




# Gitea, la forge logicielle web avec une tasse de thé (et un carré de chocolat)

- Objet : Utilisation de Gitea pour les membres DF
- Niveau requis :  
[débutant](#), [avisé](#)
- Commentaires : Pour une utilisation basique de Gitea hébergé sur DF. Connaître des rudiments de *Git* est nécessaire.
- Débutant, à savoir : [Une introduction à Git, un système de gestion de version décentralisé](#)
- Suivi :  
[à-compléter](#), [à-tester](#)
  - Création par  [bendia](#) 04/08/2019
  - Testé par <...> le <...> 
- Commentaires sur le forum : [Lien vers le forum concernant ce tuto](#) <sup>1)</sup>

Nota :


Contributeurs, les  sont là pour vous aider, supprimez-les une fois le problème corrigé ou le champ rempli !

## Introduction

Gitea est une [forge logicielle web](#), c'est à dire un outil en ligne permettant à plusieurs personnes (développeurs ou non) de collaborer à la création de logiciels. Elle permet entre autre de :

- Gérer les version du code source grâce à [git](#), soit via la ligne de commande, mais aussi graphiquement avec l'interface web.
- Gérer les droits sur les différents dépôts grâce à la création d'organisations pouvant contenir des équipes.
- Créer et gérer des tickets pour signaler et traiter des bugs, proposer des améliorations et répartir les tâches à accomplir.
- Créer et gérer un wiki, pour documenter le code ou l'utilisation du logiciel par exemple

Ce tuto n'a pas pour objectif d'expliquer l'utilisation de git, il existe pour cela de nombreux tuto, dont un livre sous licence CC<sup>2)</sup>. Il ne traitera également pas de toutes les possibilités offertes par Gitea.







 Si certains veulent s'y lancer, pourquoi pas des tutos spécifiques pour des fonctions précises (genre les *hook* ?)

## Le tableau de bord


Le *Tableau de bord* est la page principale qui s'affiche après la connexion.



A partir de ce tableau de bord, on va pouvoir :

-  accéder aux réglages et profils du compte
-  créer de nouveaux dépôts, de nouvelles organisations
-  Voir les notifications
-  visualiser les dépôts et organisations sur lesquels nous travaillons (notez qu'on peut également en ajouter avec le signe +)
-  Changer le contexte du tableau de bord, pour passer naviguer entre votre compte propre et les différentes *organisations* auxquelles vous appartenez
- On voit également au milieu un graphique montrant votre activité, et l'activité des projets que vous suivez
-  Dans la barre de navigation, on peut également se diriger vers les tickets de vos dépôts, les demandes d'ajouts, et l'explorateur permettant de visualiser les dépôts publics du serveur.

## Profils et réglages

 Pour modifier son profil (nom, bio, site web, mot de passe ...), gérer son mot de passe, gérer ses dépôts et organisations et ajouter une clé SSH, c'est par là

### Ajouter une clé SSH

Habituellement, pour pousser des modifications dans un dépôt depuis sa machine personnelle, on utilise *SSH*. Il va donc falloir ajouter une [clé SSH](#) grâce à l'onglet **Clés SSH/GPG**


On donne ensuite le nom à la clé, et on copie son contenu dans le cadre qui va bien avant de sauvegarder



Avec ça, vous voilà prêt à interagir avec *Gitea* en ligne de commande directement depuis votre machine 😊

## Les dépôts

### Créer un dépôt

Le dépôt, c'est la collection de fichiers qui constitue le code source du logiciel (merci VV222 😊). Ce dépôt a un propriétaire (vous ou une organisation). Pour créer un dépôt, nous pouvons cliquer sur *Nouveau*, puis *Nouveau dépôt*, ou sur le bouton  du cadre *Dépôts/Organisation*, et cette page apparaît alors.



