

Installation Windows

Auf Windows entfallen die macOS-spezifischen Direktiven (`UseKeychain` und `AddKeysToAgent`), da die Verwaltung systemweit vom Windows **OpenSSH-Agent-Dienst** übernommen wird.

Schritt-für-Schritt-Anleitung für Windows (PowerShell)

1. **OpenSSH-Dienst prüfen (einmalig als Admin):** Stellt sicher, dass Windows Keys dauerhaft im Speicher hält. Falls noch nicht erledigt, öffne die **PowerShell als Administrator**:

```
Set-Service -Name ssh-agent -StartupType Automatic  
Start-Service ssh-agent
```

2. **Neuen SSH-Key für GitHub generieren:** Separaten Key anlegen, damit Gitea unberührt bleibt. Führe in der normalen PowerShell aus:

```
ssh-keygen -t ed25519 -C "deine_email@example.com" -f $HOME\.ssh\id_ed
```

- Vergib bei Bedarf eine Passphrase.
3. **Key zum SSH-Agent hinzufügen:** Registriert den Schlüssel im Windows-Agenten. Gib deine Passphrase ein letztes Mal ein:

```
ssh-add $HOME\.ssh\id_ed25519_github
```

4. **SSH-Config anpassen (~/.ssh/config):** Trennung von GitHub und Gitea ohne macOS-spezifische Optionen. Öffne oder erstelle die Datei in VSCode:

```
code $HOME\.ssh\config
```

Füge die Konfiguration für beide Hosts ein:

```
# GitHub  
Host github.com  
HostName github.com  
User git  
IdentityFile ~/.ssh/id_ed25519_github  
IdentitiesOnly yes
```

Speichere die Datei mit **Strg + S** und schließe VSCode.

5. **Public Key in die Windows-Zwischenablage kopieren:**

```
Get-Content $HOME\.ssh\id_ed25519_github.pub | Set-Clipboard
```

Füge den Schlüssel auf github.com unter **Settings** → **SSH and GPG keys** → **New SSH Key** ein.

6. **Verbindung testen & Repository klonen:** Prüfe die Verbindung:

```
ssh -T git@github.com
```

Klone das Repository z.B.:

```
git clone git@github.com:[REPO].git
```

Wichtiger Hinweis für Git Bash auf Windows

Damit auch die **Git Bash** dieselbe `~/.ssh/config` und den laufenden Windows-Dienst verwendet, stelle sicher, dass Git den Windows-nativen SSH-Client anspricht:

```
git config --global core.sshCommand "ssh.exe"
```