

Quickstart Forge Clermont Université

Contexte

Forge Clermont Université est un outil de gestion de développement collaboratif de logiciel. Il est basé sur le logiciel libre Redmine et utilise par défaut Subversion comme outil de gestion de version.

URL : <http://forge.clermont-universite.fr>

Structure

Contexte.....	1
Connexion à la forge.....	1
Gestion d'un projet.....	2
Ajouter un membre.....	2
Gestion des demandes (tickets).....	3
Démarrage rapide avec Subversion.....	3
Organisation d'un dépôt.....	3
Initier un projet (import).....	3
Récupérer un projet (checkout).....	3
Effectuer une révision du projet (commit).....	4
Mettre à jour son répertoire local de travail (update).....	4
Ressources.....	4

Connexion à la forge

Un lien 'Connexion' situé en haut à droite de la page permet d'accéder à la page de connexion de la forge.

La forge est reliée au service d'authentification (Single Sign On) des établissements lorsque cela est possible. Pour l'instant, cette fonctionnalité est disponible pour :

- L'Université d'Auvergne, Clermont-Ferrand I
- L'Université Blaise Pascal, Clermont-Ferrand II

Accueil Projets Aide Connexion

Forge Clermont Université

Recherche:

Membre Université Blaise Pascal



Université Blaise Pascal

Connexion

Membre Université d'Auvergne



Université d'Auvergne

Connexion

Membre extérieur

Identifiant:

Mot de passe:

[Créer un compte invité](#) [Mot de passe perdu](#)

Illustration 1: Page de connexion à la forge.

Ainsi, les membres (étudiants et personnels) de ces établissements sont automatiquement reconnus en utilisant le service d'authentification. A la première connexion à la forge, le compte est automatiquement activé. Sur la page de connexion, il faut utiliser les liens qui redirigent vers les services d'authentification ad hoc (cf. 1).

Attention : seul le personnel des établissements précités a le droit de créer un nouveau projet dans la forge. Les autres utilisateurs doivent se référer soit à un membre du personnel soit à l'administrateur pour la création d'un nouveau projet.

Pour les membres extérieurs à ces établissements, il faut demander la création d'un compte invité sur la forge et ensuite utiliser le formulaire de connexion réservé aux membres extérieurs.

Gestion d'un projet

Un projet est un projet de développement informatique auquel participe un ensemble de membres. Ceux-ci doivent avoir un compte actif sur la forge.

La forge propose un certain nombre de modules pour un projet :

- Gestion et suivi des demandes
- Publication d'annonces
- Publication de documents et de fichiers
- Wiki
- Dépôt de sources
- Forums de discussions

Un projet peut être soit public soit privé. Un projet privé est visible uniquement par les membres du projet tandis qu'un projet public est visible par tous, même par les utilisateurs non enregistrés. Dans un projet public, un utilisateur inscrit sur la forge mais non membre du projet peut aussi participer aux forums de discussion et faire de nouvelles demandes (rapport de bug, demande de support de nouvelle fonctionnalité).

A chaque membre est associé un rôle au sein du projet :

- **Manager** : un *manager* a les droits d'administration sur le projet, il choisit notamment les membres du projet, leurs rôles et les modules actifs pour le projet. Il peut y avoir plusieurs *manager* par projet.
- **Developer** : un *developer* est un membre qui participe activement au développement du projet. Il a ainsi accès en écriture aux sources (via Subversion), peut créer de nouvelles demandes (rapport de bug ou demandes de nouvelles fonctionnalités), participer aux forums, modifier le wiki.
- **Reporter** : un *reporter* est un membre qui ne développe pas sur le projet mais qui peut accéder en lecture aux sources, participer aux forums de discussion et créer de nouvelles demandes.

Ajouter un membre

Pour ajouter un membre à un projet, il faut que l'utilisateur soit connu de la forge, c'est-à-dire qu'il se soit connecté au moins une fois. Ensuite, le *manager* doit aller sur la page 'Configuration > Membres' de son projet. Dans le champ texte, il faut saisir les premières lettres du prénom, du nom ou de l'identifiant de l'utilisateur et sélectionner l'utilisateur dans la liste proposée.



Illustration 2: Ajout d'un nouveau membre

Gestion des demandes (tickets)

Une *demande* est une demande d'assistance (un « ticket ») auprès de l'équipe de développement. Forge Clermont Université permet de mettre en place un véritable suivi des demandes en associant à chaque demande un historique des commentaires des utilisateurs et des développeurs, et les révisions du code associées à la demande. Le gestion des tickets est un élément essentiel du développement collaboratif car il permet la remontée des demandes de la part des utilisateurs auprès des développeurs, le suivi et la classification de ces demandes et l'organisation des développeurs pour répondre à la demande (une demande peut être assignée à un développeur). Par ailleurs, l'ensemble des demandes constitue petit à petit une base de connaissance sur le projet.

