

Quelques utilitaires pour aller avec Request Tracker

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 3 décembre 2007

<https://www.bortzmeyer.org/utilitaires-request-tracker.html>

Le logiciel Request Tracker est le plus répandu aujourd'hui des logiciels de gestion de tâches. Il sert pour les centres d'assistance aux utilisateurs, pour garder trace des demandes, aux développeurs pour se souvenir des bogues et autres demandes, à tous pour suivre les tâches en cours. Je présente ici quelques utilitaires que j'ai développés les différentes fois où j'ai utilisé Request Tracker.

Request Tracker est un logiciel libre et, en outre, il est très extensible. On peut modifier son comportement de plusieurs façons, en changeant les pages Web <<http://wiki.bestpractical.com/view/CleanlyCustomizeRT>>, en développant des scripts <<http://wiki.bestpractical.com/view/Scrip>> (non, il n'y a pas de fautes de frappe), en attaquant directement la base de données en SQL, etc.

Ayant utilisé à plusieurs reprises Request Tracker, dans des environnements très différents, j'ai développé plusieurs utilitaires que je présente ici un peu en vrac. Je ne garantis pas leur bon fonctionnement, d'autant plus que certains avaient été développés pour de vieilles versions et pas tenus à jour depuis. Utilisez-les donc comme source d'inspiration mais ne leur confiez pas vos vies aveuglément. (Il en existe beaucoup d'autres <<http://wiki.bestpractical.com/view/Contributions>>.)

rt2dot.py (en ligne sur <https://www.bortzmeyer.org/files/rt2dot.py>) produit, via Graphviz un graphique des tâches en cours et de leurs relations (puisque Request Tracker permet d'enregistrer les relations entre les tâches, #56 dépend de #13, #566 a un rapport avec #1922, etc). Il est écrit en Python et fonctionne en lisant directement la base de données (le schéma simple et bien documenté <<http://wiki.bestpractical.com/view/DBSchema>> de Request Tracker rend cela très facile, comme cela devrait l'être avec toutes les bases relationnelles).

rt2gnuplot.py (en ligne sur <https://www.bortzmeyer.org/files/rt2gnuplot.py>) produit, via Gnuplot, un graphique de la quantité de tickets ouverts et fermés pendant une période donnée. Attention, comme avec toutes les métriques du travail, à ne pas le prendre trop au sérieux, ou bien les employés vont commencer à fermer les tickets pas finis pour augmenter leur score! Il est écrit en Python et fonctionne en lisant directement la base de données. Son fonctionnement est un peu compliqué car il utilise Cheetah pour créer le fichier de commandes de gnuplot. Le gabarit de ce fichier est :

```
set terminal png
set xlabel "$period from now ($now)"
set ylabel "Tickets"
set title "Activity of the Request Tracker ticketing system of CODEV-NIC"
plot "rt.dat" using 1:2 with lines title "Tickets opened", \
      "rt.dat" using 1:3 with lines title "Tickets closed"
```

et, pour le lancer, j'utilise make avec ce Makefile (le fichier `rt.pickle` est créé par `rt2gnuplot.py`):

```
rt.pickle: rt.dat

rt.png: rt.dat rt.gp
      gnuplot rt.gp > $@

rt.dat: svncommits2gnuplot.py
      ./rt2gnuplot.py > $@

rt.gp: rt.gp_tmpl rt.pickle
      cheetah fill --pickle rt.pickle --iext gp_tmpl --oext gp rt.gp_tmpl

# The real dependency is to the RT system
.PHONY: rt.dat
```

`SomeQueues` (en ligne sur <https://www.bortzmeyer.org/files/SomeQueues.tar.gz>) est un vieux scrip (plus précisément une `ScripCondition` <<http://wiki.bestpractical.com/view/Condition>>) pour tester si une transaction appartient à un ensemble de queues, ensemble décrit par une expression rationnelle. Comme tous les scrips, et comme Request Tracker lui-même, il est écrit en Perl.

Pour exporter automatiquement la liste des tickets ouverts vers une machine qui n'est pas connectée en permanence, j'utilise simplement, depuis cron, la ligne de commande de Request Tracker <<http://wiki.bestpractical.com/view/UseRtTool>>, moins susceptible de subir des changements que le schéma de la base de données :

```
rt ls -t ticket -l -o +Created "(Status='new' or Status='open')" | \
      mutt -e 'set charset=utf-8' -s "Tickets RT" \
          stephane@monportable.internatif.org
```

`rt-summary.py` (en ligne sur <https://www.bortzmeyer.org/files/rt-summary.py>) est un script qui envoie par courrier tous les matins un rappel des tâches les plus importantes, classées par priorité décroissante. Il lit dans la base la liste des utilisateurs Request Tracker et envoie à chaque utilisateur un bref rappel des tickets dont il est responsable. Il est écrit en Python et fonctionne en lisant directement la base de données.