

TRAC/SVN (Projet Pydza)

Vous vous apprêtez à travailler dans un projet, et ce projet utilise TRAC et Subversion (SVN).

Mais qu'est ce que TRAC/SVN ?

TRAC/SVN correspond en fait à deux applications exécutées sur un serveur Linux.

Nous allons faire une présentation des deux et voir comment ils fonctionnent.

TRAC

TRAC correspond à un site web accessible à cette adresse : <http://pydza.labo-linux.org/trac>. Celui-ci va permettre de voir ce qu'il faut faire pour mener à bien le projet, qui va faire quoi, et de voir aussi l'avancement du projet.

Certaines parties de TRAC sont accessibles au public, d'autres parties nécessitent d'avoir un identifiant et un mot de passe.

Commençons par voir la chose la plus importante de TRAC : les **tickets**.

Un ticket, c'est en fait une action à faire. Ça peut par exemple être : *Créer un monstre volant*.

Un ticket se compose des choses suivantes :

- Un numéro unique
- Un titre, on vient de le voir
- Une description, pour expliquer clairement ce qu'il faut faire
- Un component, c'est en fait un moyen de classer les tickets. On peut avoir des components comme : *programmation, graphismes ...*
- La personne qui a créé le ticket
- Une priorité, qui permet de voir quels tickets sont à faire en premier
- Une milestone, ça représente une «échéance». Par exemple une version alpha du jeu.
- Une version du ticket. Correspond à la version du projet auquel se rapporte le ticket.
- La personne à qui le ticket est assigné (qui a donc choisi de le faire). Un ticket est actif par défaut. Une fois que le but du ticket est atteint on peut le fermer.

Maintenant que nous avons vu ça, nous pouvons faire une présentation du site :

En haut de la page vous avez un lien pour vous connecter, et en-dessous différents onglets (selon le cas vous pouvez en voir plus ou moins, entre autre si vous n'êtes pas connecter).

L'onglet Wiki : correspond à la page principale, peut contenir une description du projet.

L'onglet Timeline : nous verrons plus tard ce qu'il contient

L'onglet Roadmap : vous voyez ici les différentes milestone, un avancement ainsi que le nombre de tickets fermés ou actifs.

L'onglet Browse Source : nous verrons aussi plus tard ce qu'il contient.

L'onglet View Ticket : très important. C'est ici que nous pouvons voir les différents tickets. Lorsque vous cliquez dessus vous avez plusieurs choix, prenez le premier : *Active Tickets*. Vous voyez ainsi tous les tickets actifs.

Sont affichées aussi toutes les infos vues précédemment. Lorsque vous cliquez sur un ticket vous pouvez : ajouter des commentaires, changer la description, la priorité, le component, la milestone et la version.

Plus bas vous pouvez accepter le ticket, ainsi ce sera votre nom qui sera écrit dans *owner*. Vous pouvez aussi fermer le ticket en cochant *resolve as : fixed*.

Dans l'ordre, quand on veut participer au projet :

1. On choisi un ticket sur lequel on veut travailler
2. On regarde si quelqu'un n'a pas déjà accepté le ticket (colonne *owner*), si c'est le cas vous pouvez demander à la personne si elle bosse actuellement dessus ou si vous pouvez le prendre.
3. Si tout est bon on clique sur le ticket, et on choisi *accept ticket*, puis *Submit changes*.
4. Une fois que vous avez fait tout ce qui concerne ce ticket il ne faut pas oublier de le fermer (*resolve as : fixed*).

L'onglet New Ticket : permet de créer un nouveau ticket.

SVN

Passons maintenant à SVN. SVN va nous permettre de mettre en commun le résultat de notre travail.

En fait, un serveur va stocker les fichiers du projet.

Une personne qui rejoint le projet va faire ce que l'on appelle un *checkout*, ainsi elle aura une copie locale du projet sur son ordinateur.

Ensuite la personne va travailler dessus, puis quand elle s'arrête elle fera un *commit*, c'est-à-dire qu'elle enverra ses modifications au serveur.

Ainsi, la prochaine personne qui fera un *update* récupérera directement les nouvelles modifs.

