

# Creating your own Super Repo

---

## Step 1 of 13 - Create the repo

First you need to create a repository at [github.com](https://github.com). Alternatively you can create a repo on your own machine if you prefer.

# Creating your own Super Repo

---

## Step 2 of 13 - Clone your repo

Clone the repository and make sure the master is checked out.

...

```
git clone https://github.com/username/newsuper.git
```

```
cd newsuper
```

```
git checkout master
```

...

## Step 3 of 13 - Add your documentation

It's best practise to add a README.md and LICENSE.md to your super repo. Both of these files are in markdown format and the README will automatically be presented on github.com.

The README should have the following information on what the purpose of your super repo is, and how to install it.

The LICENSE.md should contain information about what license the repo falls under.

Once you have created the files, add them to git.

...

```
git add README.md LICENSE.md
```

...

# Creating your own Super Repo

---

## Step 4 of 13 - Create the .gitignore file

The .gitignore file contains the files and directories that git should ignore when checking in. This is useful for your development directory so config files don't get included in the repository.

The following files should be included in the .gitignore file

```
***
site/.htaccess
site/ciniki-sync.php
site/ciniki-rest.php
site/ciniki-json.php
site/ciniki-manage.php
site/ciniki-login.php
site/paypal-ipn.php
site/ciniki-apis.php
site/ciniki-api.ini
site/ciniki-manage.ini
site/phpinfo.php
site/index.php
site/ciniki-code
site/ciniki-cache
site/ciniki-backups
logs/
/ssl
***
```

Add .gitignore to git:

```
***
git add .gitignore
***
```

## Step 5 of 13 - Setup your dev web server

Setup your development server to point at your checked out super repo directory. It's recommended to setup a separate virtual host and keep a logs directory. It's unnecessary for development, but you can also include a self signed SSL cert if required.

# Creating your own Super Repo

---

## Step 6 of 13 - Setup your htaccess file

Create your htaccess file. The following is recommended:

...

```
# Block evil spam bots
# List found on : http://perishablepress.com/press/2006/01/10/stupid-
htaccess-tricks/#sec1
RewriteBase /
RewriteCond %{HTTP_USER_AGENT} ^Anarchie [OR]
RewriteCond %{HTTP_USER_AGENT} ^ASPSseek [OR]
RewriteCond %{HTTP_USER_AGENT} ^attach [OR]
RewriteCond %{HTTP_USER_AGENT} ^autoemailspider [OR]
RewriteCond %{HTTP_USER_AGENT} ^Xaldon\ WebSpider [OR]
RewriteCond %{HTTP_USER_AGENT} ^Xenu [OR]
RewriteCond %{HTTP_USER_AGENT} ^Zeus.*Webster [OR]
RewriteCond %{HTTP_USER_AGENT} ^Zeus
RewriteRule ^.* - [F,L]

# Block access to internal code

Options All -Indexes
RewriteEngine On
# Force redirect to strip www from front of domain names
RewriteCond %{HTTP_HOST} ^www\.(.*)$ [NC]
RewriteRule ^(.*)$ http://%1/$1 [R=301,L]
# Allow access to artweb themes and cache, everything is considered
public
RewriteRule ^ciniki-web-layouts/(.*\.)?(css|js|png|eot|ttf|woff|svg)$
ciniki-mods/web/layouts/$1$2 [L]
RewriteRule ^ciniki-web-themes/(.*\.)?(css|js|html|png|jpg)$ ciniki-
mods/web/themes/$1$2 [L]
RewriteRule ^ciniki-web-cache/(.*\.)?(css|js|jpg|gif|png|mp3|ogg|wav)$
ciniki-mods/web/cache/$1$2 [L]
RewriteRule ^ciniki-mail-cache/(.*\.)?(pdf|js|jpg|png|mp3|ogg|wav)$
ciniki-mods/mail/cache/$1$2 [L]
RewriteRule ^ciniki-code/(.*\.)?(zip|ini)$ ciniki-code/$1$2 [L]
```

## Step 7 of 13 - Create the directory structure

The following directories should be created inside your super repo:

...

```
mkdir site
mkdir site/ciniki-backups
mkdir site/ciniki-cache
mkdir site/ciniki-lib
mkdir site/ciniki-mods
mkdir site/ciniki-storage
```

...

