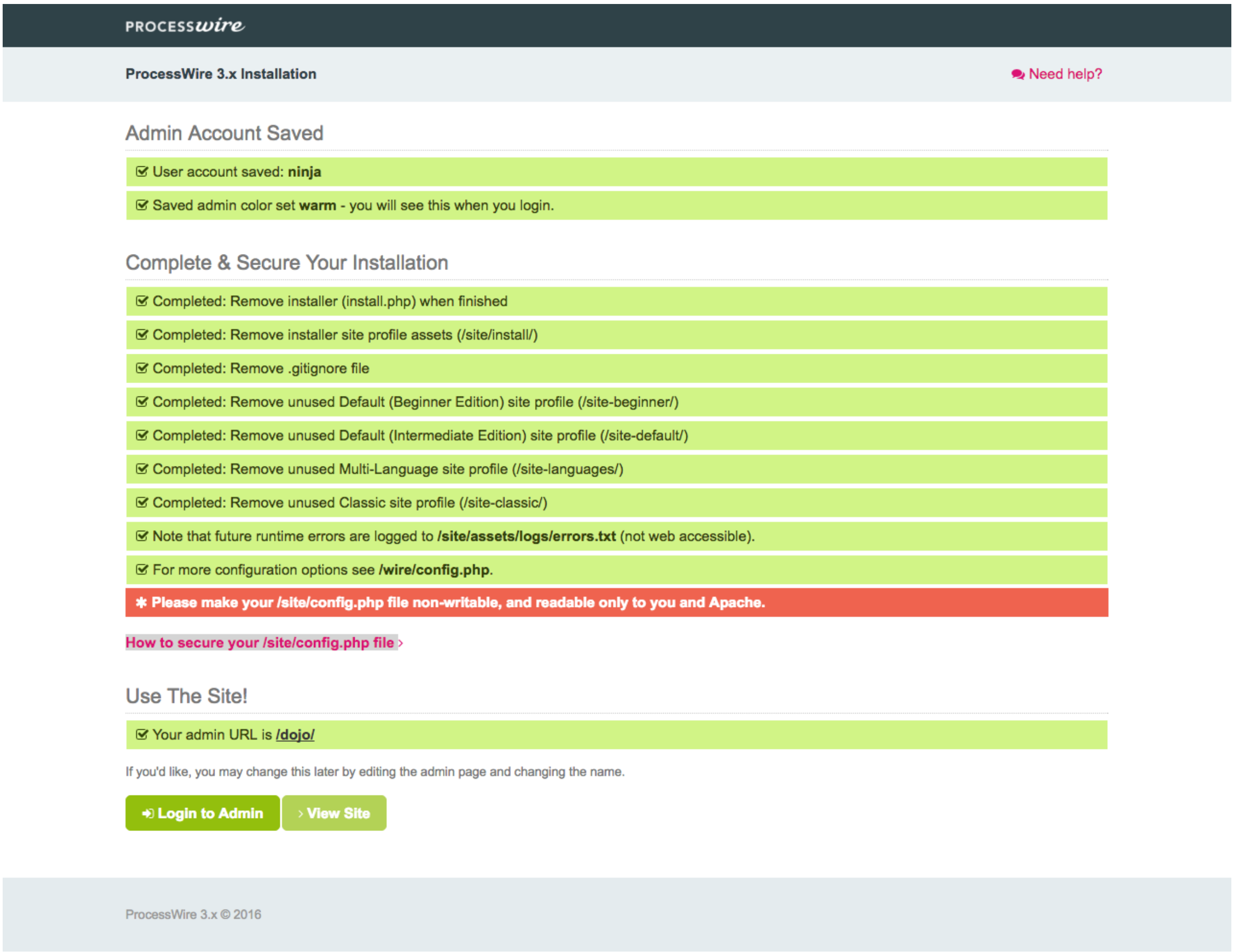
ProcessWire versions 2.6.9 and up could also use this shorter form:

```php

<?php

require "index.php";
$admin = $users->get('admin'); // or whatever your username is
$admin->setAndSave('pass', 'your-new-password');

?>
```

Remember to delete those codes when the job is done.

Jobs Done

Congratulations!, you now have a fully functional*ProcessWire* site. You can now visit you admin url and start creating awesome sites. It’s recommended on production enviroments that you *config.php* is configured with the correct permissions. (See File Permissions docs for more info).



ProcessWire 3.x Installation

Need help?

