

Kontribuieren und publizieren auf GitHub im Namen der Telekom

Die Deutsche Telekom nutzt (auch) [GitHub](#) zur Veröffentlichung ihrer eigenen Open-Source-Projekte (**Publishing**). Als 'Dach' dafür pflegt sie eine Telekom-GitHub-Organisation <https://github.com/telekom>. Zudem begrüßt es die Deutsche Telekom, wenn ihre Mitarbeiter von der Firma genutzte Open-Source-Software weiterentwickeln und das Ergebnis an die (GitHub)-Projekte der Community zurückgeben (**Contribution**).

In beiden Fällen müssen Mitarbeiter für diesen Teil ihrer Arbeit einen GitHub-Account nutzen. Es hat sich bereits mehrfach herausgestellt, dass die Nutzung von zwei verschiedenen Accounts - einem privaten und einem beruflichen - sehr unhandlich ist. Dieses Dokument erklärt daher, wie Angestellte einen **auch** im beruflichen Kontext nutzbaren *privaten* GitHub-Account anlegen und/oder einrichten.

Allgemeines vorab.

Wer geistiges Eigentum der Deutschen Telekom als Open-Source-Software kontribuiert, kann dies nur im Rahmen seiner Telekom-Tätigkeit und nur im Auftrag der Telekom tun. Sonst wäre es illegal. Für die Open-Source-Community ist es umgekehrt wichtig, dass die legale Freigabe dokumentiert wird. Denn sie soll ja davon ausgehen können, dass sie die per Lizenzierung zugesagten Rechte auch wirklich von der Telekom erhalten haben.

Für die Publikation und Kontribution haben wir Leitfäden erstellt. Darin enthalten ist auch die Aufgabe, dass sich die Kontributoren / Developer die Zustimmung dazu bei ihren Linienvorgesetzten und/oder Projektleitern holen. Eine solche Zustimmung schließt die Zustimmung ein, GitHub-Accounts zu nutzen.

Trotzdem benötigen wir Regeln, wie wir in den Accounts kenntlich machen, dass wir (gerade) als Telekommitarbeiter arbeiten.

Den GitHub-Account einrichten!

Die deutsche Telekom erlaubt ihren Mitarbeitern, ihre 'privaten' GitHub-Accounts auch für die Telekomarbeit zu nutzen. Sie zwingt sie aber nicht dazu: Ihr Mitarbeiter darf sich auch einen spezifischen Telekom-GitHub-Mitarbeiter-Account anlegen. Aber die Deutsche Telekom möchte, dass die genutzten Accounts, seien sie 'privat' oder 'spezifisch', auf eine bestimmte Weise eingerichtet werden. Dazu gehören folgende Schritte:

- **Einen GitHub-Account aktivieren**, indem man
 - einen neuen Account generiert (Option SIGN-UP bei GitHub) *[der eigentliche Accountname darf durchaus kryptisch sein und den Rückschluss auf die Person verschleiern. Die Telekom wünscht sich aber natürlich, dass ihre Mitarbeiter auch nach außen zu ihr stehen.]*
 - oder aus seinen eigenen (privaten) Accounts den auswählt, unter dem man den Telekomcode etc. auf GitHub publizieren will.
- ****Die eigene T-Email-Adresse (you@telekom.de / you@t-systems.de / you@magenta.at) hinzufügen****, indem man sie in seinem GitHub-Account rechts oben über den Bereich

Settings/Emails` einträgt. *[Ob überhaupt und wenn ja, welche Email-Adresse öffentlich sichtbar sein soll, darf jeder selbst entscheiden.]*

- **Den eigenen Arbeitgeber kenntlich machen**, in dem man sein Profil unter <http://github.com/der-eigene-account> editiert und dort als company 'Deutsche Telekom AG' etc. einträgt *[Gerne kann man daran (@telekom) anhängen. So wird automatisch ein Bezug zur entsprechenden GitHub-Organisation erzeugt. Alle anderen Angaben (Photo, Beschreibung, etc.) möge jeder Mitarbeiter so einrichten, wie es ihm nützlich erscheint und wie es den allgemeinen Verhaltensregeln der Telekom entspricht.]*

