

Indefero, un bug tracker très complet

Par slash-dev



OPENCCLASSROOMS

www.openclassrooms.com

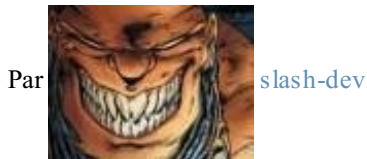
*Licence Creative Commons 6 2.0
Dernière mise à jour le 7/09/2010*

Sommaire

Sommaire	2
Et Si On Parlait ? (ESOP)	1
Indefero, un bug tracker très complet	3
Présentation	3
indefero, c'est quoi exactement ?	3
Pré-requis	3
Connaissances	4
Matériel/logiciel	4
Installation	4
Organisation	4
Indefero	5
Pluf	5
Pear	6
Configuration	6
idf.php	6
path.php	9
Base de données	9
Apache	10
Utiliser Git	12
Fonctionnement de git	12
Gestion des dépôts	12
Un problème de permissions	13
Gestion d'un dépôt git	15
Premiers pas avec Indefero	16
Administration de la forge	16
Utilisation d'un projet	16
Administration d'un projet	17
Partager	18



Indefero, un bug tracker très complet



Par

slash-dev

Mise à jour : 07/09/2010

Difficulté : Facile



Vous savez utiliser un système de gestion de version de sources et vous voulez créer un dépôt sur votre serveur avec un système complet de wiki/documentation, affichage des révisions, un bug tracker, etc ? Alors ce tutoriel est fait pour vous.

Ce tutoriel vous apprendra mettre en place Indefero qui est un projet s'inspirant du [système de google-codes](#) utilisé par exemple par le [projet chromium](#).

Ce tutoriel est basé sur la documentation en ligne (en anglais) que j'ai essayé de rendre plus facile et mieux guidée.

Sommaire du tutoriel :



- [Présentation](#)
- [Pré-requis](#)
- [Installation](#)
- [Configuration](#)
- [Utiliser Git](#)
- [Premiers pas avec Indefero](#)

Présentation

indefero, c'est quoi exactement ?

Pour ceux qui connaissent [Trac](#) (utilisé notamment par Assembla avec SVN pour ceux qui connaissent), Indefero est un équivalent également open-source qui offre une interface web d'administration des dépôts Git, SVN et Mercurial ainsi que leurs utilisateurs et ceux du site. Il peut gérer plusieurs dépôts et contient entre autres un bug tracker, une documentation et une gestion des patches proposés par les utilisateurs. C'est l'outil idéal pour un travail collaboratif.

Vous pouvez avoir un aperçu de ce à quoi Indefero ressemble sur [leur propre site](#). Les sources de Indefero sont gérées par Git et Indefero lui-même.

Indefero propose de nombreuses fonctionnalités dont voici les principales selon moi :

- Compatibilité avec Git, SVN et Mercurial
- Affichage des diff de chaque révision ou chaque fichier
- Administration des projets et utilisateurs
- Possibilité de rendre des projets privés
- Système de tickets (bug tracker)
- Navigation dans les sources et versions à partir d'un navigateur internet
- Création d'une documentation genre wiki
- Zone de téléchargement divers
- Téléchargement des sources à différentes versions du projet sous forme de Zip
- Aperçu des changements récents
- Choix de la langue pour les utilisateurs (français ou anglais)

Pour une liste plus complète, voir [la page d'accueil en anglais](#).

Pré-requis

Connaissances

Dans ce tutoriel, vous n'allez pas apprendre à utiliser un système de gestion de version. Si vous ne savez pas utiliser Git ou Svn je vous recommande donc d'aller lire les tutoriels suivants :

- Git : [Tutoriel de M@teo21](#)
- SVN : [Big tuto de Dalshim \(en cours de rédaction\)](#) ou bien [le mini-tuto d'apognu](#)
- Mercurial : [Mini-tuto de Meianki](#)



Dans ce tutoriel je n'aborderai l'utilisation de de Indefero qu'avec git. SVN viendra probablement plus tard. Mercurial ne sera probablement pas traité car je ne connais pas du tout.

Il est également conseillé d'avoir quelques connaissances en administration Linux et Apache.

Matériel/logiciel

L'objectif d'un dépôt de gestionnaire de version est d'être accessible de n'importe où et n'importe quand. Il est donc fortement conseillé d'avoir un serveur avec un nom de domaine (sachant qu'il est possible d'avoir plusieurs noms de domaines gratuits avec Azote). Il est possible de le faire en local mais l'utilité est assez réduite.

- Php 5.2.4 ou plus récent
- MySQL ou PostgreSQL
- Serveur web apache ou Nginx
- Git et/ou Subversion et/ou Mercurial
- Un accès à la ligne de commande sur votre serveur (ssh)
- Autorisations de mise en place d'une tâche cron

Le reste sera installé lors du tutoriel.



Dans ce tutoriel, j'explique pour une base de données MySQL et un serveur web apache et un serveur. A vous d'adapter à votre SGBD, votre serveur web ainsi que votre distribution de Linux. Ce tutoriel devrait également être assez simple à adapter pour Mac OS.



