CODING Code Repositories

Getting Started

Product Documentation

Copyright Notice

Trademark Notice

Tencent Cloud

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Getting Started

Last updated：2023-12-25 17:08:18

Operational Scenarios

CODING Code Repository (CODING-CR) is based on the Git open-source version control system. With Git, your local computer and CODING each have a complete code repository, so both can engage in distributed version management. To manage code on your local computer, you must first download, install, and set up Git. If you don't need to manage code on your local computer, CODING allows you to perform most code repository operations in your browser. This document explains how to get started with CODING-CR.

Prerequisites

Before using CODING_CR, you must activate the CODING DevOps service for your Tencent Cloud account. Follow the pop-up prompts to go to **Personal Settings** and enter your email, password, and mobile number. You will use this information when performing subsequent operations, such as code cloning or push.

Local Environment Initialization

Install Git

Windows

1. Download Git from the Git website and follow the prompts to install it. We recommend using the default options.
2. After installation, right-click to open Git Bash and get started with Git.

Linux

On a Linux system, you can use the system's package management tool to directly install the pre-compiled Git binary installation package.
On a Fedora/Centos system, use yum to install Git: `yum install git-core` .
On a Ubuntu/Debian system, use apt-get to install Git: `apt-get install git` .

macOS

1. Run the following command to install the package management tool Homebrew.

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/
```

Note:

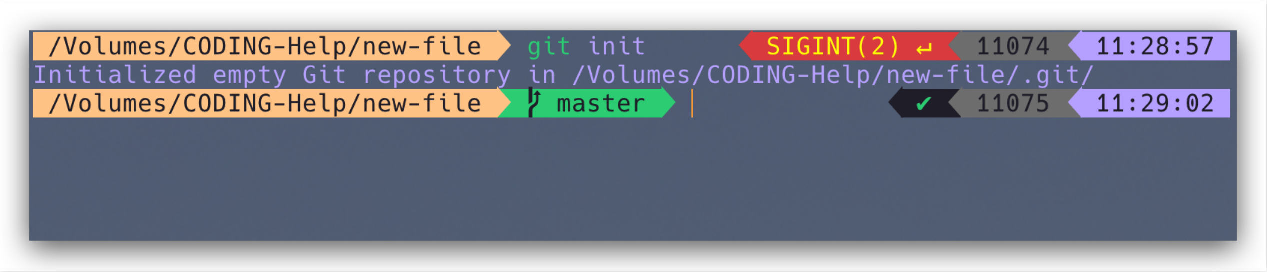
If an error occurs, see Open Source Repositories and download from a domestic source.

2. After Homebrew is installed, enter `brew install git` to install Git.

3. After Git is installed, run `git --version` to check your current Git version.

Local environment initialization

After Git is installed, create a folder, open the folder, and enter `git init` to initialize the local environment.



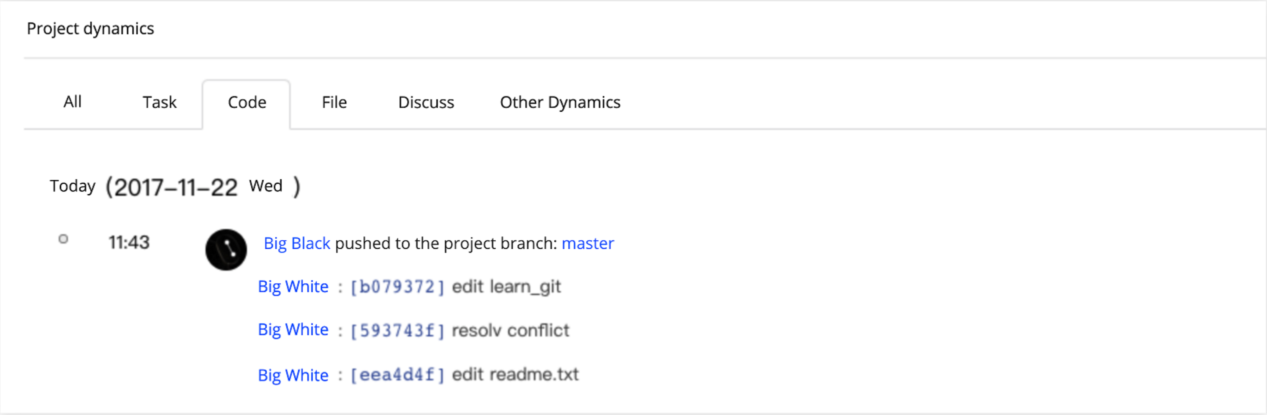
```
/Volumes/CODING-Help/new-file  git init                    SIGINT(2) ↵  11074  11:28:57
Initialized empty Git repository in /Volumes/CODING-Help/new-file/.git/
/Volumes/CODING-Help/new-file   master |                          ✓  11075  11:29:02
```

Set user information

After Git is installed, you should immediately set a committer name and email address, which will be used to create a record for each commit. Use the following command to set user information.

