Configuring a Rails build

Guide to configuring a Rails project in Jenkins

I had some problems getting Jenkins to play along with the "Jenkins ruby metrics plugin", so I wrote this guide to document my setup. Hope it helps someone else out.

Based loosely on: http://www.juretta.com/log/2008/11/11/hudson_test_drive_part_1_rails/

Notice that Saikuro reports aren't currently working.

Prerequisites

Jenkins needs to have the following plugins installed: "Jenkins Ruby Plugin", "Jenkins ruby metrics plugin". Also, the build server needs Ruby and rubygems.

The build process will take care of installing any gems, as long as they are in "config/environment.rb".

Application

edit Rakefile and add the following:

```ruby
require 'rubygems'
# Install:
#  sudo gem install ci_reporter
gem 'ci_reporter'
require 'ci/reporter/rake/test_unit' # use this if you're using Test::Unit
```

Install the rcov-plugin, by running:

```bash
./script/plugin install http://svn.codahale.com/rails_rcov
```

Add a dummy test case, to please rcov. Put the following in "test/unit/dummy_test.rb", "test/functional/dummy_test.rb" and "test/integration/dummy_test.rb":

```ruby
# This is because rcov chokes, when there are no test cases found.
require 'test_helper'

class DummyTest < ActiveSupport::TestCase
  def test_truth
    assert true
  end
end
```

(Once you add real test cases to the project, you can remove these again)

Unless you require a MySql database, you can set the test database to sqlite. Otherwise, you'll have to create a test database somewhere, and put credentials in "database.yml". Remember to add sqlite as a dependency to "config/environment.rb", if you use it.

Since Rake can't always install missing gems through the "rake gems:install" task, we'll have to put the install the following plugin:

```bash
./script/plugin install http://github.com/troelskn/gemconf_plugin.git
```

SVN add new files and commit the changes.
Jenkins

Select “New Job”

Check option “Build a free-style software project” and enter a name. The name should be CamelCaseWithoutSpaces

Under “Source Code Management”, select “SVN”

Enter:

Repository URL: http://blah/project/trunk
Local module directory (optional): trunk

Under “Build Triggers”, select “Poll SCM”

Enter:

# Poll every 5 minutes
*/5 * * * *

Select “Add Build Step” -> “Execute shell”

Enter:

# enter application root
cd trunk
# install missing gems
script/install_gems
# set env vars
export GEM_PATH=/usr/local/lib/ruby/gems/1.8/gems
export CI_REPORTS=results
export RAILS_ENV=test
# Prepare for rcov
[ -d "coverage" ] && rm -rf coverage
mkdir coverage
# invoke rake
rake db:create db:migrate ci:setup:testunit test:test:rcov --trace RCOV_PARAMS="-I/usr/local/lib/ruby/gems/1.8/gems --aggregate coverage/aggregate.data"

(Note that if your GEM_PATH differs, you'll have to adjust it in the above)

If you'd prefer to only make coverage reports for unit tests, you can replace “test:test:rcov” with “test:units:rcov test”

Under “Post-build Actions”, select “Publish JUnit test result report”, “Publish Rcov report” and “Publish Rails stats report”

In “Test report XMLs”, enter “trunk/results/*.xml”

In “Rcov report directory”, enter “coverage/units”

Click “Build Now” to test the configuration.