

Infrastructure - GithubRedmineRepoSync - # 2

This is also available as [Redmine HowTo](#)

Keeping Redmine repository in sync with Github without dedicated plugin (Apache CGI + Github Webhook)

This is a solution in case you don't want to install additional plugins just to keep repository synchronised. It requires you to have Apache webserver with access to repository you are trying to sync. Apache has to support running CGI scripts.

Clone Github repository

Clone repository and make sure it is accessible by webserver:

```
mkdir /var/lib/redmine/repo
chown apache /var/lib/redmine/repo
su -u apache git -C /var/lib/redmine/repo clone https://github.com/username/repo_name.git
```

Enable WS for repository management in Redmine

Go to <https://your.redmine.com/settings?tab=repositories> and:

- select: *Enable WS for repository management*
- generate a repository management WS API key and save it for next step

Prepare CGI script

Any script you run on your server will do. Below is an example of Bash script that pulls git repository and notifies Redmine to fetch changesets (substitute <repository-api-key> with your own):

```
#!/bin/sh
# Requires: jq

REPO_PATH='/var/lib/redmine/repo'

# Empty stdin, Apache issue https://bz.apache.org/bugzilla/show_bug.cgi?id=44782
REPO_NAME=$(cat <&0 | jq '.repository.name' | tr -cd 'A-Za-z0-9_-')
if [ -z "${REPO_NAME}" ] || [ ! -d "${REPO_PATH}/${REPO_NAME}" ]; then
    echo "Status: 400 Bad Request"
    echo "Content-Type: text/plain; charset=utf-8"
    echo
    echo "project: unrecognized"
    exit 0
fi

/usr/bin/git -C "${REPO_PATH}/${REPO_NAME}" pull -n -q
result1=$?

PROJECT_NAME=$(echo "${REPO_NAME}" | tr '_' '-')
/usr/bin/curl --max-time 60 -s "https://your.redmine.com/sys/fetch_changesets?id=${PROJECT_NAME}&key=<repository-api-key>" >/dev/null
result2=$?

if [[ $result1 && $result2 ]]; then
    echo "Status: 200 OK"
else
    echo "Status: 500 Internal Server Error"
fi

echo "Content-Type: text/plain; charset=utf-8"
echo
echo "project: ${PROJECT_NAME}"
```

```
if [[ $result1 ]]; then
    echo "git pull: ok"
else
    echo "git pull: failed"
fi

if [[ $result2 ]]; then
    echo "fetch changesets: ok"
else
    echo "fetch changesets: failed"
fi
```

Let's say you save this script under: `/var/www/cgi-bin/update-repo.cgi`

You can test if script executes properly:

```
echo <copy-input-from-github-webhook-request> | sudo -u apache /var/www/cgi-bin/update-repo.cgi
```

Configure Apache to run script whenever particular URL is requested

Inside VirtualHost of your choice just add:

```
...
# Github webhook for repository pull/update
ScriptAlias /update-repo.cgi /var/www/cgi-bin/update-repo.cgi
<Directory /var/www/cgi-bin/>
    Options ExecCGI
    AllowOverride None
    Require all granted
</Directory>
...
```

In case you use the same VirtualHost to proxy requests to your Redmine rails server, you should exclude your special URL from being proxied with:

```
ProxyPass /update-repo.cgi !
```

Configure Github Webhook

Go to your Github repository page, choose *Settings* -> *Webhooks* -> *Add webhook*. Then set:

- Payload URL: `https://your.virtualhost.com/update-repo.cgi`
- Content type: `application/json`
- Which events would you like to trigger this webhook?: Just the push event.
- Active: yes

Update webhook and you're done.