Contribution Guidelines

=======================

Want to get started working on RubyGems.org? Start here!

How To Contribute

-----------------

\* Follow the steps described in [Development Setup](#development-setup)

\* Create a topic branch: `git checkout -b awesome\_feature`

\* Commit your changes

\* Keep up to date: `git fetch && git rebase origin/master`

Once you’re ready:

\* Fork the project on GitHub

\* Add your repository as a remote: `git remote add your\_remote your\_repo`

\* Push up your branch: `git push your\_remote awesome\_feature`

\* Create a Pull Request for the topic branch, asking for review.

If you’re looking for things to hack on, please check

[GitHub Issues](https://github.com/rubygems/rubygems.org/issues). If you’ve

found bugs or have feature ideas don’t be afraid to pipe up and ask the

[mailing list](https://groups.google.com/group/rubygems-org) or IRC channel

(`#rubygems` on `irc.freenode.net`) about them.

Acceptance

----------

\*\*Contributions WILL NOT be accepted without tests.\*\*

If you haven't tested before, start reading up in the `test/` directory to see

what's going on. If you've got good links regarding TDD or testing in general

feel free to add them here!

Branching

---------

For your own development, use the topic branches. Basically, cut each

feature into its own branch and send pull requests based off those.

The master branch is the main production branch. \*\*Always\*\* should be

fast-forwardable.

Development Setup

-----------------

This page is for setting up Rubygems on a local development machine to

contribute patches/fixes/awesome stuff. \*\*If you need to host your own

gem server, please consider checking out

[Gemstash](https://github.com/bundler/gemstash). It's designed to

provide pass-through caching for RubyGems.org, as well as host private

gems for your organization..\*\*

#### Environment (Docker)

There is a `docker-compose.yml` file inside the project that easily lets you spin up

services that the application depends on such as: postgresql, memcached & elasticsearch.

\* Install Docker. See instructions at https://docs.docker.com/engine/installation/

\* run `docker-compose up` to start the required services.

Follow the instructions below on how to install Bundler and setup the database.

#### Environment (OS X)

\* Use Ruby 2.6.x (`.ruby-version` is present and can be used)

\* Use Rubygems 2.6.10

\* Install bundler: `gem install bundler`

\* Install Elastic Search:

\* Pull ElasticSearch `5.6.16` : `docker pull docker.elastic.co/elasticsearch/elasticsearch:5.6.16`

\* Running Elasticsearch from the command line:

```

docker run -p 9200:9200 -e "http.host=0.0.0.0" -e "transport.host=127.0.0.1" -e "xpack.security.enabled=false" docker.elastic.co/elasticsearch/elasticsearch:5.6.16

```

\* Note that `-e "xpack.security.enabled=false"` disables authentication.

\* Install PostgreSQL (>= 9.6.x): `brew install postgres`

\* Setup information: `brew info postgresql`

\* Install memcached: `brew install memcached`

\* Show all memcached options: `memcached -h`

\* Install Google-Chrome: `brew cask install google-chrome`

#### Environment (Linux - Debian/Ubuntu)

\* Use Ruby 2.6.x `apt-get install ruby2.6`

\* Or install via [alternate methods](https://www.ruby-lang.org/en/downloads/)

\* Use Rubygems 2.6.10

\* Install bundler: `gem install bundler`

\* Install Elastic Search:

\* Pull ElasticSearch `5.6.16` : `docker pull docker.elastic.co/elasticsearch/elasticsearch:5.6.16`

\* Running Elasticsearch from the command line:

```

docker run -p 9200:9200 -e "http.host=0.0.0.0" -e "transport.host=127.0.0.1" docker.elastic.co/elasticsearch/elasticsearch:5.6.16

```

\* Install PostgreSQL: `apt-get install postgresql postgresql-server-dev-all`

\* Help to setup database <https://wiki.debian.org/PostgreSql>

\* Install memcached: `apt-get install memcached`

\* Show all memcached options: `memcached -h`

\* Install Google-Chrome:

\* Download latest stable: `wget https://dl.google.com/linux/direct/google-chrome-stable\_current\_amd64.deb`

\* Install chrome: sudo dpkg -i google-chrome-stable\_current\_amd64.deb

#### Getting the code

\* Clone the repo: `git clone git://github.com/rubygems/rubygems.org`

\* Move into your cloned rubygems directory if you haven’t already:

`cd rubygems.org`

\* Install dependencies:

`bundle install`

#### Setting up the database

\* Get set up: `./script/setup`

\* Run the database rake tasks if needed:

`bundle exec rake db:reset db:test:prepare --trace`

\* Set up elasticsearch indexes:

`bundle exec rake environment elasticsearch:import:all DIR=app/models FORCE=y`

#### Running tests

\* Start elastic search: `elasticsearch`

\* Start memcached: `memcached`

\* Run the tests: `bundle exec rake`

#### Confirmation emails links

\* [Account confirmation email](http://localhost:3000/rails/mailers/mailer/email\_confirmation)

\* [A list of all email previews](http://localhost:3000/rails/mailers)

#### Running RuboCop

We use RuboCop to enforce a consistent coding style throughout the project.

Please ensure any changes you make conform to our style standards or else the

build will fail.

bundle exec rake rubocop

If you'd like RuboCop to attempt to automatically fix your style offenses, you

can try running:

bundle exec rake rubocop:auto\_correct

#### Importing gems into the database

\* Import gems into the database with Rake task.

`bundle exec rake gemcutter:import:process vendor/cache`

\* \_To import a small set of gems you can point the import process to any

gems cache directory, like a very small `rvm` gemset for instance, or

specifying `GEM\_PATH/cache` instead of `vendor/cache`.\_

\* If you need the index available then run `bundle exec rake gemcutter:index:update`.

This primes the filesystem gem index for local use.

#### Getting the test data

\* A good way to get some test data is to import from a local gem directory.

`gem env` will tell you where rubygems stores your gems. Run

`bundle exec rake gemcutter:import:process #{INSTALLATION\_DIRECTORY}/cache`

\* If you see "Processing 0 gems" you’ve probably specified the wrong

directory. The proper directory will be full of .gem files.

#### Getting the data dumps

\* You can use rubygems.org data [dumps](https://rubygems.org/pages/data) to test

application in development environment especially for performance related issues.

\* To load the main database dump into Postgres, use `script/load-pg-dump`. e.g.

``` bash

$ ./script/load-pg-dump -d rubygems\_development

```

#### Pushing gems

\* In order to push a gem to your local installation use a command like

the following:

``` bash

RUBYGEMS\_HOST=http://localhost:3000 gem push hola-0.0.3.gem

```

---

When everything is set up, start the web server with `rails server` and browse to

[localhost:3000](http://localhost:3000) or use [Pow](http://pow.cx)!

Database Layout

---------------

Courtesy of [Rails ERD](https://voormedia.github.io/rails-erd/)

![Rubygems.org Domain Model](https://cdn.rawgit.com/rubygems/rubygems.org/master/doc/erd.svg)