The following directories may need their permissions changes so your web server has write permissions to them. If you are running in production these directories should be set to the web server user instead of making them world writable. World writable is easier in a development environment.

...

```
chmod a+w site/ciniki-cache
chmod a+w site/ciniki-storage
```

...

# Creating your own Super Repo

---

## Step 8 of 13 - Add the submodules

You'll need some base ciniki modules for your super repo. The following modules are required unless you are going to build your own versions of them.

...

```
git submodule add git://github.com/ciniki/businesses.git site/ciniki-mods/businesses
git submodule add git://github.com/ciniki/bugs.git site/ciniki-mods/bugs
git submodule add git://github.com/ciniki/core.git site/ciniki-mods/core
git submodule add git://github.com/ciniki/images.git site/ciniki-mods/images
git submodule add git://github.com/ciniki/sysadmin.git site/ciniki-mods/sysadmin
git submodule add git://github.com/ciniki/users.git site/ciniki-mods/users
...
```

Recommended modules:

...

```
git submodule add git://github.com/ciniki/blog.git site/ciniki-mods/blog
git submodule add git://github.com/ciniki/customers.git site/ciniki-mods/customers
git submodule add git://github.com/ciniki/events.git site/ciniki-mods/events
git submodule add git://github.com/ciniki/gallery.git site/ciniki-mods/gallery
git submodule add git://github.com/ciniki/info.git site/ciniki-mods/info
git submodule add git://github.com/ciniki/links.git site/ciniki-mods/links
git submodule add git://github.com/ciniki/mail.git site/ciniki-mods/mail
git submodule add git://github.com/ciniki/subscriptions.git site/ciniki-mods/subscriptions
git submodule add git://github.com/ciniki/web.git site/ciniki-mods/web
...
```

## Step 9 of 13 - Download supporting libraries

The following libraries are required by some modules.

...

```
git clone git://github.com/Synchro/PHPMailer.git site/ciniki-lib/PHPMailer
git clone git://github.com/PHPOffice/PHPWord.git site/ciniki-lib/PHPWord
git clone git://github.com/PHPOffice/PHPExcel.git site/ciniki-lib/PHPExcel
git clone git://github.com/dropbox/dropbox-sdk-php.git site/ciniki-lib/dropbox
...
```

TCPDF will need to be downloaded and extracted into site/ciniki-lib/tcpdf

<http://sourceforge.net/projects/tcpdf/files/>

# Creating your own Super Repo

---

## Step 10 of 13 - Setup the site files

One the modules have been added, their script files need to be linked into the root of the site directory:

^^^

```
cd site
ln -s ciniki-mods/core/scripts/apis.php ciniki-apis.php
ln -s ciniki-mods/core/scripts/json.php ciniki-json.php
ln -s ciniki-mods/core/scripts/login.php ciniki-login.php
ln -s ciniki-mods/core/scripts/manage.php ciniki-manage.php
ln -s ciniki-mods/core/scripts/rest.php ciniki-rest.php
ln -s ciniki-mods/core/scripts/sync.php ciniki-sync.php
ln -s ciniki-mods/web/scripts/index index.php
ln -s ciniki-mods/core/scripts/paypal-ipn.php paypal-ipn.php
```

^^^

## Step 11 of 13 - Setup web module file permissions

The web module requires the web server to have permissions to some files for write.

^^^

```
chmod a+w site/ciniki-mods/web/cache
```

^^^

# Creating your own Super Repo

---

## Step 12 of 13 - Create your commit script

Create the commit.sh script. This script will automatically roll up the submodule commit logs and create a super repo commit message with them. An optional message can be added as an argument.

...

```
#!/bin/sh
#
# This script will commit the submodule changes to the Ciniki super
repo
#
if [ $# -eq 0 ]; then
MSG="Module Updates:
"
else
MSG="$1
Module Updates:
"
fi
if [ $# -gt 1 ]
then
echo "Too many arguments, put your message quotes.\n";
exit;
fi
SUMMARY=`git submodule summary`;
echo "$MSG$SUMMARY"
git commit -am "$MSG$SUMMARY"
...
```

Set the script executable and add the script to git:

...

```
chmod +x commit.sh
git add commit.sh
...
```

## Step 13 of 13 - Commit your changes

Commit your submodule additions and changes to the super repo.

...

```
./commit.sh 'Initial setup of my super repo'
```

...

Now push the changes to github.com

...

```
git push
```

...