

Einrichtung & Benutzung von GIT

Git ist ein [Version Control System \(VCS\)](#) das dazu dient die Versionshistorie unserer Plugins und Config Dateien zu überwachen. Somit ist es uns möglich einen einzigartigen Update Prozess zu Verfügung zu stellen der die Downtime und Bugs auf dem Server minimiert.

Hier ein kurzer Auszug von der git-scm.com/ Homepage.

Git-SCM wrote:

Git is a [free and open source](#) distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is [easy to learn](#) and has a [tiny footprint with lightning fast performance](#). It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like [cheap local branching](#), convenient [staging areas](#), and [multiple workflows](#).

Table Of Contents

- [1 Installation von Git](#)
- [2 Benutzung von Git im Windows Explorer](#)
 - [2.1 Repository Clonen \(Lokale Kopie erstellen\)](#)
 - [2.2 Änderungen Synchronisieren](#)

Installation von Git

Damit Git verwendet werden kann muss es wie jedes andere Programm erst einmal installiert werden. Anschließend muss man Git mitteilen wie der Username und E-Mail Adresse lautet. Nutzt hier am besten euren Foren oder InGame Namen und auf jeden Fall die E-Mail mit der ihr euch im Gitlab registriert habt.

1. Download & Installation von Git ([Downloadlink: https://git-scm.com/](https://git-scm.com/))
2. Registrierung auf git.raid-craft.de/users/sign_in - Benutzt hier euren Foren oder InGame Name und die gleiche E-Mail wie im Forum
3. Festlegen der Benutzerdaten
 1. Öffnet nach der Installation von Git auf eurem PC die Git Bash (über Start -> Suchen)
 2. Gebt in der Git Bash folgendes ein: `git config --global user.name "Euer Nickname"`, z.B.:
`git config --global user.name "Silthus"`
 3. Anschließend müsst ihr noch eure E-Mail angeben: `git config --global user.email "eure@mail.com"`, z.B.:
`git config --global user.email "m.silthus@gmail.com"`
4. Generieren eines SSH Keys zur Authentifizierung mit dem Gitlab Server
 1. Öffnen der Git Bash (über Start -> Suchen)
 2. In der Git Bash folgendes eingeben um einen Private & Public Key zu erzeugen: `ssh-keygen`
 3. Kein Challenge Password eingeben und alles mit YES bestätigen. Der Public & Private Key wird im `C:\Users\DeinUser\.ssh\` erzeugt.
Der Private Key heißt `id_rsa` und der Public Key `id_rsa.pub` Der Public Key wird später in eurem Gitlab Profil eingetragen. Den Private Key solltet ihr gut unter Verschluss halten und nicht an andere senden da er euch eindeutig identifiziert und euch berechtigt auf unser Gitlab zuzugreifen.
 4. Öffnet nun den Public Key `id_rsa.pub` mit einem Text Editor und kopiert den gesamten Inhalt (STRG+A & STRG+C)
 5. Logt euch im Gitlab ein und ruft folgenden Link auf: git.raid-craft.de/profile/keys/new
 6. Fügt den gerade kopierten Key dort in das Key Feld ein (STRG+V)
 7. Da Tortoise Git einen Putty Key (.ppk) benötigt muss der Private Key erst noch konvertiert werden. Ladet euch dazu PuttyGen herunter: the.earth.li/~sgtatham/putty/latest/x86/puttygen.exe
 8. Öffnet nun PuttyGen, geht auf Conversions -> Import Key -> Wählt hier den `id_rsa` Key aus dem `C:\Users\DeinBenutzer\.ssh\id_rsa.pub` aus
 9. Speichert anschließend den Private Key `Save private key als id_rsa.ppk`
5. Download & Installation von Tortoise Git (Downloadlink: <https://code.google.com/p/tortoisegit/>)

raid-craft.de/index.php?attach...91076af3e323711f0c385f366
raid-craft.de/index.php?attach...91076af3e323711f0c385f366

Git ist nun vollständig installiert und kann sowohl über IntelliJ IDEA (Entwickler IDE) und über das Windows Context Menü im Explorer (Rechtsklick) verwendet werden.

