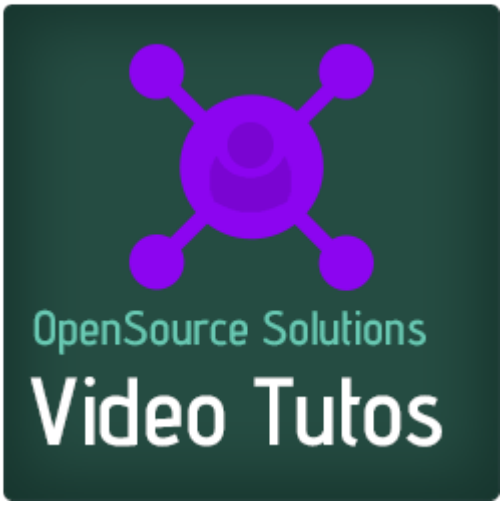
Tiki et le workflow Git

Tiki et le workflow Git

 Bernard Sfez - 2019-07-18 20:39



Dans cette vidéo j'aborderai :

- Les options d'installation de Tiki
- Comment installer Git sur votre ordinateur
- Comment récupérer une branche Tiki depuis Git sans tout l'historique
- Comment mettre à jour, comment vérifier le statut, etc.
- Le flux de travail Git pour les développeurs
- Ce qui change pour les contributeurs Tiki utilisant SVN
- Comment créer votre fork et configurer votre branche
- Comment commiter et soumettre vos modifications
- Comment créer une merge request et la soumettre pour approbation
- Comment maintenir votre branche à jour avec le reste du code
- Cherry-Picking (rétroportage)
- Modifier votre message de commit avant le push

Bonjour internet, je suis Bernard Sfez, spécialiste de Tiki. Dans cette vidéo nous allons voir ensemble comment utiliser Git avec Tiki.
Mais d'abord une astuce rapide à propos de Tiki.
Vous pourriez avoir besoin d'informations et même de diagnostics sur la configuration et les paramètres de votre serveur où vous avez installé un Tiki. Les développeurs de Tiki ont intégré un script autonome qui vérifie plus de 100 éléments différents et fournit un retour contextuel.
Connectez-vous en tant qu'administrateur dans votre Tiki, allez dans vos Panneaux de Contrôle et dans la barre de navigation Admin sous l'élément Outils, cliquez sur Vérification du Serveur.
Si vous trouvez cette astuce utile et que vous aimez ma vidéo, cliquez simplement sur le bouton j'aime et abonnez-vous à ma chaîne. Si vous avez une astuce que vous souhaitez partager avec les autres Tikiers, ajoutez-la dans les commentaires de cette vidéo.

Maintenant, commençons ce Tutoriel Express Tiki sur Tiki et Git avec la participation de Fabio

Montefuscolo, un type formidable, développeur Tiki qui a été très impliqué dans le passage de la communauté Tiki à Git et qui a en fait écrit toute la documentation sur Tiki et Git. Cette documentation que j'ai utilisée pour ce tutoriel est disponible sur : https://dev.tiki.org/Git

Cette vidéo est divisée en deux parties.
La première section pour les non-développeurs du noyau et la seconde pour les développeurs Tiki.
Bien que je vous encourage à regarder les deux, vous pouvez vous arrêter à la première section ou sauter directement à la seconde.
Dans la première section :

- Les options d'installation de Tiki
- Comment installer Git sur votre ordinateur
- Comment récupérer une branche Tiki depuis Git sans tout l'historique
- Comment mettre à jour, comment vérifier le statut, etc.

La deuxième section concerne le flux de travail Git pour les développeurs.

- Le flux de travail Git pour les développeurs
- Ce qui change pour les contributeurs Tiki utilisant SVN
- Comment créer votre fork et configurer votre branche
- Comment commiter et soumettre vos modifications
- Comment créer une merge request et la soumettre pour approbation
- Comment maintenir votre branche à jour avec le reste du code
- Cherry-Picking (rétroportage)
- Modifier votre message de commit avant le push

Options d'installation de Tiki

Utiliser Git est l'une des options tout comme SVN mais, nous encourageons les non-développeurs à installer Tiki en utilisant les packages réguliers publiés tous les 3 mois et disponibles sur : https://tiki.org/download ou si vous avez besoin d'une version plus récente, vous pouvez utiliser l'archive tarball de pré-version de Tiki construite en continu toutes les six heures sur : https://dev.tiki.org/Daily-Build
Certains utilisateurs préféreront utiliser une version de Tiki mise à jour en continu et il y a des raisons valables à cela. Parce qu'ils ont besoin d'une fonctionnalité fraîche ou bêta qui évolue, parce qu'ils préfèrent intégrer les correctifs au fur et à mesure, etc. Utiliser Git pour mettre à jour continuellement votre Tiki est possible un peu comme nous lancions précédemment le programme php "svnup.php" dans le dossier Doc Devtools ou en utilisant le shell avec la commande "svn update".
Alors installons Git sur votre ordinateur.