Il est également possible d'utiliser une base de données SQLite mais c'est déconseillé par Indefero pour un environnement de production.

Installation

Tout d'abord il va falloir installer Indefero puis installer deux frameworks (Pear et Pluf) qu'il utilise pour fonctionner.

Organisation

Sur mon serveur, tous mes sites web sont situés dans le répertoire /www. Vous pouvez préférer /home/www ou autre c'est comme ça vous convient. A partir de maintenant mon /www sera pour vous le répertoire dans lequel vous mettez vos sites web.



Le répertoire /www ne doit pas être accessible à partir d'internet. C'est juste le répertoire contenant les répertoires racines des sites webs (virtualhost) hébergés sur le serveur.

Selon la documentation, ils conseillent de mettre les sources d'Indefero et de Pluf dans ce répertoire. Cependant, j'ai préféré déplacer les sources de Pluf car ce n'est pas un site web, il ne mérite donc pas un répertoire dans /www directement pour moi. Je l'ai donc installé dans un sous-répertoire d'Indefero.

Pour quant à lui sera installé automatiquement et autre part.

Indefero

la première étape consiste à récupérer les sources. Elles se trouvent justement dans [la section "Téléchargements" d'Indefero](#). Il suffit de récupérer l'adresse du fichier zip à télécharger et utiliser wget pour le récupérer sur le serveur soit :

Code : Console

```
wget http://projects.ceondo.com/p/indefero/downloads/32/get/
```



Vous remarquerez le 32 dans l'URL. C'est l'identifiant de release. Si vous voulez télécharger une ancienne version, c'est possible en changeant ce nombre donc si une release 33 sort, vous téléchargerez la release 32 qui ne sera donc pas à jour. Vérifiez bien la version actuelle sur le site avant de copier-coller cette commande.

Vous avez maintenant reçu les sources compressées d'Indefero. il suffit de les dézipper et de supprimer l'archive comme suit :

Code : Console

```
unzip indefero-1.0.zip  
rm indefero-1.0.zip  
mv indefero-1.0 indefero
```

Vous pouvez un peu regarder l'architecture du répertoire décompressé. Il contient notamment un répertoire script contenant divers scripts pour son fonctionnement que nous verront plus tard, un répertoire src contenant l'ensemble des sources d'Indefero ainsi que les fichiers de configuration, et enfin le répertoire www qui sera le SEUL répertoire accessible à partir du web.

Dans le répertoire src on peut voir un seul répertoire nommé IDF qui contient toutes les sources d'Indefero. J'ai donc préféré installer le framework plus dans le répertoire src (soit /www/indefero/src plutôt que dans mon répertoire /www).

Pluf

Nous allons maintenant installer le framework Pluf. Vous pouvez l'installer ou bon vous semble. J'ai cependant préféré le répertoire src de Indefero comme expliqué au-dessus. A vous d'adapter les commandes. Comme Indefero, il faut récupérer les sources sur [Le site officiel](#) et les décompresser comme suit :

Code : Console

```
cd /www/indefero/src  
wget http://projects.ceondo.com/p/pluf/source/download/master/ --output-document=pluf.zip  
unzip pluf.zip  
rm pluf.zip  
mv pluf-master pluf
```



Comme vous pouvez le voir, j'ai ajouté une option `--output-document` à la commande `wget` sinon le fichier créé s'appelle `index.html`.

Pluf est maintenant installé. Rappelez-vous à quel endroit vous l'avez installé, vous en aurez besoin par la suite.

Pear

Installation

Pear est dans les dépôts pour les distributions de Linux "Debian-like", il suffit donc de faire :

Code : Console

```
aptitude install php-pear
```

Si vous n'avez pas Pear dans les dépôts, vous pouvez suivre les [instructions du site officiel](#), simples à suivre bien qu'elles soient en anglais.

Configuration

Pour fonctionner, Indefero a besoin de deux classes : Mail et Mail_mime. Pour les installer il suffit de faire :

Code : Console

```
pear upgrade-all  
pear install --alldeps Mail  
pear install --alldeps Mail_mime
```

Tout est maintenant installé, vous avez toutes les cartes en mains pour pouvoir faire fonctionner Indefero. Il ne reste plus qu'un travail de configuration à faire.

Configuration

Pour sa configuration, Indefero repose sur deux fichiers de configuration. Ils se trouvent dans le répertoire `indefero-1.0/src/IDF/conf/`. Placez-vous dans ce répertoire. Vous pouvez remarquer qu'il contient 3 fichiers. Seulement deux nous intéressent, ce sont les deux finissant par `-dist`. Les fichiers `-dist` contiennent une configuration par défaut qu'il va falloir modifier. Pour les mettre en service, il faut créer une copie de chaque fichier en supprimant l'extension `-dist`.

idf.php

Pour mettre en place ce fichier, tapez la commande :

Code : Console

```
cp idf.php-dist idf.php
```

Maintenant il va falloir éditer ce fichier de façon à ce qu'il corresponde à votre système. Si vous regardez vous-même les variables, vous comprendrez vite lesquelles il faut modifier mais je vais quand même vous aider et vous dire lesquelles j'ai

modifiées dans l'ordre d'apparition. Je ne mets ici que les variables qui ne concernent pas un système de gestion de version particulier. La configuration de chaque système sera décrite plus tard dans la section appropriée.

