

Commandes Git de base

Voici une liste de quelques commandes Git de base pour vous aider à utiliser Git.

Pour plus de détails, consultez les [didacticiels Atlassian Git](#) pour une présentation visuelle des commandes et des workflows Git, ainsi que des exemples.

Tâche git	Remarques	Commandes Git
Dites à Git qui vous êtes	Configurez le nom de l'auteur et l'adresse email à utiliser avec vos commits. Notez que Git enlève certains caractères (par exemple, des points de fin) <code>user.name</code> .	<code>git config --global user.name "Sam Smith"</code> <code>git config --global user.email sam@example.com</code>
Créer un nouveau référentiel local		<code>git init</code>
Extraire un dépôt	Créez une copie de travail d'un référentiel local:	<code>git clone / chemin / vers / référentiel</code>
	Pour un serveur distant, utilisez:	<code>git clone nom d'utilisateur @ hôte: / chemin / ver</code>
Ajouter des fichiers	Ajouter un ou plusieurs fichiers à la mise en scène (index):	<code>git add <nom du fichier></code> <code>git add *</code>
Commettre	Valider les modifications en tête (mais pas encore dans le référentiel distant):	<code>git commit -m "Message de validation"</code>
	Commencez tous les fichiers que vous avez ajoutés <code>git add</code> , ainsi que tous les fichiers que vous avez modifiés depuis:	<code>git commit -a</code>
Pousser	Envoyez les modifications à la branche principale de votre référentiel distant:	<code>maître d'origine git push</code>
Statut	Répertoriez les fichiers que vous avez modifiés et ceux qu'il vous reste à ajouter ou à valider:	<code>statut git</code>
Se connecter à un référentiel distant	Si vous n'avez pas connecté votre référentiel local à un serveur distant, ajoutez le serveur pour pouvoir y accéder:	<code>git remote add origin <server></code>
	Répertoriez tous les référentiels distants actuellement configurés:	<code>git remote -v</code>
Branches	Créez une nouvelle branche et passez à celle-ci:	<code>git checkout -b <nom de branche></code>
	Passer d'une branche à une autre:	<code>git checkout <nom de branche></code>
	Répertoriez	<code>branche de git</code>

Contenu connexe

Ressources Git

Télécharger une archive de Bitbucket Server

Importation de code à partir d'un projet existant

Valider et transmettre les modifications à Bitbucket Server

Intégration de Bitbucket Server aux applications Atlassian

Encore besoin d'aide?

La communauté Atlassian est là pour vous.

[Demander à la communauté](#)

	<p>Récupérez toutes les branches de votre référentiel et vous indiqueriez également dans quelle branche vous vous trouvez actuellement:</p>	<pre>git branch -a</pre>
	Supprimer la branche de fonctionnalité:	<pre>git branch -d <nom de branche></pre>
	Poussez la branche dans votre référentiel distant pour que d'autres puissent l'utiliser:	<pre>origine de git push <nom de branche></pre>
	Poussez toutes les branches dans votre référentiel distant:	<pre>git push --all origine</pre>
	Supprimer une branche de votre référentiel distant:	<pre>origine de git push: <nom de branche></pre>
Mise à jour depuis le référentiel distant	Récupérez et fusionnez les modifications sur le serveur distant dans votre répertoire de travail:	<pre>git pull</pre>
	Pour fusionner une autre branche dans votre branche active:	<pre>fusion de git <nom de branche></pre>
	<p>Voir tous les conflits de fusion:</p> <p>Afficher les conflits avec le fichier de base:</p> <p>Prévisualisez les modifications avant de fusionner:</p>	<pre>git diff</pre> <pre>git diff --base <filename></pre> <pre>git diff <sourcebranch> <targetbranch></pre>
	Après avoir résolu manuellement les conflits, vous marquez le fichier modifié:	<pre>git add <nom du fichier></pre>
Mots clés	Vous pouvez utiliser le marquage pour marquer un ensemble de modifications significatif, tel qu'une version:	<pre>balise git 1.0.0 <commitID></pre>
	Commitid est les caractères principaux de l'ID de changeset, jusqu'à 10, mais doit être unique. Obtenez l'ID en utilisant:	<pre>journal git</pre>
	Poussez toutes les balises dans le référentiel distant:	<pre>git push --tags origine</pre>
Annuler les modifications locales	<p>Si vous vous trompez, vous pouvez remplacer les modifications dans votre arbre de travail par le dernier contenu en tête:</p> <p>Les modifications déjà ajoutées à l'index, ainsi que les nouveaux fichiers, seront conservés.</p>	<pre>git checkout - <nom du fichier></pre>

	<div>Au lieu de cela, pour supprimer toutes les modifications et les validations locales, extraire l'historique le plus récent du serveur et y pointer votre branche maître locale, procédez comme suit:</div>	<div>git chercher l'origine</div> <div>Réinitialisation git - origine dure / maître</div>
Chercher	<div>Rechercher dans le répertoire de travail pour foo():</div>	<div>git grep "foo()"</div>

Dernière modification le 21 sept. 2017

Est-ce que cela a été utile? ☐ Oui ☐ Non [Fournir des commentaires sur cet article](#)

Propulsé par [Confluence](#) et [Scroll Viewport](#).