Comment installer Git sur votre ordinateur

Vérifions d'abord si git n'est pas déjà installé en utilisant votre shell, terminal sur votre machine locale ou en utilisant votre accès ssh pour un serveur distant. Si vous ne savez pas de quoi je parle, c'est le signe que vous devriez vous arrêter ici car utiliser git nécessite un minimum de

connaissances.
Ce tutoriel et les informations données sont orientés développeurs ou développeurs potentiels et si ce n'est pas votre cas, vous pouvez et devriez vraiment installer Tiki en utilisant le package régulier publié sur tiki.org/download et tout ira bien.
Donc dans votre Terminal, en utilisant :

```
git —version
```

vous verrez quelle version de git votre système exécute. Il s'exécute sur mon ordinateur et je peux voir la version mais si aucun n'est en cours d'exécution sur le vôtre, alors vous devez installer git en utilisant le lien que je donne sous cette vidéo sur atlassian point com et utiliser l'installateur ou le package correspondant pour votre système. Après l'installation, exécutez à nouveau

```
git —version
```

pour confirmer que votre git est correctement installé.

Comment récupérer une branche Tiki depuis Git sans tout l'historique

Une fois cela fait, vous pouvez utiliser Git sur votre système en utilisant n'importe quelle interface graphique disponible ou même depuis votre IDE comme je le fais. Pour ce tutoriel, je vais continuer à utiliser mon terminal. Il est important de comprendre la commande et le flux de travail, ensuite vous pourrez utiliser n'importe quel outil que vous préférez. Donc, avec git installé, clonons, créons une copie de travail de Tiki depuis le dépôt officiel de Tiki. Le dépôt Tiki est disponible sur GitLab : https://gitlab.com/tikiwiki/tiki Le package complet fait un peu moins de 3 Go et contient tout l'historique et les commits effectués dans Tiki. Vous pouvez tout récupérer ou juste une partie en fonction de vos besoins. Pour cloner une branche spécifique de Tiki sans tout l'historique, vous devez ajouter des paramètres pour limiter le téléchargement. Par exemple, si vous voulez créer un Tiki 20 avec un minimum d'historique dans un dossier nommé "mytiki20", vous pouvez taper dans votre terminal :

```
git clone --branch=20.x --depth=1 https://gitlab.com/tikiwiki/tiki.git mytiki20
```

Après quelques minutes, votre copie de travail de Tiki sera prête à être configurée et connectée à la base de données comme d'habitude.

Comment mettre à jour, comment vérifier le statut, etc.

Pendant que vous avez votre Tiki installé, le dépôt Tiki continuera de changer et d'être mis à jour. Pour voir les différences entre votre copie de travail et le dépôt distant, vous devez récupérer les changements et indiquer ce que vous voulez comparer et avec quoi. Laissez-moi vous montrer comment. Je tape

```
git fetch
```

, puis pour être sûr d'utiliser le bon nom, je fais un

```
git branch -a
```

pour voir toutes les branches disponibles et voir sur quelle branche je suis. Ici je suis sur 20.x. Oh et par défaut git utilise l'éditeur VIM qui nécessite que vous tapiez "q" pour quitter l'éditeur. Une fois que j'ai toutes les informations, je peux utiliser la commande diff pour faire la comparaison :

```
git diff 20.x origin/20.x
```

Je peux voir que quelque chose de nouveau a été commité et je peux mettre à jour ma copie de travail en utilisant

```
git pull
```

Je peux aussi récupérer aveuglément les changements et mettre à jour sans vérifier en faisant git fetch puis

```
git pull
```

C'est tout pour cette première partie, c'est très très basique mais j'espère que cela vous donnera assez de pistes pour commencer à apprendre et utiliser Git avec Tiki.
Commençons maintenant la deuxième partie de la vidéo qui donnera plus de détails et d'informations précieuses pour les contributeurs Tiki.