Outre le nom (qui ne doit contenir que des caractères alphanumériques, des tirets -, des tirets bas \_, et des points. ), on remarquera les deux cases à cocher :

- *Rendre le dépôt privé* ne permettra qu'à vous (ou aux personnes que vous ajouterez comme *collaborateurs* de voir ce dépôt.
- *Initialiser le dépôt (ajoute les fichiers .gitignore, Licence et LISEZMOI)* va créer ces 3 fichiers en

même temps que votre dépôt. Vous pouvez choisir le type de licence qui s'appliquera à votre logiciel (GPLV3, MIT ...) et une copie de celle-ci sera ajoutée dans le fichier *Licence*. Le fichier *LIZEZMOI* se remplira du contenu du cadre *Description*. Ce fichier peut être formaté en [Markdown](#) et son contenu s'affichera converti en HTML sous la liste des fichiers du dépôt.

On peut également créer un dépôt nu. Une fois créer on y trouve alors toutes les commandes permettant de créer son dépôt en local, puis, de le pousser, ou alors de pousser un dépôt local déjà existant.

## Travailler avec un dépôt

On a désormais un dépôt avec ses premiers fichiers.



- copier directement l'adresse (ssh ou http) permettant de cloner le dépôt
- télécharger ces fichiers dans une archive.

## Sujets

Les sujets, sont des *tags* qu'on peut appliquer au dépôts, et qui permettrons de lister tous les dépôts ayant un même sujet dans l'explorateur. Par exemple, si on donne le langage dans lequel est codé l'application, on pourra lister tous les dépôts publics codés dans ce même langage.

## Paramètres

permet notamment d'activer ou désactiver le gestionnaire de ticket, le wiki, de modifier la visibilité du dépôt (publique/privé), d'ajouter des collaborateurs, de changer le propriétaire, de l'archiver ou le supprimer.

## Modifications de fichiers via l'interface web

- créer un nouveau fichier vide, ou de téléverser des fichiers
- Cliquez sur le nom d'un fichier l'affiche, il est alors modifiable directement en ligne avec le bouton .


Ces actions donnerons lieu à des nouveaux *commits*.

## Les organisations

Il s'agit d'un regroupement d'utilisateurs qui s'organise en équipe. Cela permet de gérer plus finement les droits sur chaque dépôt appartenant à *l'organisation* qu'un simple dépôt personnel collaboratif.

 ou le bouton  du cadre *Organisation* pour créer une nouvelle *Organisation*.

On se retrouve alors sur le tableau de bord de l'organisation. On peut alors créer des dépôts qui appartiendront à l'organisation de la même manière que pour un utilisateur.

 pour accéder à la page de l'organisation.

## Les équipes



L'équipe *Owners* est automatiquement créée avec le créateur de l'organisation comme membre. Les membres de cette équipes ont tous les droits sur tous les dépôts appartenant à l'organisation. Ce sont les administrateurs de l'organisation.

On peut ajouter de nouveaux membres à une équipe en cliquant sur son nom

 pour créer une nouvelle équipe.



On peut alors ajouter des membres à l'équipe, et des dépôts sur lesquelles cette équipe aura des droits.



Ne pas oublier cette dernière étape, sinon, l'équipe ne peut rien faire 😊

## Les droits



Lors de la création d'une équipe, il faudra gérer les droits.

- *Accès en lecture* est utile pour donner un accès en lecture seule aux dépôts privés
- *Accès en écriture* se règle finement pour donner ces droits sur les différentes fonctions de *Gitea*, gestionnaire de ticket, wiki ... De cette façon, on peut par exemple créer une équipe capable de trier les tickets et maintenir le wiki, mais pas de pousser du code sur la branche.
- *Accès Administrateur* sans commentaires, c'est bien expliqué.

## Les tickets

Le système de tickets s'apparente à un forum spécifique à chaque dépôt ayant pour but notamment de signaler des bugs, proposer des améliorations. Il se traduira donc souvent en tâches à effectuer. Plusieurs mécanismes permettent d'organiser ces tâches, comme les *étiquettes*, les *jalons*, la gestion des tickets qui en dépendent ou l'attribution des tickets.

## Les étiquettes

Ces étiquettes vont permettre de classer les tickets par thème. Sur un nouveau dépôt, il faut créer les étiquettes. On peut générer un jeu d'étiquettes par défaut avec Utiliser le jeu de label.