Benutzung von Git im Windows Explorer

Das Tortoise Git Context Menü kann jederzeit im Windows Explorer über einen Rechtsklick aufgerufen werden.

Repository Clonen (Lokale Kopie erstellen)

Um nun ein zentrales "Repository" lokal zu "clonen" und damit zu arbeiten muss folgendes durchgeführt werden.

1. Suchen des entsprechenden Projekts in Gitlab, z.B. Quests (die Suchleiste ist oben rechts)
2. Auswahl der SSH Clone URL
3. Lokal im Windows Explorer in einem beliebigen Verzeichnis (z.B. C:\Raid-Craft\Configs\)) Rechtsklick -> `Git clone`
4. Bei der URL die gerade kopierte URL einfügen
5. Bei Putty Key den vorher generierten `Putty Private Key` auswählen

Nach einem Klick auf OK wird eine lokale Kopie des Repositories erstellt.

raid-craft.de/index.php?attach...91076af3e323711f0c385f366raid-craft.de/index.php?attach...91076af3e323711f0c385f366

Anschließend sollte noch unbedingt der richtige "Branch" ausgecheckt werden da alle Änderungen immer auf dem Testserver stattfinden müssen. Dazu in dem gerade erstellten (geclonten) Ordner einen Rechtsklick machen: Tortoise Git -> Switch/Checkout und dort den `remotes/origin/test` Branch wählen.

raid-craft.de/index.php?attach...91076af3e323711f0c385f366raid-craft.de/index.php?attach...91076af3e323711f0c385f366

Änderungen Synchronisieren

Damit die Änderungen von anderen Team Mitgliedern auch lokal sichtbar sind muss das Git Repository synchronisiert werden. Dazu im vorher "geclonten" Ordner einen Rechtsklick machen und auf `Git Sync...` klicken. Anschließend `Pull` auswählen. Nun hat man alle Änderungen die Remote vorgenommen wurden auch lokal. Es ist weiterhin wichtig auf dem `test` Branch zu arbeiten da sonst die jeweiligen Änderungen nicht auf den Testserver übernommen werden.

raid-craft.de/index.php?attach...91076af3e323711f0c385f366

Nun kann man Änderungen an den Config Dateien vornehmen, neue hinzufügen und bestehende verschieben oder löschen. Zum Beispiel kann man nun neue Quests hinzufügen oder an bestehenden weiter arbeiten. In meinem Beispiel füge ich eine neue Quest für die Welt Faldoria hinzu. Für eine detaillierte Anleitung wie man Quests erstellt gibt es das super Tutorial von [@Kiachan](#) -> Quest Konfiguration.

Sobald man alle Änderungen vorgenommen hat und diese auf dem Testserver testen will müssen die Änderungen eingchecked und "gepusht" werden. Dazu im Ordner einen Rechtsklick machen und auf `Git Commit...` klicken. In das Message Feld eine kurze prägnante Nachricht eingeben was geändert wurde und über den Knopf `All` alle neuen oder entfernten Dateien hinzufügen. Außerdem sollte der Haken bei `Whole Project` gesetzt werden damit auch Dateien außerhalb des aktuellen Ordners mit einbezogen werden. Sobald man auf `Commit` gedrückt hat öffnet sich ein weiteres Fenster dort auf `Push` drücken um die lokalen Änderungen auf dem Server verfügbar zu machen.

raid-craft.de/index.php?attach...91076af3e323711f0c385f366raid-craft.de/index.php?attach...91076af3e323711f0c385f366

Anschließend muss jemand vom Technik Team die Änderungen auf den Testserver ziehen und diesen durchstarten.