# Contributing

Here are the steps for getting setup & started with contributing to Coderwall :

1. Go to [https://assemblymade.com/coderwall](https://assemblymade.com/coderwall) and sign up.

2. Link your GitHub account to your Assembly account in your profile settings.

3. [Fork assemblymade/coderwall](https://github.com/assemblymade/coderwall).

4. Install Virtualbox and Vagrant

5. Prepare your vagrant.yml and .env files

6. Execute `run.sh` (or `run.bat` on Windows)

7. If you have any issues, jump into chat, introduce yourself and ask or leave a message if no one is around.

8. Find an [interesting bounty](https://assemblymade.com/coderwall/wips) on Assembly or suggest a new one.

9. Issue a PR with your work when it is ready for review. (Note: Only PRs from those with valid Assembly account will be merged).

You're on your way to having a stake in Coderwall.

# How to set up the environment

We have videos and text instructions on how to get up and running to develop on Coderwall.

## Check out our video tutorials on how to get started with developing on Coderwall

[![Coderwall New Developer Guides](http://img.youtube.com/vi/OWqTkhbcXUM/0.jpg)](http://www.youtube.com/playlist?list=PLhlPwpqjsgvXK4n8FJBbj7KkvuOw8h3FO)

### How to work on Coderwall using Vagrant on VirtualBox

Sure you could download and install all the dependencies, services, and what not on to your local workstation but ain't nobody got time for that mess.

To save time and headaches we use Vagrant. Vagrant is a automation tool for VirtualBox that will help you spin up a virtual environment with pretty much everything in place and ready for you to start hacking (even on Windows!)

Here's everything you need to get started working on Coderwall with Vagrant TODAY!

\*At the time of writing this document we were using VirtualBox 4.3.20 and Vagrant 1.6.5.\*

\*\*WE ARE USING VIRTUALBOX 4.3.20 DUE TO COMPATABILITY ISSUES WITH VBOX GUEST ADDITIONS.\*\*

#### Vagrant! I already know what to do.

\_\_If you're an experienced Vagrant user then you can fetch the base box and register it yourself.\_\_

There's only a VirtualBox basebox right now.

vagrant box add coderwall\_v3 https://s3.amazonaws.com/coderwall-assets-0/vagrant/coderwall\_v3.box

#### Vagrant? VirtualBox? Let's take this one step at a time.

If you're running Windows, [here's a guide written by one of our members on how to get set up.](https://github.com/assemblymade/coderwall/blob/master/docs/getting\_started\_on\_windows.md)

1. \*\*Install VirtualBox\*\*

Grab the VirtualBox installer from \*\*[here](https://www.virtualbox.org/wiki/Downloads)\*\*.

[![Download the Vbox installer and extensions from here](https://cloud.githubusercontent.com/assets/987305/5224101/af7ed37c-7689-11e4-949b-5116138f49f7.png)](https://www.virtualbox.org/wiki/Downloads)

The \_required\_ version is \*\*VirtualBox 4.3.20.\*\*

I recommend installing VirtualBox 4.3.20 Oracle VM VirtualBox Extension Pack for the extra drivers.

2. \*\*Install Vagrant\*\*

[Vagrant](http://vagrantup.com) is the recommended way to run Coderwall on your own machine. You need to download and install.

Grab the Vagrant installer from \*\*[here](http://www.vagrantup.com/downloads.html)\*\*.

\_At the time of writing this documentation the current version is Vagrant 1.6.5.\_

Follow the installation instructions for your platform on the Vagrant download page.

After installing Vagrant we need to add a couple plugins.

If you're on a OS X/Linux system you can install the plugins by running:

vagrant plugin install vagrant-vbguest

vagrant plugin install vagrant-cachier

The vagrant-vbguest plugin will each help with keeping the VirtualBox Guest Additions up-to-date.

3. \*\*Git assemblymade/coderwall\*\*

[Fork the code](https://github.com/assemblymade/coderwall) if you haven't already done so.

mkdir -p ~/assemblymade

cd ~/assemblymade

Depending on your choice of protocols: \_(this will take a while to run so you may want to grab some coffee)\_

\* git clone https://github.com/your\_username/coderwall.git coderwall

\* git clone git@github.com:your\_username/coderwall.git coderwall

Add upstream:

\* git remote add upstream https://github.com/assemblymade/coderwall.git

\* git remote add upstream git@github.com:assemblymade/coderwall.git

I am going to assume that the project is cloned into your home directory in and into a directory structure like `~/assemblymade/coderwall`.

4. \*\*Fire it up! Fire it up! Fire it up!\*\*

Now that you've got VirtualBox and Vagrant installed with the source code cloned in `~/assemblymade/coderwall` we can start up the Vagrant instance.

cd ~/assemblymade/coderwall

./run.sh # or run.bat if you're on Windows

Since this is probably the first time you're running this command it's going to take a VERY long time (bandwidth willing) to run. This is because Vagrant needs to fetch the Coderwall base box from the Internet and it's about 1.4GB. Fortunately that really only has to be done once (unless the base box get's updated but that's another story).