Flux de travail Git

Examinons cela en détail. Git est un système de contrôle de version distribué et est utilisé pour implémenter des contributions supplémentaires à des millions de projets dans le monde. Certains d'entre vous l'utilisent déjà et il y a beaucoup d'informations sur Git sur internet mais il reste essentiel de comprendre qu'en utilisant Git, tout développeur pourra soumettre ses modifications au projet Tiki et que le code modifié sera disponible pour que tout le monde puisse le voir, le comparer et le tester avant d'être fusionné dans la branche correspondante de Tiki.

Ce qui change pour les contributeurs Tiki utilisant SVN

Avec Git sur Tiki, vous avez le dépôt Tiki (upstream), votre fork du projet et votre copie de travail. Vous pousserez vos modifications vers votre fork puis soumettrez vos modifications pour approbation. Une fois approuvée, votre modification sera fusionnée de votre fork vers le dépôt Tiki.
Pour résumer, vous devez créer votre fork utilisateur GitLab depuis le dépôt Tiki.

Vous créez votre copie de travail depuis votre fork sur votre ordinateur local.
Vous codez et éditez sur votre copie de travail locale.
Vous commitez également sur votre copie de travail locale.
Une fois que vous avez terminé, vous poussez vos modifications vers votre fork.
Ensuite, vous créez une nouvelle demande de fusion et participez au processus d'approbation.
C'est le processus, faisons-le maintenant.

Comment créer votre fork et configurer votre branche

Connectez-vous ou inscrivez-vous sur GitLab et allez sur le projet Tiki.
Cliquez sur le bouton Fork en haut de la page. J'ai déjà un fork donc ça n'en créera pas un mais la première fois ça devrait prendre entre 10 et 20 minutes.
Maintenant depuis mon propre dépôt, je vais créer un clone local.
De ce clone qui contient toutes les versions de l'historique et branches de Tiki, je pourrai faire un checkout d'une copie de travail pour n'importe quelle branche qu'il contient. Par défaut, il est réglé sur master (trunk). Je vais faire un checkout (basculer) vers 20.x en tapant :

```
git checkout 20.x
```

Et au passage, je peux voir si tout est à jour avec la branche 20.x originale ou si quelque chose a été modifié. Ici je peux voir qu'il y a eu un commit sur un fichier. Comme nous l'avons vu précédemment, je peux mettre à jour ma copie de travail en faisant git fetch puis git pull.
Maintenant, faisons comme si je voulais corriger quelque chose. Je vais créer un checkout différent en utilisant un nom qui signifie quelque chose pour que les autres développeurs puissent voir, comprendre de quoi il s'agit.

```
git checkout -b fixing-something
```

En faisant cela, j'ai créé une branche sur mon dépôt local donc je peux commencer à éditer des fichiers.

Comment commiter et soumettre vos modifications

Maintenant je peux modifier un fichier de ma copie de travail. Je vais éditer le fichier README.
Enregistré.

Je vérifie avec git status et je peux voir que j'ai un fichier modifié dans ma copie de travail mais pas encore dans la zone de staging. Il est affiché en rouge.

```
git add README
```

ajoutera ce fichier à la zone de staging.

```
```

```
git status
```

à nouveau et maintenant le nom du fichier est affiché en vert.
Je commite le fichier vers mon dépôt local et j'ajoute un message de commit, encore une fois mais important, ceci est local uniquement, aucun changement n'est encore envoyé vers notre dépôt fork distant. Je vais utiliser

```
git commit -m "[FIX] Ceci est ma correction du fichier README"
```

tagué selon la convention Tiki.
J'entre git log et je peux voir les derniers fichiers commités et le dernier est celui que je viens de commiter avec son numéro de hash. Nous verrons plus tard comment ce numéro de hash sera utilisé pour un cherry-picking, pour ajouter cette modification sur différentes branches.
Maintenant je peux pousser ma branche avec mes modifications vers mon fork au point d'origine de mon dépôt local, le dépôt distant sur GitLab en utilisant :

```
git push -u origin fixing-something
```

Vous aurez les informations importantes affichées et ici tout a bien fonctionné. Avec git et GitLab, vous aurez de l'aide tout au long du processus. Ici vous pouvez voir qu'ils ont déjà prévu que vous devriez créer une demande de fusion et donnent quelques conseils. De retour sur GitLab, je fais un rafraîchissement et je peux voir une mise à jour d'activité concernant le commit que je viens de pousser. Dans le sélecteur de branche, ma nouvelle branche est là aussi.

