

Einen GitHub-Account erstellen – Schritt für Schritt



Du möchtest deinen Code online sichern, von überall darauf zugreifen oder später vielleicht sogar mit anderen zusammenarbeiten? Dann ist ein **GitHub-Account** der logische nächste Schritt nach der lokalen Git-Einrichtung. GitHub ist die weltweit größte Plattform für Code-Hosting und Versionsverwaltung – und die Registrierung ist kostenlos. Hier erfährst du, wie du deinen Account erstellst und welche Einstellungen du von Anfang an im Blick haben solltest.

Die Registrierung durchführen

1. GitHub-Website aufrufen

Öffne deinen Browser und gehe zu github.com. Auf der Startseite siehst du prominent einen Button wie „**Sign up**“ oder „**Registrieren**“.

2. E-Mail-Adresse eingeben

GitHub fragt dich zuerst nach deiner E-Mail-Adresse. Verwende eine Adresse, auf die du **langfristig Zugriff** hast – idealerweise dieselbe, die du auch in deiner lokalen Git-Konfiguration eingetragen hast. Das sorgt dafür, dass deine Commits später korrekt mit deinem GitHub-Profil verknüpft werden.

3. Passwort wählen

Wähle ein **starkes, einzigartiges Passwort**. GitHub enthält oft sensiblen Code und ist ein attraktives Ziel für Angreifer. Ein Passwort-Manager kann hier sehr hilfreich sein.

4. Benutzernamen festlegen

Der Benutzername ist ein wichtiger Punkt, über den du dir **vorab Gedanken machen** solltest (siehe nächster Abschnitt).

5. E-Mail-Verifizierung

GitHub schickt dir eine Bestätigungs-E-Mail. Klicke auf den Link darin, um deine Adresse zu verifizieren – erst dann ist dein Account vollständig aktiviert.

6. Optionale Personalisierung

Nach der Registrierung führt dich GitHub durch ein paar optionale Fragen (z. B. „Wofür nutzt du GitHub?“ oder „Wie groß ist dein Team?“). Diese dienen hauptsächlich dazu, dir passende Empfehlungen zu geben – du kannst sie überspringen, wenn du möchtest.

Den richtigen Benutzernamen wählen

Der Benutzername auf GitHub ist **öffentlich sichtbar** und Teil der URL zu deinen Repositories (z. B. `github.com/deinname/projekt`). Deshalb lohnt es sich, hier einen Moment nachzudenken:

- **Professionalität vs. Kreativität:** Wenn du GitHub auch für berufliche Zwecke nutzen möchtest (Bewerbungen, Portfolio, Open-Source-Beiträge), empfiehlt sich ein **seriöser, wiedererkennbarer Name**. Kombinationen aus Vor- und Nachname funktionieren gut, z. B. `maxmustermann` oder `m-mustermann`. Namen wie `xxxCoderKing_xXx` wirken weniger professionell.
- **Einzigkeit:** Der Name muss auf GitHub einzigartig sein. Beliebte Namen sind oft schon vergeben – sei also kreativ, aber bleib lesbar.
- **Änderbarkeit:** Du kannst deinen Benutzernamen später ändern, aber das hat Konsequenzen: Alle URLs zu deinen Repositories ändern sich, und Links von außen führen ins Leere. Wähle also möglichst von Anfang an einen Namen, mit dem du langfristig zufrieden bist.

“ **Tipp:** Wenn du planst, GitHub als Portfolio zu nutzen, überlege, ob der Name zu deinem „Personal Branding“ passt – also zu dem Eindruck, den du hinterlassen möchtest.

Wichtige Profileinstellungen nach der Registrierung

Sobald dein Account steht, solltest du einen Blick in die **Einstellungen** werfen. Du erreichst sie über dein Profilbild oben rechts → **Settings**. Hier die wichtigsten Bereiche:

Profil-Informationen

Unter **Public profile** kannst du grundlegende Informationen über dich hinterlegen:

- **Name:** Dein vollständiger oder angezeigter Name (unabhängig vom Benutzernamen).

- **Bio:** Eine kurze Beschreibung von dir – was du machst, welche Technologien dich interessieren.
- **Location:** Optional dein Standort.
- **Website/Social Links:** Hier kannst du auf dein Portfolio, LinkedIn oder andere Profile verlinken.

Diese Informationen sind **öffentlich** und helfen anderen, dich einzuordnen – besonders wenn du in Open-Source-Projekten aktiv wirst oder dein Profil als Portfolio nutzt.

E-Mail-Einstellungen

Unter **Emails** findest du wichtige Optionen:

- **Primary email:** Die Haupt-E-Mail-Adresse für Benachrichtigungen und Account-Wiederherstellung.
- **Keep my email addresses private:** Diese Option solltest du **aktivieren**, wenn du nicht möchtest, dass deine echte E-Mail-Adresse in Commits öffentlich sichtbar ist. GitHub stellt dir dann eine anonymisierte Adresse zur Verfügung (z. B. `123456+deinname@users.noreply.github.com`), die du in deiner lokalen Git-Konfiguration verwenden kannst.
- **Block command line pushes that expose my email:** Verhindert, dass du versehentlich Commits mit deiner echten E-Mail hochlädst.

