

Associating an email with your GPG key

Your GPG key must be associated with a GitHub Enterprise Server verified email that matches your committer identity.

Mac Windows Linux

If you're using a GPG key that matches your committer identity and your verified email address associated with your account on your GitHub Enterprise Server instance, then you can begin signing commits and signing tags.

- 1 Open TerminalTerminalGit Bash.
- 2 Use the `gpg --list-secret-keys --keyid-format=long` command to list the long form of the GPG keys for which you have both a public and private key. A private key is required for signing commits or tags.

Shell



```
gpg --list-secret-keys --keyid-format=long
```

Note: Some GPG installations on Linux may require you to use `gpg2 --list-keys --keyid-format LONG` to view a list of your existing keys instead. In this case you will also need to configure Git to use `gpg2` by running `git config --global gpg.program gpg2`.

- 3 From the list of GPG keys, copy the long form of the GPG key ID you'd like to use. In this example, the GPG key ID is `3AA5C34371567BD2`:

Shell



```
$ gpg --list-secret-keys --keyid-format=long
/Users/hubot/.gnupg/secring.gpg
-----
sec  4096R/3AA5C34371567BD2 2016-03-10 [expires: 2017-03-10]
uid                               Hubot <hubot@example.com>
ssb  4096R/4BB6D45482678BE3 2016-03-10
```

- 4 Enter `gpg --edit-key GPG key ID`, substituting in the GPG key ID you'd like to use. In the following example, the GPG key ID is `3AA5C34371567BD2`:

```
gpg --edit-key 3AA5C34371567BD2
```

- 5 Enter `gpg> adduid` to add the user ID details.

```
gpg> adduid
```

- 6 Follow the prompts to supply your real name, email address, and any comments. You can modify your entries by choosing **N**, **C**, or **E**.

```
Real Name: OCTOCAT
Email address: "octocat@github.com"
Comment: GITHUB-KEY
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?
```

- 7 Enter **0** to confirm your selections.

- 8 Enter your key's passphrase.

- 9 Enter `gpg> save` to save the changes

```
gpg> save
```

- 10 Enter `gpg --armor --export GPG key ID`, substituting in the GPG key ID you'd like to use. In the following example, the GPG key ID is `3AA5C34371567BD2`:

```
$ gpg --armor --export 3AA5C34371567BD2
# Prints the GPG key, in ASCII armor format
```

- 11 Upload the GPG key by [adding it to your GitHub account](#).

Further reading

- ["Checking for existing GPG keys"](#)
- ["Generating a new GPG key"](#)
- ["Using a verified email address in your GPG key"](#)
- ["Adding a GPG key to your GitHub account"](#)
- ["Signing commits"](#)
- ["Signing tags"](#)

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)