

Git-Schnellreferenz

Stefan Gast

27. März 2009

1 Repository anlegen

Aufgabe	Befehl
Lokales Repository erzeugen	<pre>cd myproject git init git add . git commit</pre>
Öffentliches Repository anlegen	<pre>newgrp projektgruppe mkdir meinprojekt.git cd meinprojekt.git git init --bare</pre>
Öffentliches Repository mit Schreibzugriff für Gruppe	<pre>newgrp projektgruppe mkdir meinprojekt.git cd meinprojekt.git git init --bare --shared=group</pre>
Kopie eines Projektes erzeugen	<pre>git clone ssh://username@server.de/pfad</pre>

2 Informationen abfragen

Aufgabe	Git-Befehl	SVN-Befehl
Log anzeigen	<code>git log</code>	<code>svn log</code>
Änderungen seit letzter Übertragung	<code>git diff</code>	<code>svn diff</code>
Versionen vergleichen	<code>git diff <key1>..<key2></code>	<code>svn diff <re_a>:<re_b></code>
Status ausgeben	<code>git status</code>	<code>svn status</code>

3 Git für SVN-Benutzer

Schritt	Git-Befehl	SVN-Befehl
1. Neueste Version aus zentralem Repository holen	<code>git pull</code>	<code>svn update</code>
2. Bearbeiten der Dateien	-	-
3. ggf. Dateien hinzufügen	<code>git add <file1> <file2></code>	<code>svn add <file1> <file2></code>
4. Änderungen kontrollieren	<code>git diff</code>	<code>svn diff</code>
5. Neue Version an zentrales Repository schicken	<code>git commit -a</code> <code>git push</code>	<code>svn commit</code>

4 Änderungen rückgängig machen

Aufgabe	Befehl
Arbeitskopie zurücksetzen	<code>git reset --hard HEAD</code>
Commit zurücknehmen	<code>git revert <key></code>

5 Tags

Aufgabe	Befehl
Tag erzeugen	<code>git tag <tagname> [<key>]</code>
Tags anzeigen	<code>git tag</code>
Tag löschen	<code>git tag -d <tagname></code>

6 Arbeiten mit mehreren Entwicklungszweigen

Aufgabe	Befehl
Zweig anlegen	<code>git branch <name> [<start>]</code>
Zweige anzeigen	<code>git branch</code>
Zweig wechseln	<code>git checkout <zweig></code>
Zweige zusammenführen	<code>git checkout <ziel></code> <code>git merge <quelle></code>
Zweig löschen	<code>git branch -d <zweig></code>
Neuen lokalen Zweig an öffentliches Repository schicken	<code>git push <remote> <neuer_zweig></code>

7 Verteilte Entwicklung

Aufgabe	Befehl
Mit anderem Repository verbinden	<code>git remote add <name> <adresse></code>
Remote Branches auf neuesten Stand bringen	<code>git fetch <remote_name></code>
Änderungen ansehen	<code>git diff HEAD..<remote>/<zweig></code>
Änderungen in aktuellen lokalen Zweig übernehmen	<code>git merge <remote>/<zweig></code>