Généralement, il est donc très important de faire un *update* avant de commencer à travailler, puis un *commit* quand on s'arrête.

En revanche si le *commit* que vous allez faire empêche le fonctionnement du jeu mieux vaut ne pas en faire.

Chaque fois que l'on fait un *commit*, cela crée une nouvelle *révision*.

A savoir : il est tout fait possible de revenir à une révision précédente, c'est l'un des intérêts de SVN.

Retournons sur TRAC, pour les deux onglets que nous n'avons pas vu :

L'onglet Timeline : ici vous voyez toutes les modifications liées aux fichiers, et aussi certaines infos des tickets. Vous pouvez voir quand un ticket est créé, ou fermé. Vous voyez aussi les nouvelles révisions, par exemple : *Changeset [88] by jonass* signifie que jonass a fait un *commit*. Celui-ci correspond à la *révision 88*. Vous voyez la date et un message descriptif. Si vous cliquez sur le numéro de la révision, vous voyez les modifs qu'elle apporte.

L'onglet Browse Source : celui-ci vous permet de voir les fichiers du projet contenu sur le serveur. Vous pouvez voir de quand date la dernière modif, qui l'a fait et à quelle révision ça correspond.

Petite info : dans les descriptions et commentaires de ticket, si vous voulez parler du ticket 3, ne mettez pas juste 3, mais mettez #3, ainsi cela fera un lien vers le ticket 3. De même si vous voulez parler d'une révision, ne mettez pas 88, mais r88, cela fera aussi un lien.

Il ne nous reste plus qu'à voir comment faire un *checkout* et un *commit* !

D'abord il faut installer le paquet *subversion*. Donc `urpmi subversion` ou `apt-get install subversion` (ou autre) selon votre distribution.

Dans cet exemple nous allons faire un *checkout* dans un dossier `pydza` qui sera dans notre dossier `~/Projets` :

```
cd ~/Projets
svn checkout http://pydza.labo-linux.org/svn/ --username login pydza/
```

En remplaçant `login` par votre `login`.

On peut ensuite travailler sur le projet.

Lorsque vous voulez mettre à jour (pour voir si des modifications ont été fait) la commande est :

```
svn update (ou svn up)
```

Viens le moment de faire son *commit* pour envoyer les modifs au serveur, la commande est :

```
svn commit
```

Si vous avez fait des modifications un éditeur de texte en console va s'ouvrir.

A cet endroit vous pouvez taper une description pour indiquer les modifications que vous avez fait.

Il est primordial de toujours mettre une description.

Vous pouvez ensuite sauvegarder et quitter l'éditeur de texte et le `commit` se fera.

Idéalement vous pouvez même faire un `update` juste avant de `commit`, et si jamais il y a eu du changement vérifier que tout marche encore avec vos modifs.

Vous pouvez aussi utiliser « `svn stat` » pour voir les modifications locales que vous avez.

Attention : quand vous utilisez `svn stat`, `up` ou `commit` il ne se produira pas la même chose selon le répertoire dans lequel vous êtes.

Si vous êtes par exemple dans `/dossierpersonnel/pydza/` et que vous faites un `svn update` celui-ci se fera dans tous les sous-repertoires. De même pour un `commit`.

En revanche si vous faites un `svn commit` en étant dans le dossier `/dossierpersonnel/pydza/trunk/media/`, `svn` ne commitera que les fichiers modifiés dans ce dossier et ces sous-dossiers.

Ne pas oublier : un *update* avant de commencer à travailler, un *commit* après.

Cela permet de partager le travail, de voir l'avancement et de faire une sauvegarde !

Si vous voulez rajouter un fichier (un nouveau niveau ou n'importe quoi), il ne suffit pas de le créer dans le dossier ou dans un de ses sous-dossiers, il faut aussi le dire à SVN, pour qu'il l'envoie sur le serveur !

Pour cela il suffit de faire `svn add` puis le nom du fichier ou du dossier.

`svn add` n'envoie pas le nouveau fichier, il dit juste que lors du prochain `commit` il y aura un nouveau fichier à envoyer.

Pour voir le reste des commandes vous pouvez faire `svn help` ou faire un tour sur internet.

Voilà pour la présentation de TRAC et SVN !