Comment créer une merge request et la soumettre pour approbation

Pour créer une demande de fusion, je clique sur le bouton vert Créer une Merge Request. Cela ouvre un formulaire où je peux voir des champs que je peux remplir pour soumettre mon commit pour approbation au reste de la communauté. Le processus d'approbation n'est pas automatique, d'autres développeurs peuvent discuter de votre commit, suggérer des modifications et approuver pour qu'il soit fusionné dans le code principal.

Comment maintenir votre branche à jour avec le reste du code

Comme nous le comprenons, ce processus n'est pas terminé lorsque vous soumettez votre demande de fusion, le code de Tiki peut changer. Dans GitLab, il y a une fonction de mise en miroir des dépôts qui maintiendra votre branche à jour. Sur votre fork, vous allez dans Paramètres, Dépôt où il est indiqué Mise en miroir des dépôts, cliquez sur le bouton développer. Dans le champ "URL du dépôt Git", collez l'URL du dépôt du projet Tiki. Il se mettra à jour immédiatement puis régulièrement.

Cherry-Picking (rétroportage)

Il n'y a pas de backport dans Git et il y a le cherry picking. Le cherry picking dans Git signifie choisir un commit d'une branche et l'appliquer sur une autre. Cela contraste avec d'autres

méthodes comme merge et rebase qui appliquent normalement plusieurs commits sur une autre branche.
Pour faire un cherry picking, vous avez besoin du hash du commit que vous voulez appliquer sur une autre branche. Vous pouvez le trouver sur GitLab juste à côté de la ligne de commit ou dans votre git log dans la branche où vous l'avez commité. Faisons un cherry-picking (backport) de notre modification de la branche 20.x vers 19.x.
Je récupère le numéro de hash de mon commit depuis mon log :

```
git log
```

Je le copie. Ensuite, je fais un checkout de la branche 19.x sur mon dépôt local pour définir la copie de travail. Et ici je tape

```
git cherry-pick et je colle le numéro de hash
```

Bien sûr, mon test n'a pas été repoussé vers le dépôt distant donc je ne peux pas terminer et taper entrée, mais vous avez compris l'idée.

Modifier votre message de commit avant le push

La communauté Tiki garde une trace du backport en ajoutant au début du message de commit le numéro de révision SourceForge. Il n'y a pas encore de mécanisme pour aider avec cela et vous devrez modifier manuellement et amender votre message de commit avant de le pousser au moment où j'ai publié ce tutoriel.
Tant que votre commit n'a pas encore été poussé, vous pouvez modifier le message de commit et ajouter le numéro de révision SourceForge au message de commit. Pour ce faire, entrez :

```
git commit --amend
```

À partir de là, cela lancera l'éditeur vi ou tout autre éditeur si vous avez modifié le paramètre par défaut de git et ouvrira votre liste de commits en attente. Ici, je clique sur i pour éditer le texte.
J'ajoute un faux numéro de révision et j'enregistre en quittant :wq
Je vérifie que c'est fait en utilisant git log
Tout est bon.

C'est tout pour ce tutoriel vidéo. Il existe plusieurs autres commandes git qui peuvent être utiles

```
git branch -r
git log -p
git log -p monfichier
git rebase
```

```
git reset
git reset --hard
git remote show origin
git diff
git revert
git blame mon_fichier
git config --global core.editor "nano --wait"
```

ainsi que beaucoup de paramètres ou même l'option partagée décrite sur les pages de documentation dev de Tiki que nous avons vues précédemment.
Vous avez beaucoup de documentation en ligne et vous trouverez plus que ce dont vous avez besoin selon le cas que vous essayez de résoudre. C'est trop important pour un Tutoriel Express Tiki donc cherchez sur Google.
De plus, votre IDE comme phpstorm intègre l'interface et les commandes Git donc peut-être que je ferai une vidéo à ce sujet bientôt.

S'il vous plaît, si vous aimez cette vidéo, cliquez sur l'icône j'aime et partagez partout où vous pensez que cela devrait l'être.
Si vous ne voulez pas manquer mon prochain tutoriel et voulez être notifié quand je publie une nouvelle vidéo, cliquez sur la cloche, le bouton d'abonnement de ma chaîne youtube.

Merci encore d'avoir regardé et que la puissance de Tiki soit avec vous.