



INTRODUCTION TO GIT & GITHUB

by Bobby Ilier



Table of Contents

Sobre o livro	6
Sobre o autor	7
Patrocinadores	8
Ferramenta de geração do PDF do Ebook	9
Ferramenta de geração do ePub do Ebook	10
Capa do Livro	11
Licença	12
 Introdução ao Git	 13
 Controle de Versão	 14
 Instalando o Git	 16
 Comandos Básicos do Shell	 18
 Configuração do Git	 22
 Introdução ao GitHub	 26
Estrelas do GitHub	29
 Inicializando um projeto Git	 30
 Git Status	 32
 Git Add	 34

Git Commit	36
Assinando Commits	38
Git Diff	43
Git Log	45
Gitignore	47
Chaves SSH	56
Git Push	59
Criando e vinculando um repositório remoto	60
Enviando commits	61
Verificando o repositório remoto	62
Git Pull	63
Branches do Git	66
Git Merge	72
Revertendo alterações	78
Resetando alterações (⚠ Resetar é perigoso ⚠)	79
Git Clone	83
Fork no Git	85

```
git log
```

Aqui está um exemplo do processo:

```

root@do-dev:~/demo# git init .
Initialized empty Git repository in /root/demo/.git/
root@do-dev:~/demo# touch demo.txt
root@do-dev:~/demo# git add .
root@do-dev:~/demo# git commit -m "First commit"
[master (root-commit) 45f651d] First commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 demo.txt
root@do-dev:~/demo# echo "Wrong changes..." > demo.txt
root@do-dev:~/demo# git add .
root@do-dev:~/demo# git commit -m "Wrong commit..."
[master 9688e23] Wrong commit...
 1 file changed, 1 insertion(+)
root@do-dev:~/demo# git log
commit 9688e23761a6ccbbfaa4362a391b50a63d4ba39a (HEAD -> master)
Author: Bobby Iliev <bobby@bobbyiliev.com>
Date:   Wed Jan 8 08:49:49 2020 +0000

    Wrong commit...

commit 45f651dd4e83205fe1a72ac16bf0d7a3ecfba904
Author: Bobby Iliev <bobby@bobbyiliev.com>
Date:   Wed Jan 8 08:49:27 2020 +0000

    First commit
root@do-dev:~/demo# git reset --soft HEAD~1
root@do-dev:~/demo# git log
commit 45f651dd4e83205fe1a72ac16bf0d7a3ecfba904 (HEAD -> master)
Author: Bobby Iliev <bobby@bobbyiliev.com>
Date:   Wed Jan 8 08:49:27 2020 +0000

    First commit
root@do-dev:~/demo# echo "Fixed changes..." > demo.txt
root@do-dev:~/demo# git add .
root@do-dev:~/demo# git commit -m "Fixed commit..."
[master 081f0fb] Fixed commit...
 1 file changed, 1 insertion(+)
root@do-dev:~/demo# git log
commit 081f0fbf3d32ad7e7946e1ec4cc5b432fc699fef (HEAD -> master)
Author: Bobby Iliev <bobby@bobbyiliev.com>
Date:   Wed Jan 8 08:50:28 2020 +0000

    Fixed commit...

commit 45f651dd4e83205fe1a72ac16bf0d7a3ecfba904
Author: Bobby Iliev <bobby@bobbyiliev.com>
Date:   Wed Jan 8 08:49:27 2020 +0000

    First commit
root@do-dev:~/demo#

```

Nota: Você pode resetar suas alterações voltando mais de um commit usando a seguinte sintaxe:

```
git reset --soft HEAD~n
```

onde **n** é o número de commits que você quer voltar.

Outra abordagem seria usar **git revert COMMIT_ID**.

