Upmin Admin is an open source project and we want to encourage contributions from everyone. In order to make things a little easier, we have outlined some info to try to make it easier for everyone.

## Filing an issue

When filing an issue, please try to mark it appropriately. If you have a question, label it as a question. If it is an bug, please label it as a bug and provide the following:

\* Steps to reproduce the issue.

\* The version of upmin-admin, Rails, Ruby, and any other relevant gems. If possible, a copy of your Gemfile and your Gemfile.lock are useful here.

\* Any relevant stack traces ("Full trace" preferred).

In most cases, this information is enough to determine the cause of your issue and fix it.

Any issue that are open for too long without necessary information will be closed. The issue can be re-opened if the information is provided.

## Pull requests

We love it when users submit pull requests to fix bugs and add features.

Before issuing a pull request, please make sure that all tests are passing, and try to add a few basic tests for whatever feature or bug you are fixing. If backwards compatibility needs to be broken, please provide a good reason for doing so.

Here's a quick guide:

1. Fork the repo.

2. Run the tests. We only take pull requests with passing tests, and it's great

to know that you have a clean slate:

$ bundle install

$ rake # DO NOT USE bundle exec

3. Make your changes.

4. Add a test for your change. If you are refactoring or adding documentation you don't need new tests, but if any bugs are fixed or new features are added you should add some tests.

5. Make sure your tests still pass - see #2 for how.

6. Push to your fork and submit a pull request.

7. Wait for us. We will try to respond within a few days, at which time we will either make suggestions or reject/accept the change.

Some things that will increase the chance that your pull request is accepted,

taken straight from the Ruby on Rails guide:

\* Use Rails idioms and helpers

\* Include tests that fail without your code, and pass with it

\* Update the documentation, the surrounding one, examples elsewhere, guides,

whatever is affected by your contribution

Syntax:

\* Two spaces, no tabs.

\* No trailing whitespace. Blank lines should not have any space.

# New line at the end of the file.

# Try to limit the number of characters on a line to around 80, but this isn't a hard limit.

\* Prefer &&/|| over and/or.

\* `MyClass.my\_method(my\_arg)` not `my\_method( my\_arg )` or my\_method my\_arg.

\* `a = b` and not `a=b`.

\* `a\_method { |block| ... }` and not `a\_method { | block | ... }` or

`a\_method{|block| ...}`.

\* Follow the conventions you see used in the source already.

\* Try to use parenthesis in most cases. This isn't the norm with most Ruby or Rails applications, but we tend to use them so try to as well.

\* Use the keyword `return` when your method's return value is supposed to be used. eg `is\_admin?` would `return false`, but `set\_age(age)` may return the age now, but a future change may cause it to return something else. This helps document when other code should use a return value from a method.

We don't always follow these rules perfectly, so if you see us missing one feel free to make a commit updating it.

If you are using Sublime Text, here are a few preferences that will help with this trimming whitespace, tabs, and creating newlines on files:

```json

{

"ensure\_newline\_at\_eof\_on\_save": true,

"margin": 2,

"tab\_size": 2,

"translate\_tabs\_to\_spaces": true,

"trim\_trailing\_white\_space\_on\_save": true

}

```