Code : PHP - idf.php

```
<?php
$cfg['debug'] = false; // A mettre à faux pour la mise en
production
?>
```

Vous pouvez ensuite commenter toutes les lignes concernant git, svn ou mercurial. Nous les mettrons en place dans leurs parties respectives.

Mails

La partie suivante est la configuration de l'e-mail et serveur smtp, il est possible de désactiver cette option en définissant la variable \$cfg['send_emails'] à false. Cependant les mails sont utiles notamment pour vérifier la validité d'un utilisateur lors de son inscription, un mail lui est envoyé automatiquement avec un code de vérification. Si vous n'avez pas de serveur SMTP, pas de panique, google est là ! 😊

Pour configurer Indefero avec votre adresse gmail, c'est très simple, il suffit d'avoir une adresse valide et de définir la configuration comme cela :

Code : PHP - idf.php

```
<?php
// Cette variable contient la liste des administrateurs avec leur
adresse mail
// Un mail leur sera envoyé si une erreur survient en non debug
mode
// Mettez donc votre adresse mail
$cfg['admins'] = array(
    array('Admin', 'VOTRE_ADRESSE@gmail.com'),
);
$cfg['send_emails'] = true;
$cfg['mail_backend'] = 'smtp';
$cfg['mail_auth'] = true;
$cfg['mail_host'] = 'ssl://smtp.gmail.com';
$cfg['mail_port'] = 465;
$cfg['mail_username'] = 'VOTRE_ADRESSE@gmail.com';
$cfg['mail_password'] = 'MOT_DE_PASSE';
?>
```

Allé, je fais un peu de zèle, si vous avez déjà une adresse gmail et que vous ne voulez pas avoir à vous connecter à l'une puis l'autre, il est non seulement possible de recevoir une copie des mails reçu sur votre adresse personnelle mais en plus de pouvoir choisir l'expéditeur lors de l'envoi d'un mail. Donc quand vous envoyez un mail, en un clic vous pouvez l'envoyer à partir de votre adresse d'administration. Pour cela, allez dans votre compte personnel puis dans paramètres->"Comptes et importations"->"Vérifier les messages avec le protocole POP3". Ici cliquez sur "Ajouter un compte de messagerie POP3" et suivez les instructions.

Revenons maintenant à notre configuration :

Code : PHP - idf.php

```
<?php
// Ces deux variables dépendent de votre configuration apache
// Nous reviendrons dessus plus tard
// Par défaut mettes-les comme tel :
```

```

$cfg['idf_base'] = '/index.php';
$cfg['url_base'] = 'http://NOM_DE_DOMAINE';

// Pour accéder au répertoire media et upload. On ne l'a pas déplacé
// Donc il suffit de mettre à jour le nom de domaine
$cfg['url_media'] = 'http://NOM_DE_DOMAINE/media';
$cfg['url_upload'] = 'http://NOM_DE_DOMAINE/media/upload';
$cfg['upload_path'] = '/www/indefero/www/media/upload'; // chemin absolu
$cfg['upload_issue_path'] = '/www/indefero/attachments';

// Je ne sais pas trop à quoi ça sert mais ils conseillent qu'elle fasse
// plus de 40 caractères, faites-vous plaisir
$cfg['secret_key'] =
'jhkfg2s4h1rss56g21j564dd21b56rf2s2k52mioi5d2s6zs';

// Mettez votre adresse que vous voulez. La deuxième est une
// adresse no-reply
$cfg['from_email'] = 'ADRESSE@gmail.com';
$cfg['bounce_email'] = 'ADRESSE@gmail.com';
?>

```

Base de données

Bien entendu, Indefero utilise une base de données. Vous pouvez choisir la base de données que vous voulez utiliser entre MySQL et PostgreSQL. Les variables permettent la connexion à la base de données, les noms sont assez explicites. Je compte sur vous pour créer la base et initialiser les utilisateurs de la manière qu'il vous convient. Si vous ne savez pas comment faire, la documentation d'Ubuntu devrait vous aider. <http://doc.ubuntu-fr.org/mysql>

Code : PHP

```

<?php
$cfg['db_login'] = 'LOGIN';
$cfg['db_password'] = 'MOT_DE_PASSE';
$cfg['db_server'] = ''; // Probablement localhost

// Utile seulement pour MySQL, à commenter pour PostgreSQL
$cfg['db_version'] = '5.0'; // Connectez-vous à mysql, elle sera affichée

// Cette chaîne sera préfixée à chaque nom de table. ça permet de faire tourner
// plusieurs Indefero sur une même table.
$cfg['db_table_prefix'] = 'indefero_';
$cfg['db_engine'] = 'MySQL'; // Ou PostgreSQL selon votre besoin
$cfg['db_database'] = 'NOM_DATABASE';
?>