Den lokalen Git-Client konfigurieren

Whether you use your private or a work GitHub account, your work-related commits should always be made with your work e-mail address. If you are working with CLI this is very easy, simply run `git config --global user.email "YOUR_EMAIL"` in a terminal. If you use a git GUI client, it is very likely that the above command also works, if not please refer to the documentation of your git client.

Unabhängig davon, ob ein privates oder ein geschäftliches GitHub-Konto verwendet wird, sollten alle arbeitsbezogenen Commits immer mit der geschäftlichen E-Mail-Adresse erfolgen. Wenn die git CLI verwendet wird, ist dies sehr einfach. Der Terminal-Befehl `git config --global user.email "YOUR_EMAIL"` setzt die E-Mail die in den Commits verwendet wird. Wenn ein Git-GUI-Client verwendet wird, ist es sehr wahrscheinlich, dass der obige Befehl auch funktioniert. Wenn nicht, sollte die Dokumentation des Git-Clients die notwendigen Schritte enthalten.

Die Zwei-Faktor-Authentifizierung 'anknipsen'!

Bei einfachen Projekten loggt man sich mit dem Paar *account/passwort* unter <http://github.com/> ein. Dieses Paar nutzt man auch, wenn man auf der Kommandozeile per `git` kontribuiert. Würde dieser Account gehackt, wären die Projekte gefährdet, zu denen der Account kontribuieren kann. Für die Telekom bedeutete das, dass ein Fremder 'in ihrem Namen' spräche - was unerwünscht ist. Deshalb sollen sukzessive alle von Telekom-Mitarbeitern genutzt GitHub-Accounts auf die Zwei-Faktor-Authentifizierung umgestellt werden. Das meint Folgendes:

- **Sich mit dem entsprechenden GitHub-Account einloggen**
- **Alle Email-Adressen verifizieren**, indem man unter Settings/Emails die entsprechenden Schritte ausführt
- **Die 2FA (Zweifaktorauthentifizierung) anstellen**, indem man
 - **die app andOTP** (Android) oder den **Google Authenticator** (Android oder iOS) oder **FreeOTP** (Android oder iOS) **auf seinem Smartphone installiert** *[Achtung: die als Default angebotene App 'authy' nicht installieren und nutzen: sie ist selbst nicht Open-Source-Software und ihr Gebrauch könnte die Deutsche Telekom unter gewissen Umständen Geld kosten.]*
 - In seinem GitHub-Account unter Settings/Security **die 'two-factor authentication' anschalten**.
 - **Die Methode "Setup using an app" aktivieren**
 - Die angebotenen **Recovery Codes downloaden** und (z.B. in KeePass) sichern. *[Bei Verlust droht der Account dauerhaft unbenutzbar zu werden.]*
 - Den dann erscheinenden **QR-Code mit der installierten App einscannen** und
 - die von der App **zurückgegebene Nummer im GitHub-Anfragedialog eintippen**.

In Zukunft wird man beim erneuten Einloggen mit seinem GitHub-Account nicht mehr nur nach dem Passwort gefragt, sondern auch nach einer Nummer, die man sich aus der installierten App immer wieder neu abholen kann und muss. Nach dem Einloggen kann man die GitHub-Tools dann frei nutzen.

Einige Fallen.