```
$ git config --global user.name "Your name"
$ git config --global user.email "Your email"
```

For example, if your CODING account is named **Dahei** and your Git user information is: **name – Dabai** and **Email –** dabai@coding.net, when your code is pushed to your CODING repository, the status is displayed as follows:



Today (2017–11–22 Wed)

11:43 Big Black pushed to the project branch: master

Big White : [b079372] edit learn_git

Big White : [593743f] resolv conflict

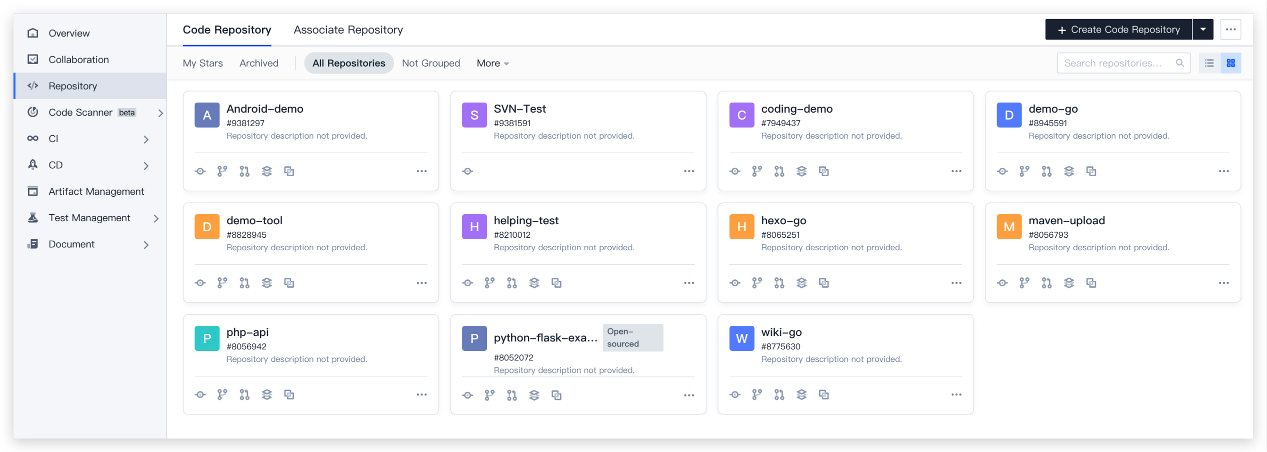
Big White : [eea4d4f] edit readme.txt

Note:

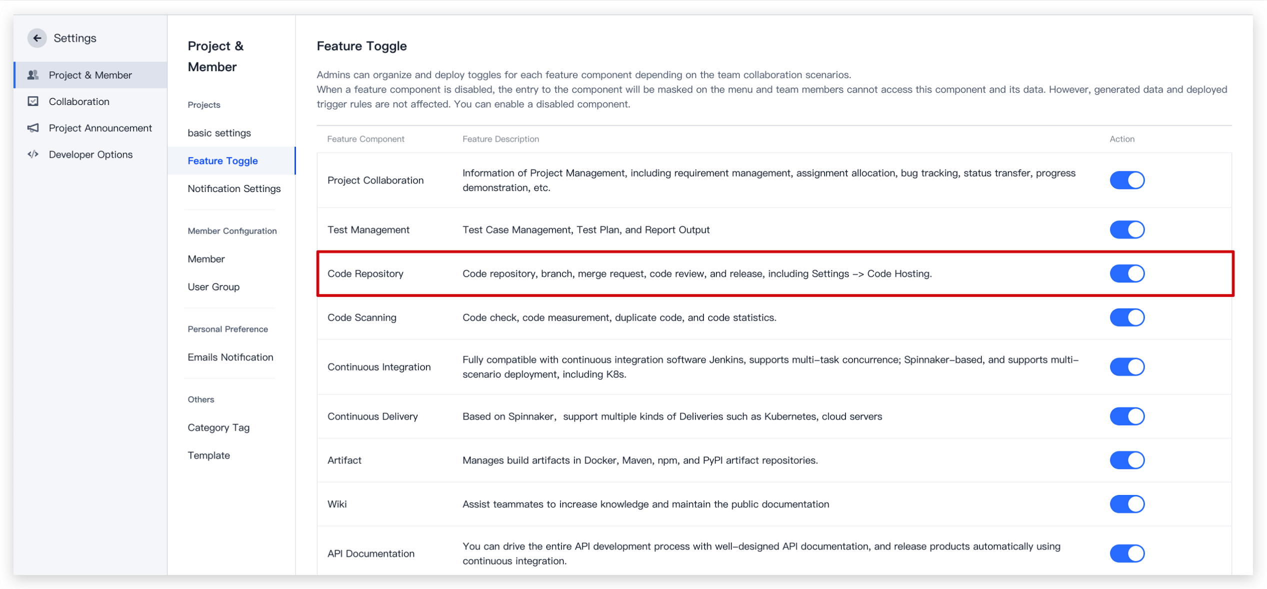
You can configure custom user information in Git. We recommend you directly use the username and email address of your CODING account for efficient collaboration.

Create a CODING Code Repository

1. Open any project and click **Code Repositories** on the left navigation bar to open the Code Repository Management page, where you can create or open a code repository.


Overview
Collaboration
Repository
Code Scanner beta
CI
CD
Artifact Management
Test Management
Document
Code Repository
Associate Repository
+ Create Code Repository
My Stars
Archived
All Repositories
Not Grouped
More
Search repositories...
A Android–demo #9381297 Repository description not provided.
S SVN–Test #9381591 Repository description not provided.
C coding–demo #7949437 Repository description not provided.
