



INTRODUCTION À SUBVERSION ET TRAC

Cyrille Baudouin (baudouin@enib.fr)

Gestion de sources, contrôle de versions et gestion de projet.

INTRODUCTION

SUBVERSION (SVN)

- Serveur Linux (ssh + apache/HTTP)
- Sauvegarde des fichiers sources
- Sauvegarde itératives des versions

TRAC

- Serveur Linux (apache/python/HTTP)
- Gestion de projet

INTRODUCTION

SUBVERSION (SVN)

- Serveur Linux (ssh + apache/HTTP)
- Sauvegarde des fichiers sources
- Sauvegarde itératives des versions

TRAC

- Serveur Linux (apache/python/HTTP)
- Gestion de projet

SUBVERSION

GESTION DE SOURCES

- Sauvegarde des sources courantes : “trunk”
- Developpements parallèles : “branches”
- Identification des livrables (releases) : “tags”
- Vue locale = sources sur le serveur + sources et modifications locales

CONTRÔLE DE VERSIONS ET TRACABILITÉ

- Modifications horodatées et identifiées par utilisateur
- Possibilité de revenir (localement) à une ancienne version
- “Mélange” de versions

SUBVERSION

GESTION DE SOURCES

- Sauvegarde des sources courantes : “trunk”
- Developpements parallèles : “branches”
- Identification des livrables (releases) : “tags”
- Vue locale = sources sur le serveur + sources et modifications locales

CONTRÔLE DE VERSIONS ET TRACABILITÉ

- Modifications horodatées et identifiées par utilisateur
- Possibilité de revenir (localement) à une ancienne version
- “Mélange” de versions

A QUOI SERT SVN ?

GESTION DES SOURCES

- Sauvegarde des sources courantes : “trunk”
- Developpements parallèles : “branches”
- Identification des livrables (releases) : “tags”
- Vue locale = sources sur le serveur + sources et modifications locales

CONTRÔLE DES VERSIONS ET TRACABILITÉ

- Modifications horodatées et identifiées par utilisateur
- Possibilité de revenir (localement) à une ancienne version
- “Mélange” de versions

A QUOI SERT SVN ?

GESTION DES SOURCES

- Sauvegarde des sources courantes : “trunk”
- Developpements parallèles : “branches”
- Identification des livrables (releases) : “tags”
- Vue locale = sources sur le serveur + sources et modifications locales

CONTRÔLE DES VERSIONS ET TRACABILITÉ

- Modifications horodatées et identifiées par utilisateur
- Possibilité de revenir (localement) à une ancienne version
- “Mélange” de versions

COMMENT ÇA MARCHE ?

COTÉ SERVEUR

- Sauvegarde itérative et encodé des données sources
- TOUTES les modifications sont enregistrées, et ne peuvent être modifiées.

COTÉ CLIENT

- Une vue locale “instantanée” des données
- client console “svn” + commande svn ou commande unix
- client graphique

COMMENT ÇA MARCHE ?

COTÉ SERVEUR

- Sauvegarde itérative et encodé des données sources
- TOUTES les modifications sont enregistrées, et ne peuvent être modifiées.

COTÉ CLIENT

- Une vue locale “instantanée” des données
- client console “svn” + commande svn ou commande unix
- client graphique

A QUOI SERT TRAC ?

GESTION DE PROJET

- Partage d'information : "wiki"
- Organisation des composants : "components"
- Organisation des jalons : "milestones"
- Gestion des défauts (bugs) et attribution de tâches : "tickets"
- Accès aux sources svn (toutes les versions) : "browser"
- Visualisation et suivi des modifications (wiki, sources, tickets...)

COMMENT ÇA MARCHE ?

COTÉ SERVEUR

- Serveur web + moteur TRAC (python)
- Répertoire de travail isolé pour chaque projet (css et html séparés)

COTÉ CLIENT

- Un navigateur internet !
- adresse interne : <http://svn.cerv.fr/>
- adresse externe : <http://svn-cerv.ath.cx/>

COMMENT ÇA MARCHE ?

COTÉ SERVEUR

- Serveur web + moteur TRAC (python)
- Répertoire de travail isolé pour chaque projet (css et html séparés)

COTÉ CLIENT

- Un navigateur internet !
- adresse interne : `http://svn.cerv.fr/`
- adresse externe : `http://svn-cerv.ath.cx/`