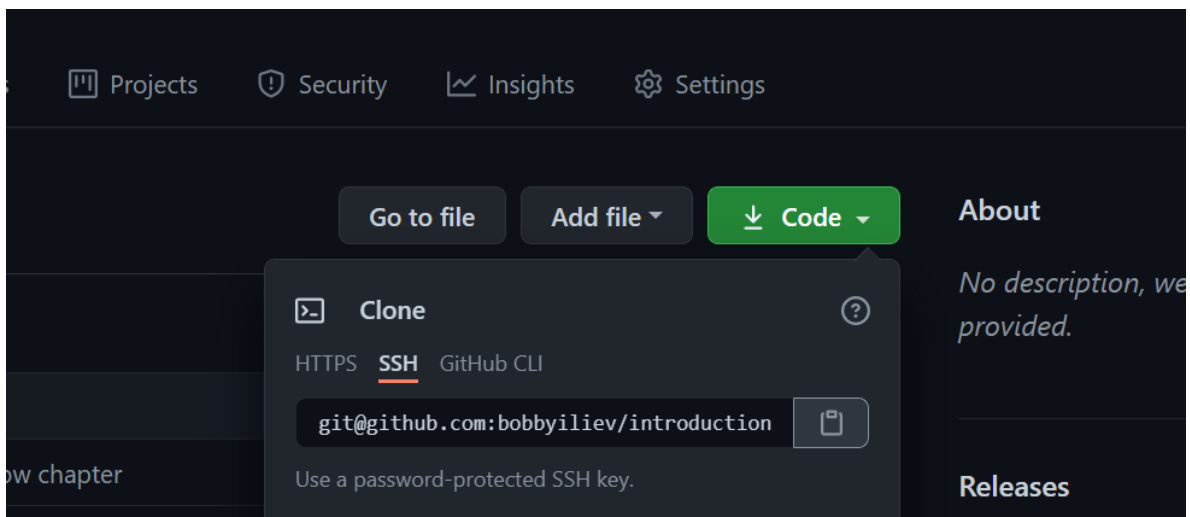
Aqui está um vídeo rápido demonstrando como fazer isso:

[Revertendo alterações](#)

Git Clone

Na maioria das vezes, em vez de começar um projeto do zero, você vai entrar em uma empresa e começar a trabalhar em um projeto existente, ou contribuir para um projeto open source já estabelecido. Nesse caso, para obter o repositório do GitHub para sua máquina local, você precisa usar o comando `git clone`.

A maneira mais simples de clonar seu repositório do GitHub é primeiro visitar o repositório no navegador, clicar no botão verde **Code** e escolher o método que deseja usar para clonar o repositório:



No meu caso, prefiro o método SSH, pois já tenho minhas chaves SSH configuradas conforme o capítulo 14.

Como estou clonando este repositório [aqui](#), a URL ficará assim:

```
git@github.com:bobbyiliev/introduction-to-bash-scripting.git
```

Depois de copiar isso, volte ao terminal, vá até o diretório onde deseja clonar o repositório e execute o seguinte comando:

```
git clone git@github.com:bobbyiliev/introduction-to-bash-scripting.git
```

A saída será parecida com isto:

```
Cloning into 'introduction-to-bash-scripting'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 215 (delta 7), reused 14 (delta 4), pack-reused
194
Receiving objects: 100% (215/215), 3.08 MiB | 5.38 MiB/s,
done.
Resolving deltas: 100% (114/114), done.
```

Basicamente, o comando `git clone` faz o download do repositório do GitHub para sua pasta local.

Agora você pode começar a fazer alterações no projeto criando um novo branch, escrevendo código e finalmente commitando e enviando suas alterações!

Uma coisa importante a lembrar é que, caso você não seja o mantenedor do repositório e não tenha permissão para enviar alterações diretamente, será necessário primeiro fazer um fork do repositório original e então clonar o repositório forkado da sua conta. No próximo capítulo, veremos o processo completo de como fazer um fork de um repositório!

Fork no Git

Ao contribuir para um projeto open source, você não poderá fazer alterações diretamente no projeto. Somente os mantenedores do repositório têm esse privilégio.

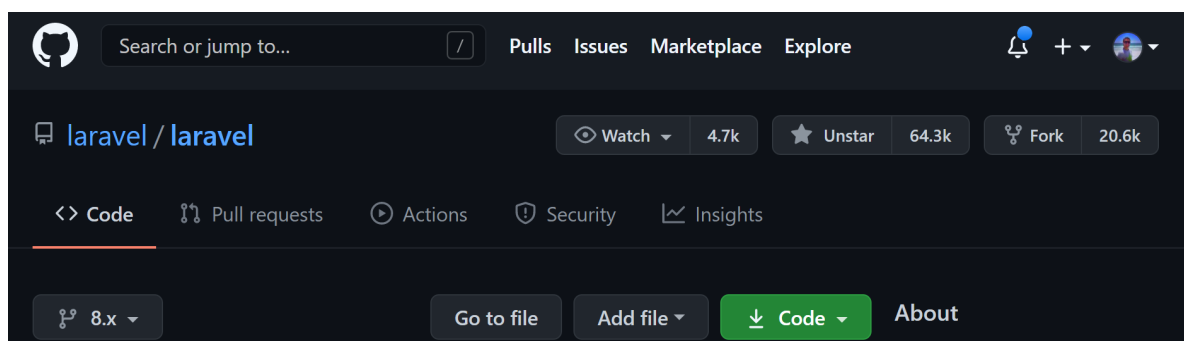
O que você precisa fazer é fazer um fork do repositório específico, realizar as alterações no projeto forkado e então enviar um pull request para o projeto original. Você aprenderá mais sobre pull requests nos próximos capítulos.