Forge Clermont Université distingue 3 types de demandes :

- **Bug** : un rapport de *bug* permettant de remonter auprès des développeurs les anomalies et les incidents détectés lors du fonctionnement du logiciel.
- **Feature** : il s'agit d'une demande d'une nouvelle fonctionnalité à ajouter au logiciel.
- **Support** : il s'agit d'une demande d'aide d'un utilisateur concernant, par exemple, le déploiement ou l'utilisation du logiciel.

Un manager d'un projet peut choisir d'activer ou de désactiver la gestion d'un type de demande dans le cadre 'Trackers' de l'onglet *Configuration/Informations*.

Chaque demande a une priorité selon la gravité ou l'urgence de la demande. Les 5 niveaux de priorité sont, dans l'ordre croissant : **Low, Normal, High, Urgent, Immediate**.

Le statut d'une demande permet de formaliser le cycle de vie d'une demande au sein d'un projet. Les différents statuts sont : **New** (à la création de la demande), **Assigned** (la demande a été assignée à un développeur), **Resolved** (la demande a été traitée), **Feedback** (la solution est en attente de validation), **Closed** (la demande est définitivement clôturée), **Rejected** (la demande a été rejetée).

Enfin, pour des projets relativement importants, il est possible d'organiser les demandes dans des catégories. Ces catégories sont spécifiques à chaque projet et peuvent être définies par un manager du projet dans l'onglet *Configuration/Catégorie de demandes*.

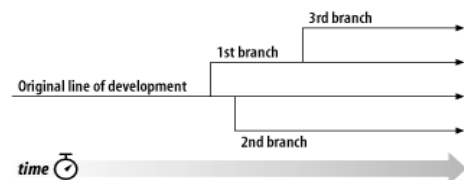
Démarrage rapide avec Subversion

Pour chaque projet de la forge, un dépôt Subversion est créé automatiquement au bout d'une heure maximum. L'URL du dépôt Subversion d'un projet de la forge est de la forme : <http://forge.clermont-universite.fr/repositories/show/<identifiant>>, où <identifiant> est l'identifiant du projet. On peut retrouver l'URL du dépôt Subversion d'un projet sur la page « dépôt » du projet.

Organisation d'un dépôt

Par convention, un dépôt Subversion contient 3 sous répertoires :

- `trunk` : qui contient la version de développement du projet
- `branches` : qui contient les différentes branches de développement
- `tags` : qui contient les versions officielles



Initier un projet (import)

- ▶ Créer localement un dossier (`sandbox` dans l'exemple) contenant les 3 répertoires, `trunk`, `branches` et `tags`.
- ▶ Mettre dans le répertoire `trunk`, le code source initial du projet
- ▶ Effectuer l'import avec la commande `svn import` :

```
$ svn import ./sandbox/ https://forge.clermont-universite.fr/svn/<identifiant> -m "Initial import"
```

Récupérer un projet (checkout)

```
$ svn checkout https://forge.clermont-universite.fr/svn/<identifiant>/trunk sandbox-dev
```

Un répertoire local de travail du projet sera créé (nommé ici `sandbox-dev`).

Effectuer une révision du projet (commit)

Une révision est un ensemble de modifications du projet (modifications de fichiers existants, ajouts et suppressions de fichiers et de répertoires).

Il faut se placer dans une copie de travail locale d'un répertoire du dépôt Subversion (créé auparavant avec une commande `svn checkout`).

Les modifications sur les fichiers déjà gérés par Subversion seront automatiquement prises en compte lors du commit. Les commandes Subversion pour effectuer des modifications sur la copie de travail sont :

- ▶ `svn add` : ajout de nouveaux fichiers ou répertoires
- ▶ `svn remove` : suppression de fichiers ou de répertoires
- ▶ `svn diff` : visualisation des différences entre le répertoire local et le dépôt
- ▶ `svn status` : liste des modifications de la révision

Pour propager une nouvelle révision, il faut ensuite utiliser la commande `svn commit` depuis le répertoire de travail :

```
$ svn commit -m "Description de la révision"
```

L'option `-m` permet de spécifier la description de la révision directement dans la ligne de commande. Si cette option est absente, `svn` va ouvrir un éditeur (vi en général) pour saisir la description.

Mettre à jour son répertoire local de travail (update)

Pour mettre à jour son répertoire de travail avec le dépôt lorsqu'il y a eu entre-temps de nouvelles révisions, il faut utiliser la commande :

```
$ svn update
```

Ressources

- Forge Clermont Université : <http://forge.clermont-universite.fr>
- FAQ : <http://forge.clermont-universite.fr/wiki/redmine-clruniv>
- Tutoriel Subversion : <http://svnbook.red-bean.com/nightly/fr/svn.intro.quickstart.html>
- Documentation complète de Subversion : <http://svnbook.red-bean.com/nightly/fr/svn-book.pdf>
- Documentations Redmine : <http://www.redmine.org/guide>

Contributeurs documentation:

Nom	Date	Type modifications	Version doc
AM	04/01/10	Version initiale	0.1
AM	12/03/10	Quickstart Subversion	0.2
AM	12/04/10	Corrections DRCH + Gestion des demandes	0.3