Pour les administrateurs du serveur, on peut ajouter des jeux d'étiquettes dans `custom/options/label` qui respecte le format `#hex-color label name ; label description 😊`



Pour ceux qui en ont les droits, on peut ensuite directement ajouter une étiquette lors de la création du ticket, ou une fois le ticket créé (même, par une autre utilisateur).

Lorsqu'une tâche est terminée, il est possible de la fermer.

On peut effectuer des recherches à partir de ces différents mécanismes pour trier les tickets, et ainsi déterminer l'ordre dans lequel effectuer ces tâches dans la liste des tickets en utilisant le menu déroulant de chaque colonne.

## Les jalons

Les jalons vont permettre de trier les tickets par objectifs. Par exemple, des tickets d'amélioration qu'on souhaite voir faire partie de la version *Tartempion* de notre logiciel pourront recevoir le jalon *Tartempion*. On peut donner des *Échéances* (date butoirs) aux *Jalons*.

## Affectation

Afin de savoir qui travaille sur un ticket, il est possible de les affecter à un ou plusieurs membres.

## Dépendance

La réalisation d'une tâche peut dépendre de la réalisation d'une autre. On peut définir que le ticket actuel dépend d'un (ou plusieurs autres)

Lorsqu'une tâche est terminée, il est possible de la fermer.

On peut effectuer des recherches à partir de ces différents mécanismes pour trier les tickets, et ainsi déterminer l'ordre dans lequel effectuer ces tâches.

## Mention fermeture depuis un message de commit

On peut mentionner et même fermer un ticket depuis un message de commit, il suffit de mettre son

numéro précédé de # pour le mentionner, et close #numéro pour le fermer.



Pour ceux qui en ont les droits, toutes les affectations, étiquetage, jalonage peuvent s'effectuer soit sur le ticket lui-même, soit depuis la liste des tickets en utilisant la case à cocher

## Le wiki

On peut créer des pages au format Markdown. Il existe cependant un éditeur qui va générer les balises. Ce wiki est en lui même un dépôt git, et on peut donc parfaitement l'écrire en local sur sa machine.



Si une page est nommée Ma page, l'URL générée est le nom de la page avec les espaces remplacés par des tiret - dans wiki. C'est assez pratique pour générer des liens entre page du wiki, puisqu'un simple lien relatif suffit. Ainsi, si je veux créer un lien de la page Home vers la page Ma Page, il suffit de mettre ça dans Home

```
[[Lien vers ma page|Ma-page]]
```

## Les versions

Une *version* est une publication du code source du logiciel sous forme d'archive (zip et tar.gz). Pour publier une version, il suffit de se rendre dans l'onglet *Version* et de cliquer sur *Nouvelle version*.


## Les demandes d'ajout



Cette partie va traiter d'une fonctionnalité plus fortement imbriquées avec git et les branches notamment. Il faut donc être un minimum à l'aise avec ces notions

L'intérêt des logiciels libres, c'est que tout un chacun peut obtenir et modifier le code source. Il est donc tout à fait possible de cloner un dépôt modifier le logiciel, et proposer ses modifications à son mainteneur.

### Bifurquer

Pour cela, il va falloir créer un nouveau dépôt dont vous serez propriétaire avec  Vous pouvez ensuite cloner puis travailler sur ce dépôt et y pousser vos modifications.

## Demander l'ajout

Lorsque vous pensez votre modification prête à être ajoutée au code principal, vous pouvez donc faire une *Demande d'ajout* (ou demande de fusion) afin que le mainteneur puisse fusionner votre travail dans le dépôt principal. Cela crée automatiquement un *ticket* dans l'onglet *Demande d'ajout*.

De son côté, le mainteneur va recevoir une notification, et va donc pouvoir décider si ces modifications peuvent être ajoutées en autorisant la fusion. Le ticket est alors automatiquement fermé.

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

2)

<https://git-scm.com/book/fr/v2>

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/atelier:git-df:gitea>

Last update: **03/07/2020 19:41**