```

Autre

Voici les trois dernières variables à modifier :

Code : PHP

```

<?php
$cfg['pear_path'] = '/usr/share/php5'; // Moi c'est php5 au lieu de php, vérifiez

```



```
// Celle-ci n'est pas obligatoire mais c'est plus rigoureux je trouve
$cfg['cache_file_folder'] = $cfg['tmp_folder'].'/cache_indefero';

// J'ai mis Paris mais vous pouvez adapter
// (cf liste http://www.php.net/manual/fr/timezones.php)
$cfg['time_zone'] = 'Europe/Paris';
?>
```

Pour ce qui est du reste, vous pouvez regarder mais il n'est pas nécessaire d'y toucher pour que ça marche.

path.php

Ce fichier contient les path de l'application. Il sera beaucoup plus rapide à éditer. Comme idf, il faut commencer par le copier :

Code : Console

```
cp path.php-dist path.php
```

Étant donné qu'on n'a pas installé Pluf au même endroit que ce que demande la documentation, il faut changer son path, soit pour moi :

Code : PHP

```
<?php
define('PLUF_PATH', dirname(__FILE__).'/../..../pluf/src');
?>
```

Base de données

Il va maintenant falloir initialiser la base de données. Ne vous inquiétez pas, Indefero s'occupe de tout. Nous allons utiliser un script qui a été mis en place. Ce script propose une option permettant de tout vérifier. Donc de faire comme s'il initialisait tout mais en fait il ne fait rien. Si le script fonctionne avec cette option, vous pouvez ensuite l'enlever et réellement exécuter le script qui cette fois initialisera tout pour de bon. Voici la fameuse commande :

Code : Console

```
cd /www/indefero/src/
php pluf/src/migrate.php --conf=IDF/conf/idf.php -a -i -d -u
```

Tant que ce code renvoie des erreurs, corrigez-les et continuez à exécuter le script jusqu'à ce qu'il n'y en ait plus. Les premières erreurs peuvent venir d'un problème dans le fichier path (s'il ne trouve pas le fichier Pluf.php ou autre), ensuite sinon c'est dans le fichier idf.php. Et enfin, si c'est une erreur dans la base de données c'est que vous avez du oublier de créer la base ou l'utilisateur ou tout simplement que les informations que vous avez rentrées dans le fichier idf.php à propos de la base de données ne sont pas correctes.

Une fois qu'il n'y a plus d'erreurs, la sortie devrait être la suivante :

Code : Console

```
PHP include path: ./usr/share/php:/usr/share/pear:/www/indefero_tuto/src/pluf/src
Install all the apps
Pluf_Migrations_Install_setup
IDF_Migrations_Install_setup
```

Vous pouvez maintenant exécuter le script sans l'option -u qui initialisera réellement Indefero cette fois. La sortie devrait être exactement la même qu'avec l'option -u.

Code : Console

```
php pluf/src/migrate.php --conf=IDF/conf/idf.php -a -i -d
```

Ce code a donc créé toutes les tables dans la base de données dont indefero a besoin pour son fonctionnement. On va maintenant créer un administrateur pour le site. Soit votre compte administrateur. Pour cela, nous allons créer une script dans le répertoire src nommé init_user.php dans lequel vous allez copier ce code et modifier les parties personnelles.

Code : PHP - init_user.php

```
<?php
require '/www/indefero/src/IDF/conf/path.php';
require 'Pluf.php';
Pluf::start('/www/indefero/src/IDF/conf/idf.php');
Pluf_Dispatcher::loadControllers(Pluf::f('idf_views'));

$user = new Pluf_User();
$user->first_name = 'VOTRE_PRENOM'; // Prénom
$user->last_name = 'VOTRE_NOM'; // Nom (obligatoire)
$user->login = 'LOGIN'; // Login en minuscule
$user->email = 'VOTRE_EMAIL@gmail.com';

// Le mot de passe sera crypté dans la base de données
$user->password = 'yourpassword';
$user->administrator = true;
$user->active = true;
$user->create();
print "Utilisateur créé\n";
?>
```

Exécutez ensuite le code avec php. Une fois exécuté, le fichier peut être supprimé.

Code : Console

```
php init_user.php
rm init_user.php
```

Indefero est maintenant configuré. Il ne reste plus qu'à configurer apache pour pouvoir y accéder.

