ChiliProject - Task # 57: Setup a local clone of the official git repository

Status:	Closed	**Priority:**	Normal
Author:	Eric Davis	**Category:**	ChiliProject - Organization
Created:	2011-01-12	**Assignee:**	Eric Davis
Updated:	2011-02-02	**Due date:**	
Remote issue URL:			
Description:	I'd say it should poll the official repository every 30-60 minutes and then run the @fetch_changesets@ to load them into the app.		

Associated revisions

2011-05-19 11:28 pm - Felix SchÃ¤fer

Merge pull request #57 from meineerde/issues/unstable/112-database-detection-library

Database detection library. #112

History

2011-01-13 12:40 am - Felix Schäfer

I have some crons and post-receive hooks to fetch from/push to github and update Redmine in the process, do you already have something or would you like to have them?

2011-01-16 10:54 am - Holger Just

I'd vote to pull the repo more often, say every 10 minutes. I find it very painful on redmine.org to see the code with such great delay. At best we should get rid of any polling and use some sort of hook (which is also possible with github) though.

2011-01-16 01:16 pm - Eric Davis

Felix: All of my scripts are custom to my server so if you have something you can share that would be great.

Holger: I'd say every 30 minutes in the short term. At every 10 minutes you run the risk of loading multiple tasks and killing the server.

Github has a github services repository where we can add our own github-hook. Then we just need to add an endpoint on our side to receive that post. It's not too much work and I think there are a few github/redmine integration plugins that we can use or build on top of.

2011-01-16 02:55 pm - Felix Schäfer

Eric Davis wrote:
> Felix: All of my scripts are custom to my server so if you have something you can share that would be great.

So this is what I have in the @post-receive@:

<pre><code>#!/bin/sh

export REPOS_PATH="$PWD"

/some/path/scm_post_hook.sh</code></pre>

And @scm_post_hook.sh@ (redacted a bit, holds some branches depending on git/svn calling it) holds:

<pre><code>#!/bin/bash

Initialization stuff
declare -A MIRROR_TO_GITHUB

User settings

# Base URI of your Redmine
REDMINE_URI="some_URI"
# The API key for the Redmine repository WS
REDMINE_API_KEY="some_key"
# List of Repositories to push to github
MIRROR_TO_GITHUB=(["redmine_doodles"]=1 ["redmine_ical"]=1) # Format: (["reponame1"]=1 ["reponame2"]=1 ["reponame3"]=1)

Computed settings
# Expects REPOS_PATH to be of the form /var/REPOS_SCM/VHOST_NAME/REPOS_NAME
REPOS_NAME=$(echo $REPOS_PATH | awk -F '/' '{print $5;}')

Actual actions
Trigger Redmine to fetch the changes for the repository's project
curl -s -o /dev/null "${REDMINE_URI}/sys/fetch_changesets?key=${REDMINE_API_KEY}&id=${REPOS_NAME}" &

# Push to github
if [[-n ${MIRROR_TO_GITHUB[${REPOS_NAME}]}]]; then
  git --git-dir=${REPOS_PATH} push --quiet github &
fi</code></pre>

This assumes all repos live in the same directory, or at least in the same depth, and that there is a @github@ remote configured in the repo as a mirror.

The crontab line for fetching is:

<pre><code>*/30 * * * * some_user cd /some/path/to/a/git-repo && git fetch --quiet github && hooks/post-receive</code></pre>

There's probably nicer/better ways, has worked fine for me for quite some time though.

2011-02-01 07:23 am - Eric Davis
Felix:

It looks like your code does push and pulls. So if you push into the private repo it would update ChiliProject and then push the changes up to github. And it also has a cronjob to pull data down from github.

Maybe we could just have a cronjob that does a git pull on the local repo, when it pulls data it triggers a post commit hook to load the new commits into ChiliProject.

2011-02-01 12:56 pm - Felix Schäfer
Eric Davis wrote:
> It looks like your code does push and pulls. So if you push into the private repo it would update ChiliProject and then push the changes up to github. And it also has a cronjob to pull data down from github.

The @post-receive@ hook pings redmine to fetch changesets for that repo, and pushes to github _if_ the project is in the @MIRROR_TO_GITHUB@ list, so pushing to github is optional on a per project/repo basis.

> Maybe we could just have a cronjob that does a git pull on the local repo, when it pulls data it triggers a post commit hook to load the new commits into ChiliProject.

That's what the crontab line does :-)

2011-02-01 10:49 pm - Felix Schäfer
- Status set to Open

2011-02-02 03:58 am - Eric Davis

- Assignee changed from Muntek Singh to Eric Davis

Going to get this setup for the launch.

2011-02-02 04:53 am - Eric Davis

I've setup a moonshine recipe to manage this for us now. I'll need to switch the cloning repo once our repo is setup (#29) but it's working good so far.

Code: https://gist.github.com/9278e19f8e28b05a831b

h3. Documentation

It installs curl which is used by the git hook.

It makes sure there is a @/srv/git_mirrors@ directory for our user account. This is so if we want to have multiple mirrors, for example official plugins. (@file "/srv/git_mirrors"@)

It then runs the git clone command (@exec('git-mirror-chiliproject')@) but only if the target doesn't exist (@creates => target_path@).

Then it sets up some variables and creates the script post-receive hook as a string (@update_repo_view_hook@). This is written out to the disk with a mode of 755 so it can be executed.

Finally a cronjob is added to run the git fetch and post-receive hook every 15 minutes (@cron@).

The actual git hook uses curl and the SysController to fetch the changesets for our project.

So overall, the code flow would be:

We push to github
Within 15 minutes, the cron kicks off
cron triggers the git fetch which loads the new changesets
Then the post-receive hook is called
Which hits the SysController, which loads the new changesets

We should be able to reuse this code on other mirror too, it's pretty simple.

2011-02-02 04:17 pm - Eric Davis

- Status changed from Open to Closed

Updated the config to use the new official repo.