- Man muss die von der App gelieferte Authentifizierungszahl recht zügig eingeben, weil sie in kurzen Abständen erneuert wird - was ja der Sinn ist, weil so der Missbrauch extrem erschwert wird.
- Wer an der Shell mit `git` arbeitet, wird bei Zugriffen auf das Remote-Repository üblicherweise nur nach seinem GitHub-Account und seinem normalen Passwort gefragt. Nach Aktivierung der Zwei-Faktor-Authentifizierung geht das nicht mehr. Stattdessen muss man sich ein von GitHub ein neues *personal access token* generieren lassen, das das bisherige Passwort ersetzt. Dazu muss man
 - im Settings-Dialog den Eintrag Developer-Settings aufrufen
 - im nächsten Dialog die Methode 'Personal Access Tokens' auswählen und im Fenster die Option 'Generate a personal access token' anklicken
 - im nächsten Dialog dem Token einen Namen geben, (mindestens) die Optionen 'repo' und 'delete- repo' auswählen und den Button 'Generate token' anklicken
 - den erschienenen Token mit Copy-&-Paste in einer Textdatei ablegen
 - bei einer Frage von git nach dem Account und dem Passwort anstelle des Passwortes den angelegten Token eingeben
- Wer den Access-Token nutzt, muss sorgsam mit C&P umgehen: Blanks vor und nachdem Token werden offensichtlich an der Shell mitausgewertet.
- Wer den Access-Token generiert, muss sorgsam die mit dem Token verbundenen Rechte setzen
- Wer seinen 'privaten' auch sonst genutzten Account mit der 2FA ausstattet, wird die Zwei-Faktor-Authentifizierung auch bei seinen privaten Accounts nutzen müssen, nicht nur bei Organisationen, die die 2FA-Nutzung fordern.

Anzeige von Mitgliedschaften in GitHub-Organisationen

Sehr oft arbeiten Projekte oder Firmen mit mehreren Repositories. Dann bietet es sich an, auf *GitHub-Organisationen* zu gründen und die Repositories darüber zu clustern. Eine solche Organisation ist <https://github.com/telekom>. Die einzelnen Repositories haben *Contributoren* oder *Collaborateure*, die Organisation kann auch davon unabhängig eigene *Mitglieder* aufweisen. Jedes Mitglied der Organisation kann für sich entscheiden, ob es seine Mitgliedschaft nach außen kundtun will.

Bei Mitarbeitern der Deutschen Telekom ist es gewünscht, dass sie in ihren GitHub-Accounts auch ihre Mitgliedschaft in der GitHub-Organisation *Telekom* anzeigen. Damit wird die Kooperation der Telekom mit der Community verdeutlicht. Hier die Schritte, die man tun muss, um seine Organisationsmitgliedschaft nach außen sichtbar werden zu lassen:

1. Sich mit seinem Account **auf GitHub einloggen**.
2. Rechts oben **das Account Drop Down Menu anwählen** (Klick auf Bild oder Muster) und **darin 'Your Profile' auswählen**.
3. **Bei den** dann links erscheinenden **Organisationen auf das Telekom Logo klicken**.
4. In dem dann erscheinenden Organisationsrepository **auf den Reiter 'People' klicken**.

5. **Den eigenen Account** händisch oder per Kommando **suchen lassen**.

6. In der Zeile den **Eintrag von *private* auf *public* umstellen**.

Links auf weitere Erklärungen

- GitHub-Page zur Einrichtung der Zwei-Faktor-Authentifizierung:
<https://help.github.com/en/github/authenticating-to-github/accessing-github-using-two-factor-authentication>
- GitHub-Page zur Einrichtung eines 2FA Personal-Access-Tokens:
<https://help.github.com/en/github/authenticating-to-github/creating-a-personal-access-token-for-the-command-line>
- GitHub-Page zur Umstellung der Sichtbarkeit einer Organisationsmitgliedschaft:
<https://help.github.com/en/github/setting-up-and-managing-your-github-user-account/publicizing-or-hiding-organization-membership>



(C) 2021 Karsten Reincke, Deutsche Telekom AG: This file is distributed under the terms of the [CC0-license](#)

Deutsche Telekom makes no warranties about the work, and disclaims liability for all uses of the work, to the fullest extent permitted by applicable law.