Se você clonar um repositório ao qual não tem acesso e tentar enviar alterações diretamente para esse repositório, receberá o seguinte erro:

```
ERROR: Permission to laravel/laravel.git denied to bobbyiliev.  
Fatal: Could not read from remote repository.  
  
Please make sure you have the correct access rights  
and the repository exists.
```

É aí que entra o Fork!

Para fazer um fork de um repositório, acesse o repositório pelo navegador e clique no botão Fork no canto superior direito:



Depois, escolha a conta para a qual deseja fazer o fork do repositório:

Instalação do GitHub CLI

Como estou usando Ubuntu, para instalar o **gh** você precisa rodar os seguintes comandos:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-key  
C99B11DEB97541F0  
sudo apt-add-repository https://cli.github.com/packages  
sudo apt update  
sudo apt install gh
```

Se você estiver no Mac, pode instalar o **gh** usando o Homebrew:

```
brew install gh
```

Para outros sistemas operacionais, recomendo seguir os passos da documentação oficial [aqui](#).

Depois de instalar o **gh**, você pode verificar se está funcionando com o comando:

```
gh --version
```

Isso mostrará a versão do **gh**:

```
gh version 1.0.0 (2020-09-16)  
https://github.com/cli/cli/releases/tag/v1.0.0
```

No meu caso, estou usando a versão mais recente **gh** v1.0.0, lançada há poucos dias.

Autenticação

Depois de instalar o **gh**, você precisa fazer login na sua conta do GitHub.

Para isso, execute o comando:

```
gh auth login
```

Você verá a seguinte saída:

```
? What account do you want to log into? [Use arrows to move,  
type to filter]  
> GitHub.com  
  GitHub Enterprise Server
```

Você pode escolher entre GitHub.com ou GitHub Enterprise. Pressione enter e siga o processo de autenticação.

Outro comando útil é o **gh help**. Ele mostra uma lista dos comandos disponíveis do **gh** que você pode usar:

USAGE

gh <command> <subcommand> [flags]

CORE COMMANDS

gist: Criar gists
issue: Gerenciar issues
pr: Gerenciar pull requests
release: Gerenciar releases do GitHub
repo: Criar, clonar, fazer fork e visualizar repositórios

ADDITIONAL COMMANDS

alias: Criar atalhos de comando
api: Fazer uma requisição autenticada à API do GitHub
auth: Login, logout e atualizar autenticação
completion: Gerar scripts de autocompletar para o shell
config: Gerenciar configuração do gh
help: Ajuda sobre qualquer comando

FLAGS

--help Mostrar ajuda do comando
--version Mostrar versão do gh

EXEMPLOS

```
$ gh issue create  
$ gh repo clone cli/cli  
$ gh pr checkout 321
```

VARIÁVEIS DE AMBIENTE

Veja '`gh help environment`' para a lista de variáveis de ambiente suportadas.

SAIBA MAIS

Use '`gh <command> <subcommand> --help`' para mais informações sobre um comando.

Leia o manual em <https://cli.github.com/manual>

FEEDBACK

Abra uma issue usando '`gh issue create -R cli/cli`'

Agora vamos clonar um projeto existente para testar. Como exemplo, podemos usar o repositório [LaraSail](#). Em vez de usar o comando padrão `git clone`, vamos usar o `gh`:

```
gh repo clone thedevdojo/larasail
```

Você verá a seguinte saída:

```
Cloning into 'larasail'...
```

Depois disso, entre na pasta:

```
cd larasail
```

Agora estamos prontos para usar alguns dos comandos mais úteis do **gh**!

This is a sample from "Introduction to Git and GitHub" by Bobby Iliev.

For more information, [Click here](#).