

Using GitHub accounts for contributing to and publishing code on behalf of Deutsche Telekom

Deutsche Telekom uses [GitHub](#) to publish its own open source projects (**Publishing**). As a clustering 'umbrella', it maintains a Telekom GitHub organization <https://github.com/telekom>. Deutsche Telekom welcomes its employees to further develop open source software used by the company and to return the results to the community's (GitHub) projects (**Contribution**)

In both cases, the employees must use a GitHub account to do so. It has already been shown several times that using two different accounts - one private and one professional - is very unwieldy. This document therefore explains how DT's employees create and / or set up a *private* GitHub account in a way that it is **also** usable in the professional context.

Some background information

Anyone who releases Deutsche Telekom's intellectual property as open source software can only do so on behalf of Deutsche Telekom, otherwise it would be illegal. Conversely, for the community, it is important that the legal release is documented, because they want to know that they have actually received the rights promised by licensing from Telekom.

For publishing of entire projects under our 'management' and for contributing to projects under 'third-party' management, we have created guidelines. They include the task of the contributors / developers to obtain the approval from their line managers and / or project managers. This consent includes the consent to use GitHub accounts.

Nevertheless, we need rules on how to indicate in the accounts that we are (currently) working as a Telekom employee.

Configuring the GitHub-Account

Deutsche Telekom allows its employees to use their 'private' GitHub accounts for Telekom work. But it doesn't force them to do so: Its employees can also create a specific Telekom GitHub employee account. But Deutsche Telekom wants the used accounts, be they 'private' or 'specific', to be set up in a certain way:

- **Activating a GitHub account** by
 - **Generating a new account** (option SIGN-UP at GitHub) *[the account name may be cryptic and conceal the conclusion about the person. Of course, Deutsche Telekom would like its employees to also stand by it externally.]*
 - **Selecting the own (private) account** under which you want to publish the Telekom code on GitHub.
- **Adding your own T-Email address** (you@telekom.de / you@t-systems.de / you@magenta.at) by logging in to your GitHub account and inserting it in the section [Settings/Emails](#). *[Everyone can decide for himself which email address shall become publicly visible.]*
- **Indicating your own employer** by editing your profile and inserting 'Deutsche Telekom AG' as company *[Feel free to append @teLekom. In this case, a reference to the GitHub organization is automatically generated. All other information (photo, description, etc.) should be set up by each*

employee in a way that appears useful to him and in accordance with the general rules of conduct of Telekom.]

Configuring your local git client

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.

Switching on the Two-Factor-Authentication

Normally, you equip your GitHub account with a password and log in with this pair *account/password*. This pair is also used if you contribute via `git` on the command line. If the account were hacked, the projects to which the account could contribute would be at risk. For Telekom, this would mean that a stranger would speak 'on its behalf' - which is unintended. Therefore, all GitHub accounts used by Telekom employees are to be gradually converted to and to be protected by the two-factor authentication:

- **Log in with your GitHub account**
- **Verify all email addresses** by taking the appropriate steps under Settings / Emails
- **Set up 2FA (Two Factor Authentication)** via
 - **installing the app** `andOTP` (Android) or `Google Authenticator` (Android or iOS) or `FreeOTP` (Android or iOS) on your smartphone *[Please do not install and use the app `authy` offered as default: it is not open source software and its use could cost Deutsche Telekom money under certain circumstances.]*
 - **enabling the 'two-factor authentication'** under Settings / Security
 - **activating the "Setup using an app" method**
 - **downloading and saving the recovery codes** (e.g. in KeePass). *[If they get lost, the account threatens to become permanently unusable.]*
 - **scanning** the appearing **QR code with the installed app** and
 - **entering the number** returned by the app **into the GitHub request dialog**.

If you now login yourself into your GitHub account, you will be asked for your password and for a number that must pick up from the installed app. After having logged in, you can use the GitHub tools in the known manner.

About some traps

- You have to enter the authentication number provided by the app quickly because it is renewed at short intervals - which is the point because it makes misuse extremely difficult.
- If you work on the shell with `git`, you will usually only be asked for your GitHub account and your normal password when accessing the remote repository. This is no longer possible after having activated the two-factor authentication. Instead using the real password, you have to have a new *personal access token* generated by GitHub, which replaces the previous password. For getting this personal access token, please
 - **Call the Developer-Settings entry** in the Settings dialog

- **Select the 'Personal Access Tokens' method and the option `Generate a personal access token`**
 - In the next dialog, **give the token a name, select (at least) the options 'repo' and 'delete-repo' and click the button `Generate token`**
 - **Save the appearing token** into a text file by copy & paste
 - **Enter the created token instead of the password** when asked by git for the account and password
- Anyone who uses the access token must be careful with C&P: blanks before and after tokens are obviously also evaluated in the shell.
 - Whoever generates the access token must carefully set the rights associated with the token
 - If you equip your 'private' account with 2FA you will also have to use two-factor authentication for your private accounts, not only for organizations that require 2FA use.

Making your memberships of GitHub organizations visible

Very often projects or companies work with several repositories. In this case, it makes sense to set up *GitHub organizations* for clustering the repositories. One such organization is <https://github.com/telekom>. The individual repositories have *contributors* or *collaborators*, the organization itself can also have its own *members*. Each member of the organization can decide for himself whether he wants to announce his membership to the outside world.

Deutsche Telekom employees should display their membership in the GitHub organization *Telekom* in their GitHub accounts. This illustrates the cooperation between Telekom and the open source community. Here are the steps you have to take for making your organizational membership visible to the outside world:

1. **Log in** to your account **at GitHub**.
2. At the top right **select the Account Drop Down Menu** (click on your picture or sample) and **select 'Your Profile'**.
3. **Click on the Telekom logo** in the list of organizations that appear on the left.
4. **Click on the tab `People`** in the arising organization repository.
5. **Search for your own account**.
6. **Change the entry from *private* to `public`** in the arising line.

Links

- GitHub-Page explaining the Two-Factor-Authentication:
<https://help.github.com/en/github/authenticating-to-github/accessing-github-using-two-factor-authentication>
 - GitHub-Page for getting a 2FA Personal-Access-Token:
<https://help.github.com/en/github/authenticating-to-github/creating-a-personal-access-token-for-the-command-line>
 - GitHub-Page explaining how to make your organisations visible:
<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.