

# Git Daten im Client hinterlegen

1. Man hat auch die Möglichkeit vom Laptop oder wo auch immer Code zu programmieren und dann hochzuladen. So braucht man das nicht auf der Github Seite zu programmieren. Dazu müssen Benutzername, Email und ein Public Key SSH KEY bei Github hochgeladen werden.
2. Im Terminal die Daten setzen.  
Wenn auf dem Computer nur ein Github Account besteht, dann können die Daten global gesetzt werden.  
Ansonsten diesen Schritt auslassen, da diese Daten im Verzeichnis des jeweiligen Git Repos hinterlegt werden, zu dessen Account dieser gehört.

```
git config --global user.email "eure_email_adresse"  
git config --global user.name "immohacker"
```

Siehe Abschnitt [Git mit mehreren Schlüsseln auf einem Client](#):

3. In Github den Schlüssel hochladen.  
Den Schlüssel auslesen mittels Terminal

```
cat ~/.ssh/id_rsa.pub
```

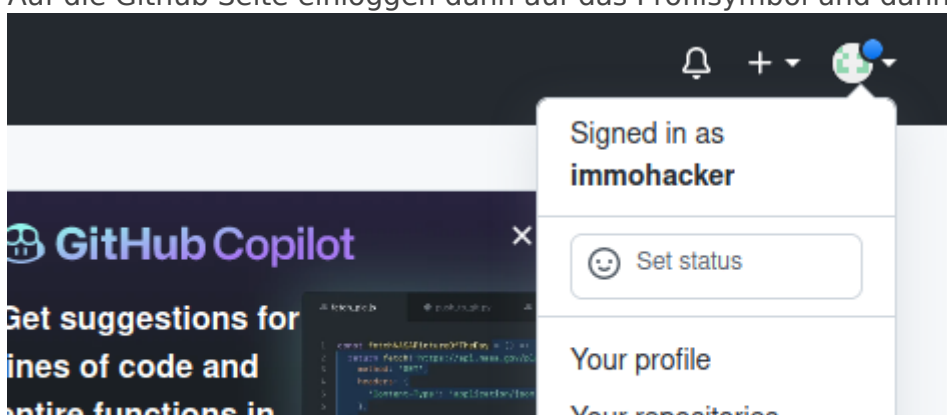
oder mit einem Editor öffnen und dann kopieren  
Ich mache das gern eben über Terminal.

Solltet Ihr noch keinen Schlüssel haben. Einfach über Terminal erstellen.  
Bei der Frage Passphrase einfach Enter drücken, nix eingeben

```
ssh-keygen -t rsa -b 4096
```

Standardmäßig wird das Homeverzeichnis mit dem Namen id\_rsa vorgeschlagen. Dieses würde ich auch beibehalten.

Auf die Github Seite einloggen, dann auf das Profilsymbol und dann Settings auswählen



Nun links im Menü auf SSH and GPG Keys, dann auf New SSH key klicken

The screenshot shows the GitHub profile page for 'immohacker'. On the left is a navigation menu with options like 'Public profile', 'Account', 'Appearance', 'Accessibility', 'Notifications', 'Access', 'Billing and plans', 'Emails', 'Password and authentication', 'SSH and GPG keys' (highlighted), 'Organizations', and 'Moderation'. The main content area has three sections: 'SSH keys' with a 'New SSH key' button and a message that there are no SSH keys; 'GPG keys' with a 'New GPG key' button and a message that there are no GPG keys; and 'Vigilant mode' with a checkbox for 'Flag unsigned commits as unverified' and a link to learn more.

Nun einen Namen vergeben, der den Rechner identifiziert, denn man könnte ja das Git Repo mit mehreren Geräten verwalten. Und den ssh key unten einfügen. Beginnt mit ssh-rsa und endet mit dem Rechnernamen.

The screenshot shows the 'Add new SSH key' form on GitHub. The 'Title' field contains 'Mein Laptop'. The 'Key' field contains a long string of characters: 'Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com''. There is an 'Add SSH key' button at the bottom right.

Nun kann auf das Konto per ssh Zugriff werden. So das der Git Client commits durchführen kann.  
Fertig.

The screenshot shows the GitHub profile page for 'immohacker' after adding the SSH key. The 'SSH keys' section now displays a list with one entry: 'Mein Laptop'. It shows the SHA256 hash (blurred), the date 'Added on 10 Jul 2022', and the status 'Never used — Read/write'. There is a 'Delete' button next to the entry. The navigation menu and other sections are the same as in the previous screenshot.

Version #4

Erstellt: 10 Juli 2022 07:01:50 von Admin

Zuletzt aktualisiert: 10 Juli 2022 11:38:40 von Admin