Apache

Pour cela on va créer un nouveau vhost dans /etc/apache2/sites-available/ nommé indefero et y mettre le contenu suivant à adapter bien évidemment :

Code : Apache - /etc/apache2/sites-available/indefero

```
<VirtualHost *:80>
# ServerName définit le nom utilisé pour le vhost. Mettez le nom
de l'hôte du domaine
ServerName VOTRE_FQDN
# ServerAlias définit les autres sous domaines pour lesquels le
serveur répondra.
ServerAlias www.VOTRE_FQDN

# DocumentRoot définit le dossier racine dans lequel seront stockés
les fichiers du site.
DocumentRoot /www/indefero/www
# Directory définit les options par défaut du répertoire
<Directory /www/indefero/www >
Options FollowSymLinks MultiViews
AllowOverride All
# Droits par défaut
Order allow,deny
allow from all
</Directory>

ErrorLog /var/log/apache2/error.indefero.log
CustomLog /var/log/apache2/access.indefero.log combined
</VirtualHost>
```



Modifiez la configuration si elle ne vous convient pas, j'utilise celle-ci et ça marche. Libre à vous de la modifier.

Il faut ensuite activer le site et recharger apache avec la nouvelle configuration.

Code : Console

```
a2ensite indefero
/etc/init.d/apache2 reload
```

Il faut maintenant donner un accès en lecture au serveur apache mais aussi un accès en écriture au répertoire upload. Nous savons bien qu'un "chmod 777" n'est pas une bonne solution du tout. Lorsque Apache est démarré, par défaut il tourne sous l'utilisateur www-data et le groupe www-data. Il suffit donc de définir comme groupe à tous les fichiers et répertoires le groupe www-data et de mettre à jour les permissions.

Code : Console

```
chgrp www-data /www/indefero/ -R
chmod 750 /www/indefero/ -R
chmod 770 /www/indefero/www/media/upload
```



Il aurait également été possible d'ajouter l'utilisateur www-data au groupe du titulaire des fichiers mais j'ai choisi cette solution car l'utilisateur est root et que j'estime qu'il n'est pas trop bon d'ajouter un utilisateur au groupe root.

Et enfin, pour finir, si vous avez le mod_rewrite sur votre serveur, pour avoir de belles url, ajoutez le fichier .htaccess dans le répertoire /www/indefero/www avec le contenu suivant :

Code : Apache - /www/indefero/www/.htaccess

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*) /index.php/$1
```

Et modifiez dans le fichier de configuration idf.php les deux variables suivantes :

Code : PHP

```
<?php
$config['idf_base'] = '';
$config['url_base'] = 'http://VOTRE_FQDN';
?>
```

Si en revanche, vous n'avez pas le `mod_rewrite`, il va vous falloir utiliser le fichier `index.php` dans l'url systématiquement avec la configuration du fichier `idf.php` suivante :

Code : PHP

```
<?php
$config['idf_base'] = 'index.php';
$config['url_base'] = 'http://VOTRE_FQDN';
?>
```

Et voilà, votre interface est enfin accessible à partir de l'url `http://VOTRE_FQDN`. Vous pouvez vous connecter avec le compte administrateur que vous avez créé auparavant. Vous pouvez également tester la connexion, si vous avez bien configuré votre serveur smtp (avec gmail par exemple, vous devriez recevoir un e-mail de confirmation de votre adresse mail lors de l'inscription). Vous pouvez également gérer les utilisateurs sur le site. Si l'interface est en anglais, vous pouvez le modifier dans votre profil pour mettre la langue en français.

Cependant, vous ne pouvez pas encore créer de projet car nous n'avons ni configuré Svn ni git. Ce sera le sujet de la partie suivante, vous y êtes presque !

Utiliser Git

Indefero ne permet pas seulement d'afficher sur une page web les informations données par Git. Il est également possible de gérer les dépôts (création/suppression) ainsi que gérer les différents utilisateurs et leurs permissions sur les dépôts. Si vous ne savez pas trop comment marche Git, je vais commencer par une explication rapide du fonctionnement de Git.

Fonctionnement de git

Tout d'abord, il faut savoir que git utilise le protocole SSH pour tous les transferts de données entre le client et le serveur. C'est un protocole sécurisé de communication qui crypte tous les transferts à l'aide d'un système de cryptographie asymétrique notamment avec RSA. Pour plus d'informations à ce sujet, je vous conseille [le tutoriel de TuxWeb](#).

Pour gérer les utilisateurs, Git les identifie par leur clé publique SSH. Indefero propose donc lorsque vous avez créé un compte d'ajouter une ou plusieurs clés publiques SSH à votre profil. A chaque projet il suffit de donner la liste des membres autorisés à accéder au dossier et le tour est joué.

Gestion des dépôts

Pour gérer les dépôts, nous allons créer un utilisateur système nommé `git`. Son répertoire home sera `"/home/git"`. Nous l'utiliserons ensuite pour créer le répertoire qui contiendra les dépôts.

Code : Console

```
adduser \  
    --system \  
    --shell /bin/bash \  
    --gecos 'git version control' \  
    --group \  
    --disabled-password \  
    --home /home/git \  
    git  
# on passe sous l'utilisateur git  
su git  
mkdir /home/git/repositories  
exit
```



Le exit à la fin permet de revenir à l'utilisateur root.

Un problème de permissions

Accès aux informations des dépôts

Pour qu'Indefero puisse afficher les différentes informations de dépôts, il lui faut la possibilité de lire les informations. Les sources d'Indefero sont exécutées par apache donc avec l'utilisateur www-data par défaut comme vu auparavant. Nous allons donc ajouter l'utilisateur www-data au groupe git puis redémarrer apache pour qu'il prenne en compte les modifications.

Code : Console