Admin Account Saved

☑ User account saved: **ninja**

☑ Saved admin color set **warm** - you will see this when you login.

Complete & Secure Your Installation

☑ Completed: Remove installer (install.php) when finished

☑ Completed: Remove installer site profile assets (/site/install/)

☑ Completed: Remove .gitignore file

☑ Completed: Remove unused Default (Beginner Edition) site profile (/site-beginner/)

☑ Completed: Remove unused Default (Intermediate Edition) site profile (/site-default/)

☑ Completed: Remove unused Multi-Language site profile (/site-languages/)

☑ Completed: Remove unused Classic site profile (/site-classic/)

☑ Note that future runtime errors are logged to **/site/assets/logs/errors.txt** (not web accessible).

☑ For more configuration options see **/wire/config.php**.

✱ Please make your /site/config.php file non-writable, and readable only to you and Apache.

How to secure your /site/config.php file ›

Use The Site!

☑ Your admin URL is **/dojo/**

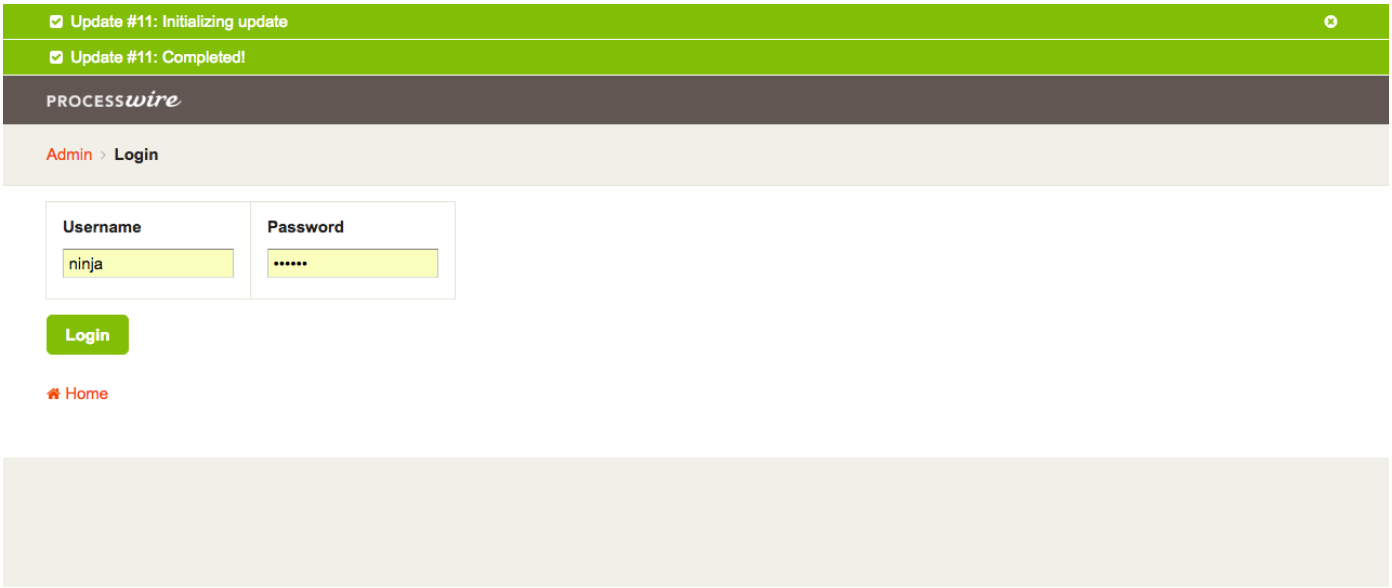
If you'd like, you may change this later by editing the admin page and changing the name.

ProcessWire 3.x © 2016

In the blank profile you will only see a Blank page with the Title*Home*.

Home

The admin panel will look similar to the following image.



Update #11: Initializing update

Update #11: Completed!

Username: ninja

Password: ••••••

Installation Completed

The installation of a typical *ProcessWire* system is easy and short. There are many ways like using a custom made*site-profile* or using *wireshell* (https://wireshell.pw). For the majority of the developers the previous step by step installation process will be good enough for their needs.

More Info

http://processwire.com/docs/install/

https://webdesign.tutsplus.com/tutorials/how-to-install-and-setup-processwire-cms–cms-25509

Special Thanks

I want to thank *fbg13* for it's corrections in the PW Forums :)