“ ⚠ **Wichtig:** Wenn du deine lokale Git-Konfiguration mit einer privaten E-Mail-Adresse eingerichtet hast und diese nicht öffentlich sein soll, aktiviere unbedingt die Datenschutzoptionen und verwende die von GitHub bereitgestellte No-Reply-Adresse.

Zwei-Faktor-Authentifizierung einrichten

Unter **Password and authentication** findest du die Option zur **Two-factor authentication (2FA)**. Diese solltest du **dringend aktivieren**:

- 2FA schützt deinen Account, selbst wenn dein Passwort kompromittiert wird.
- GitHub unterstützt Authenticator-Apps (z. B. Google Authenticator, Authy, 1Password) und Hardware-Keys.
- Ohne 2FA kannst du bei manchen Aktionen eingeschränkt sein (z. B. bei der Nutzung von GitHub Actions in bestimmten Organisationen).

Die Einrichtung dauert nur wenige Minuten und erhöht die Sicherheit deines Accounts erheblich.

SSH-Keys und Personal Access Tokens

Für die Verbindung zwischen deinem lokalen Rechner und GitHub gibt es zwei gängige Methoden:

- **SSH-Keys:** Eine sichere Methode, bei der du einen Schlüssel auf deinem Computer generierst und den öffentlichen Teil bei GitHub hinterlegst. Das ermöglicht passwortloses Pushen und Pullen.
- **Personal Access Tokens (PAT):** Wenn du HTTPS statt SSH nutzt, brauchst du ein Token anstelle deines Passworts. Diese Tokens kannst du unter **Developer settings** → **Personal access tokens** erstellen.

Für den Anfang kannst du diesen Schritt überspringen – PhpStorm fragt dich beim ersten Push nach deinen Zugangsdaten und hilft dir bei der Einrichtung. Aber es ist gut zu wissen, wo du diese Optionen findest.

Repository-Sichtbarkeit verstehen

Wenn du später ein Repository auf GitHub erstellst, wirst du nach der **Sichtbarkeit** gefragt. Es gibt zwei Optionen:

Sichtbarkeit	Bedeutung
Public	Jeder kann das Repository sehen, auch ohne GitHub-Account. Der Code ist vollständig öffentlich. Andere können ihn ansehen, forken und bei Erlaubnis auch beitragen.
Private	Nur du und Personen, denen du explizit Zugriff gibst, können das Repository sehen. Ideal für persönliche Projekte, Kundenprojekte oder Code, den du nicht teilen möchtest.

Für deine **persönlichen Lern- und Übungsprojekte** ist „Private“ oft eine gute Wahl – du kannst sie jederzeit später öffentlich machen, wenn du sie zeigen möchtest. Für Open-Source-Projekte oder Portfolio-Stücke wählst du „Public“.

“  **Gut zu wissen:** Mit einem kostenlosen GitHub-Account kannst du **unbegrenzt viele private Repositories** erstellen – früher war das ein kostenpflichtiges Feature.

Die GitHub-Oberfläche im Überblick

Nach der Anmeldung landest du auf deinem **Dashboard**. Hier ein kurzer Überblick über die wichtigsten Bereiche:

flowchart TB

```
A["GitHub Dashboard"] --> B["Repositories\nDeine Projekte"]
```

```
A --> C["Explore\nOpen-Source entdecken"]
```

```
A --> D["Notifications\nBenachrichtigungen"]
```

```
A --> E["Settings\nAccount-Einstellungen"]
```

```
B --> F["Neues Repository erstellen"]
```

```
B --> G["Bestehende Repos verwalten"]
```

- **Repositories:** Hier siehst du alle deine Projekte und kannst neue anlegen.
- **Your profile:** Deine öffentliche Profilseite mit deinen Repositories, Beiträgen und Aktivitäten.
- **Explore:** Entdecke interessante Open-Source-Projekte.
- **Settings:** Alle Account- und Sicherheitseinstellungen.

Zusammenfassung: Deine Checkliste nach der Registrierung



Nachdem du deinen GitHub-Account erstellt hast, solltest du folgende Punkte abhaken:

1. **E-Mail verifiziert** – damit alle Funktionen freigeschaltet sind
2. **Benutzername überprüft** – ist er professionell und langfristig passend?
3. **Profil ausgefüllt** – zumindest Name und eine kurze Bio
4. **E-Mail-Privatsphäre aktiviert** – wenn du deine echte Adresse schützen möchtest
5. **Zwei-Faktor-Authentifizierung eingerichtet** – für die Sicherheit deines Accounts
6. **Lokale Git-Konfiguration angepasst** – falls du die No-Reply-Adresse von GitHub nutzen möchtest

Mit diesen Grundlagen bist du bestens gerüstet, um im nächsten Schritt dein erstes Repository auf GitHub zu erstellen und es mit deinem lokalen Projekt zu verbinden! ☐☐

Revision #1

Created 2026-06-17 01:53:58 UTC by art10m

Updated 2026-06-17 01:57:33 UTC by art10m