```
usermod -a -G git www-data  
/etc/init.d/apache2 restart
```

Gestion des clés ssh

Pour qu'un utilisateur se connecte à un dépôt en SSH il faut que la clé publique de l'utilisateur figure dans le fichier .ssh/authorized_keys. Nous allons donc le créer mais comme vous devez le savoir, SSH est très rigoureux à propos des permissions son répertoire .ssh et ses fichiers qu'il contient. C'est pourquoi il va également falloir modifier les permissions.

Code : Console

```
su git  
mkdir /home/git/.ssh  
touch /home/git/.ssh/authorized_keys  
chmod 0700 /home/git/.ssh  
chmod 0600 /home/git/.ssh/authorized_keys  
exit
```

Lors de l'ajout d'une clé publique ssh dans Indefero, il va falloir qu'il ajoute cette clé dans le fichier authorized_keys. Seulement une fois de plus, l'utilisateur est www-data et seul l'utilisateur git peut modifier ce fichier. Cette fois il n'est pas possible de modifier les groupes et permissions car SSH l'interdit. Pour régler ce problème, l'équipe d'Indefero a mis un mécanisme en place.

L'idée est que lorsqu'il y a une mise à jour des clés, l'utilisateur www-data demande à l'utilisateur git de mettre à jour le fichier. Pour prévenir l'utilisateur git qu'une modification a été faite, il va créer un fichier vide dans le système de fichiers à un endroit convenu dans les fichiers configuration. Ensuite il va falloir faire exécuter régulièrement un script par l'utilisateur git qui va

regarder si le fichier existe et si oui, il va mettre à jour les clés, sinon il ne fait rien. Pour faire régulièrement exécuter un script par un utilisateur, il faut mettre en place une tâche cron. Voir le tutoriel de [tutoriel de M@teo21](#) pour plus d'informations à ce sujet. Pour que le système puisse fonctionner, il est nécessaire que les deux utilisateurs aient un accès en écriture sur le répertoire où sera placé le fichier. La documentation propose le répertoire /tmp mais il n'est pas possible à l'utilisateur git de supprimer un fichier créé par www-data donc le fichier `authorized_keys` était mis à jour à chaque fois même lorsqu'il n'y avait pas de changements (cf [rapport de bug 505 d'Indefero](#)). On va donc créer un répertoire nommé "sync" dans le répertoire home de git dans lequel on autorisera le groupe www-data à supprimer des fichiers.

Code : Console

```
su git
cd ~
mkdir sync
chgrp www-data sync
chmod 770 sync
exit
```

Les deux utilisateurs peuvent maintenant créer et supprimer des fichiers à l'intérieur de ce répertoire. Il faut maintenant ajouter la configuration suivante dans le fichier `idf.php`

Code : PHP - /www/indefero/src/IDF/conf/idf.php

```
<?php
// Où se trouvent les dépôts dans le système de fichier et l'url
// le %s représente le nom du dépôt vu qu'il peut y en avoir
plusieurs
$config['git_repositories'] = '/home/git/repositories/%s.git';
$config['git_write_remote_url'] = 'git@NOM_DE_DOMAINE:%s.git';
$config['git_remote_url'] = 'git://NOM_DE_DOMAINE/%s.git';

// configuration du plugin

// path du fichier mettant à jour le fichier des clés
$config['idf_plugin_syncgit_path_gitserve'] =
'/www/indefero/scripts/gitserve.py';
// Fichier des clés à mettre à jour
$config['idf_plugin_syncgit_path_authorized_keys'] =
'/home/git/.ssh/authorized_keys';

// Propose de supprimer réellement un dépôt automatiquement lors de
la suppression d'un projet
$config['idf_plugin_syncgit_remove_orphans'] = false;
// répertoire home de l'utilisateur git
$config['idf_plugin_syncgit_git_home_dir'] = '/home/git';
// chemin absolu du fameux fichier de synchronisation
$config['idf_plugin_syncgit_sync_file'] =
$config['idf_plugin_syncgit_git_home_dir'].'/sync/SYNC-GIT';
// path des dépôts
$config['idf_plugin_syncgit_base_repositories'] =
'/home/git/repositories';
?>
```

Il suffit ensuite de configurer votre tâche cron. Le script à exécuter est un script PHP qu'il doit falloir pouvoir exécuter à la ligne de commande. Il faut donc installer le paquet "php5-cli"

Code : Console

```
aptitude install php5-cli
```

Il faut ensuite ajouter la tâche cron. Pour cela, il faut éditer les contab en temps qu'utilisateur git soit :

Code : Console