Once Vagrant reports that you're booted up and ready to go then you'll be able to SSH into the local vm similiar to any other remote box.

# still in ~/assemblymade/coderwall

vagrant ssh

# whoa!

pwd

hostname

ls -al

# We're not in localhost any longer... well, you get what I mean

Now that you're SSH'ed into the Vagrant VM it's time to run the app.

# we're still SSH'ed into Vbox

cd ~/web

rvm current # should be ruby-2.1.2@coderwall

bundle check # should be 'The Gemfile's dependencies are satisfied'

rails s

If all went well the Rails server should start up on PORT 3000.

Now go open your favorite web browser on you host machine and navigate to [http://localhost:3000](http://localhost:3000).

If all goes well (and if it doesn't then check if another app is running on port 3000 and if there's any logging output being displayed in the window you were running Vagrant in) then you're going to be looking at the Coderwall homepage.

Congratulations! NOW GET TO WORK! Enough dilly-dallying with your DEV env.

5. \*\*Hackety Hack!\*\*

Vagrant is powerful not just because it can abstract away the mess of managing a crazy, complicated development environment while ensuring everyone has a common platform to work on. It's also powerful because it will still let you have whatever crazy, complicated editors and tools to work on your code while abstracting away the nasty details of installing and configuring Postgres.

If you're on your host computer and navigate to `~/assemblymade/coderwall` (we're all there, right?) and make changes to your code while Vagrant is running you'll be able to see the changes reflected in Vagrant immediately. Try this.

cd ~/assemblymade/coderwall

echo Hello, `whoami` from `hostname` >> HELLO.txt

vagrant ssh

cd ~/assemblymade

cat HELLO.txt #whoa.

echo Hello, `whoami` from `hostname` >> HELLO.txt

exit

cat HELLO.txt # wah-wah-wee-wa!

rm -f HELLO.txt # yeah, it's deleted in the other spot too.

Yeah, that's pretty awesome. Remember that NFS thing we glossed over earlier? That's the mechanism that Vagrant set up with VirtualBox to transparently synchronize your local folder with the folder on the VM. Now you can edit and manage your files from the comfort of your favorite OS without having to worry about copying them to the VM.

6. \*\*All done for the day - Turning your Vagrant VM off\*\*

When you're ready to call it a day and want to turn the VM off you have two options.

You can either "turn the VM off" using `vagrant halt` or you can suspend the VM using `vagrant suspend`.

Either way when you're ready to resume working just `vagrant up` and you're good to go.

7. Gems Installation and Database Migration

Remember that you are using Vagrant, so if you run ```bundle install``` or ```rake db:migrate``` directly in your terminal it will not affect the virtual machine where Coderwall is running.

In order to run these commands, in the virtual machine, all you have to do is to run ```vagrant provision```.

8. Environment Variables

If you need to change any environment variable you have to edit ```.env``` file properly and restart Rails server running:

vagrant reload ; ./run.sh

9. \*\*Thanks\*\*

I hope you enjoy working with Vagrant as much as we do and feel free to ask questions if you get stuck or have a problem. You're probably not alone and even if you're the first to encounter a rough patch you won't be the last.

## External Dependencies

### Stripe configuration

You'll need to set up a test account with Stripe for local development until this dependency is refactored out of development/test.

See [docs/configuration.md](https://github.com/assemblymade/coderwall/blob/master/docs/configuration.md#stripe) for more.

## Github configuration

You will need a Github application configured for local development until this dependency is refactored out of development/test.

- Create a new GitHub application at [https://github.com/settings/applications/new](https://github.com/settings/applications/new).

- Copy the the ENV variables that you'll need from GitHub.

![The .env will need these values](https://www.evernote.com/shard/s13/sh/3f74a2f7-82d1-46a0-af9c-28f983ad22af/6adc72742c10ddd4ff3c1b711b8d0e27/deep/0/OAuth-Application-Settings.png)

## How to run Coderwall locally.

We use Vagrant to isolate all of our dependencies without polluting your normal working environment.

You're free to not use Vagrant but by the time you're done setting up you'll probably already have given up and installed Vagrant.

## \*\*Protip for Contributors\*\*

When committing a Pull Request for non-application/test code please add [`[skip ci]`](http://docs.travis-ci.com/user/how-to-skip-a-build/) to your commit message.

# Code Conventions and Style Guide

Please refer to the community Ruby & Rails Style Guides created by [bbatsov](https://github.com/bbatsov), author of [Rubocop](https://github.com/bbatsov/rubocop).

[Ruby Style Guide](https://github.com/bbatsov/ruby-style-guide/blob/master/README.md)

[Rails Style Guide](https://github.com/bbatsov/rails-style-guide/blob/master/README.md)