```
su git
crontab -e
```

Et ajouter la ligne suivante :

Code : Autre

```
5 * * * * php /www/indefero/scripts/gitcron.php
```



Ce script exécutera la mise à jour toutes les 5 minutes. Au plus vous le réduirez, au plus vite un utilisateur sera ajouté lorsqu'il sera inscrit. À vous d'adapter en fonction de vos besoins.

Votre travail est enfin fini et votre installation est opérationnelle ! Maintenant vous pouvez commencer à l'utiliser. Je vais maintenant rapidement vous montrer comment créer un projet et ajouter des utilisateurs à celui-ci. Si vous voudrez aller plus loin, la partie "Premiers pas avec Indefero" vous aidera à découvrir une partie des fonctionnalités.

Gestion d'un dépôt git

Création du projet

Pour créer un nouveau projet git, c'est vraiment très rapide. Connectez-vous avec votre compte administrateur que nous avons créé auparavant. Allez ensuite dans la section "Administration de la forge". Vous verrez ici la liste des projets existants s'il y en a. Allez ensuite dans la section "Créer un projet". Il vous sera ensuite demandé un nom et un nom court qui permettra de créer l'URL du projet. Choisissez ensuite git et ajoutez les noms des propriétaires des projets et des membres. Si vous rentrez des noms invalides, le formulaire ne pourra pas être validé. Et enfin, si vous souhaitez que votre projet soit privé et autoriser l'accès aux membres seulement, vous pouvez cocher la case "privé".

Ajout d'utilisateurs

Vous faites maintenant partie du projet et vous pouvez y contribuer mais avant cela, il faut spécifier une clé publique ssh pour chaque membre. Allez donc dans espace membre en cliquant sur votre nom en haut à gauche puis allez dans la section "Mettre à jour votre compte". Dans cette section, vous pouvez changer vos informations personnelles ainsi que gérer vos clés ssh. Il vous suffit donc d'ajouter une clé dans le champ "Ajouter une clef publique SSH:" et de mettre à jour votre compte. L'utilisateur sera donc reconnu grâce à cette clé.

Initialisation du projet

Votre dépôt git n'est pas encore créé. Pour le créer, il suffit de suivre les instructions dans la section source de votre projet qui dit de faire :

Code : Console

```
git init
git add .
git commit -m "initial import"
git remote add origin git@NOM_DOMAINE:NOM_PROJET.git
git push origin master
```

Voilà, votre projet est maintenant créé et initialisé. Vous pouvez gérer les utilisateurs qui y auront accès, etc. A vous de découvrir toutes les fonctionnalités. Pour une introduction, vous pouvez lire la partie "Premiers pas avec Indefero".

Premiers pas avec Indefero

Administration de la forge

Gestion des projets

Avec Indefero, il est facile de créer/supprimer des projets. Pas besoin d'utiliser la ligne de commande si tout est bien configuré. Pour le créer c'est très simple, il suffit d'aller dans la section "Créer un projet" et suivre les instructions. Les noms des membres doivent exister dans la base de données pour pouvoir valider le formulaire.



Lors de la suppression d'un projet sur l'interface d'Indefero, il est possible de choisir si on veut que les dépôts (gérés pas Git, SVN ou Mercurial et pas Indefero) soient également supprimés. Pour permettre cela, il faut définir la `<?php $cfg['idf_plugin_sync***_remove_orphans'] ?>` dans le fichier `idf.php` à true avec les *** dépendant du gestionnaire de sources utilisé (Git, SVN ou Mercurial).

Gestion des utilisateurs

Dans l'onglet utilisateurs, on peut voir la liste des utilisateurs et leurs permissions. Ces permissions concernent la forge en général et pas les projets spécifiques. Il est possible de définir des administrateurs ou des membres du staff. Il est également possible d'activer/désactiver un compte.

Sur la gauche il est possible de voir un lien "Voir les utilisateurs non confirmés." qui porte un nom assez clair. Ce sont les utilisateurs qui n'ont pas validé leur compte avec le adresse email. Il est ensuite possible de confirmer manuellement l'inscription des utilisateurs.

Utilisation d'un projet

Indefero offre plusieurs outils utiles chacun dans un onglet séparé.

Téléchargements

Comme son nom l'indique il permet à l'équipe de mettre en ligne des fichiers à télécharger comme par exemple les différents exécutables pour les différents OS et différentes versions de votre programme. Regardez par exemple [la section téléchargement pour les sources d'Indefero](#). Vous remarquerez qu'on peut voir la version la plus récente. Sur le menu de gauche, vous pouvez également choisir de trier par OS, type, etc. Ce projet ne contient qu'une seule archive de type zip pour Linux et Mac donc le tri n'est pas vraiment utile. Cependant vous pouvez également trouver un lien "Voir les fichiers obsolètes." qui permet de lister tous les téléchargements qui ont été marqués comme obsolètes (Deprecated).

Si vous avez les permissions, vous pouvez ajouter un nouveau téléchargement. Il vous faudra donner un résumé et description ainsi que le fichier à mettre en ligne. Il est ensuite possible de mettre des étiquettes aux téléchargements. Vous avez la possibilité d'en mettre jusqu'à 6. En double cliquant sur un champ, une liste d'étiquettes apparaît. Il vous suffit d'en sélectionner une quelconque ou bien d'en écrire une manuellement. Elles permettent de spécifier par exemple le type (exécutable, paquet, archive...), le système d'exploitation (tous, Windows, Linux...) ainsi que le statut "Deprecated" si le fichier est déprécié et qu'une version plus récente existe (comme expliqué au dessus). Il est possible de modifier et personnaliser la liste des étiquettes proposées par défaut mais ce sera le sujet de la partie suivante "Administration d'un projet".

Documentation

Il est possible de mettre en ligne une documentation. Pour cela, Indefero propose un système genre Wiki. Comme les téléchargements, si vous avez les droits vous pourrez ajouter une page dans laquelle il vous sera demandé le titre, une

description et le contenu rédigé à la manière d'un Wiki. Un système d'étiquette semblable aux téléchargement est disponible.

Tickets

Cette section est en fait le bug tracker d'Indefero. Il est possible de créer un ticket comportant un résumé et une description d'un bug trouvé sur le projet ou bien une amélioration qu'on voudrait voir apparaître. Il est également possible de joindre un fichier qui aiderait à rendre le problème plus clair. Il est possible d'y mettre un statut de la même façon que les étiquettes. Au début le statut est "nouveau". Plus tard il est possible de le considérer comme fermé/résolu (si ce n'est pas un bug mais ou s'il a été résolu). Les autres utilisateurs peuvent ensuite ajouter des messages à chaque ticket pour proposer des solutions ou suggestions.

Sources

Cette section permet de se déplacer dans les sources, de voir les différences entre les versions de façon conviviale, de télécharger un fichier, de télécharger les sources complètes du projet à une version spécifique, voir la liste des changements, etc.

Revue de code

Cette section permet de déposer des patches en spécifiant un résumé, une description, le commit concerné par ce patch et surtout le fichier contenant les modifications.

Administration d'un projet

Les membres ayant été définis comme propriétaires d'un projet peuvent administrer le projet. Pour cela, il suffit d'aller dans la section "Administration d'un projet" la plus à droite.

Résumé du projet

Cette section permet de spécifier le nom du projet, une courte description ainsi qu'une description complète du projet qui sera affichée sur la page d'accueil du projet.

Téléchargements et documentation

Nous avons vu tout à l'heure qu'il était possible de mettre des étiquettes sur des fichiers à télécharger ou des pages de documentation à partir d'une liste. C'est ici qu'on peut créer ces listes. Ces listes sont sous la forme :

Code : Autre

Classe:Nom = Description

La classe est facultative. Comme on peut le voir il existe une classe OpSys qui englobe la liste des systèmes d'exploitations, pareil pour le type. Il est donc possible de créer de nouvelles classes.

Le second permet donner la liste des classes pour lesquelles on veut avoir seulement une étiquette par téléchargement pour cette classe. Par exemple par défaut il y a le champ type. C'est logique car un fichier ne peut pas être à la fois un exécutable et une archive zip. C'est pour ça que la classe Type apparaît dans ce champ. A l'inverse la classe OpSys n'apparaît pas car un fichier peut être utilisable pour Windows et Linux par exemple.

Gestionnaire de tickets

Même principe que les téléchargements et documentations sauf que les deux premiers champs concernent les statuts des tickets. Le premier concerne les noms des tickets qui seront considérés comme ouverts ou actifs. Par exemple si le ticket a un statut de "nouveau" il est considéré comme actif. A l'inverse un ticket ayant un statut de "résolu", "invalide" ou "duplicat" (quand quelqu'un signale un bug déjà signalé auparavant), le ticket sera considéré comme fermé. Il y a ensuite le même système d'étiquettes que la documentation et téléchargements.

Membres du projet

Cette section permet de gérer les permissions des membres. Un propriétaire est un membre ayant les droits de lecture et écriture sur les dépôts ainsi que l'accès à l'interface d'administration du projet. Un membre est comme un propriétaire sauf qu'il n'a pas accès à l'interface d'administration.

Accès aux onglets et notifications

Cette section permet de définir les permissions d'accès au site. On peut choisir pour chaque section qui peut y accéder (ouvert à tous, utilisateurs authentifiés, membres, propriétaires ou tout simplement fermé).

La section utilisateurs autorisés supplémentaires permet de donner seulement le droit de lecture aux dépôts à ces utilisateurs. Ils ne pourront donc pas faire de commit sur le serveur mais pourront récupérer les sources.

Ce tutoriel est terminé. J'espère qu'il vous aura aidé et que vous n'avez pas rencontré de problèmes. J'ajouterai bientôt une partie sur la création de dépôts SVN et peut-être une sur Mercurial si je m'y mets.

Si vous avez la moindre remarque sur ce tutoriel, imprécision, erreur ou quoi que ce soit, merci de m'en faire part.

Pour de l'aide, vous pouvez aller sur le chanel IRC officiel (#indefero sur les serveurs FreeNode). C'est en anglais mais certains membres parlent français si vraiment vous ne parlez pas anglais.

Partager