D demo–go #8945591 Repository description not provided.
D demo–tool #8828945 Repository description not provided.
H helping–test #8210012 Repository description not provided.
H hexo–go #8065251 Repository description not provided.
M maven–upload #8056793 Repository description not provided.
P php–api #8056942 Repository description not provided.
P python–flask–exa... Open–sourced #8052072 Repository description not provided.
W wiki–go #8775630 Repository description not provided.

2. If the code repository entry is not displayed, a project admin must open the project, click **Project Settings** in the lower-left corner, and go to **Project and Members** > **Functions** to enable the code repository function.



Feature Toggle

Admins can organize and deploy toggles for each feature component depending on the team collaboration scenarios.
When a feature component is disabled, the entry to the component will be masked on the menu and team members cannot access this component and its data. However, generated data and deployed trigger rules are not affected. You can enable a disabled component.

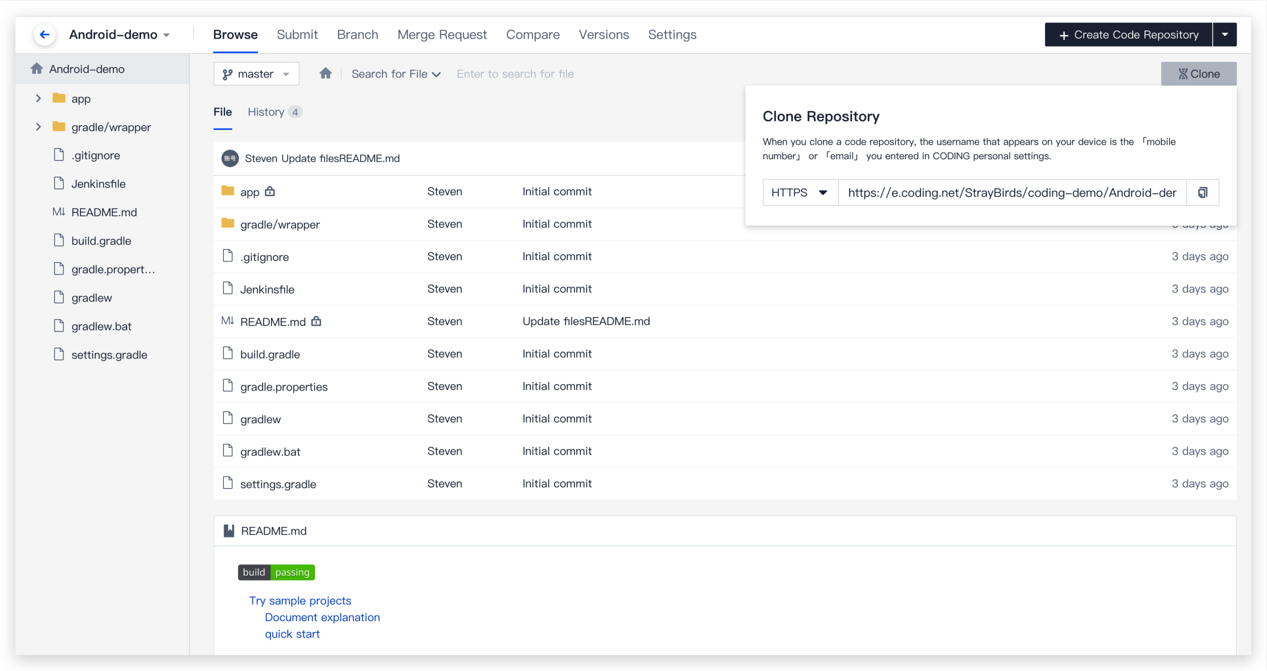
Feature Component	Feature Description	Action
Project Collaboration	Information of Project Management, including requirement management, assignment allocation, bug tracking, status transfer, progress demonstration, etc.	●
Test Management	Test Case Management, Test Plan, and Report Output	●
Code Repository	Code repository, branch, merge request, code review, and release, including Settings -> Code Hosting.	●
Code Scanning	Code check, code measurement, duplicate code, and code statistics.	●
Continuous Integration	Fully compatible with continuous integration software Jenkins, supports multi-task concurrence; Spinnaker-based, and supports multi-scenario deployment, including K8s.	●
Continuous Delivery	Based on Spinnaker, support multiple kinds of Deliveries such as Kubernetes, cloud servers	●
Artifact	Manages build artifacts in Docker, Maven, npm, and PyPI artifact repositories.	●
Wiki	Assist teammates to increase knowledge and maintain the public documentation	●
API Documentation	You can drive the entire API development process with well-designed API documentation, and release products automatically using continuous integration.	●

Push and Pull Code

This section will demonstrate how to pull code from and push code to remote repositories to facilitate cloud-based coding.

Pull code from remote repository

After initializing a local code repository, you can open the terminal in a folder and run the `git clone + repository address` command to pull code.



Clone Repository

When you clone a code repository, the username that appears on your device is the 「mobile number」 or 「email」 you entered in CODING personal settings.

HTTPS https://e.coding.net/StrayBirds/coding-demo/Android-der

Steven Update filesREADME.md

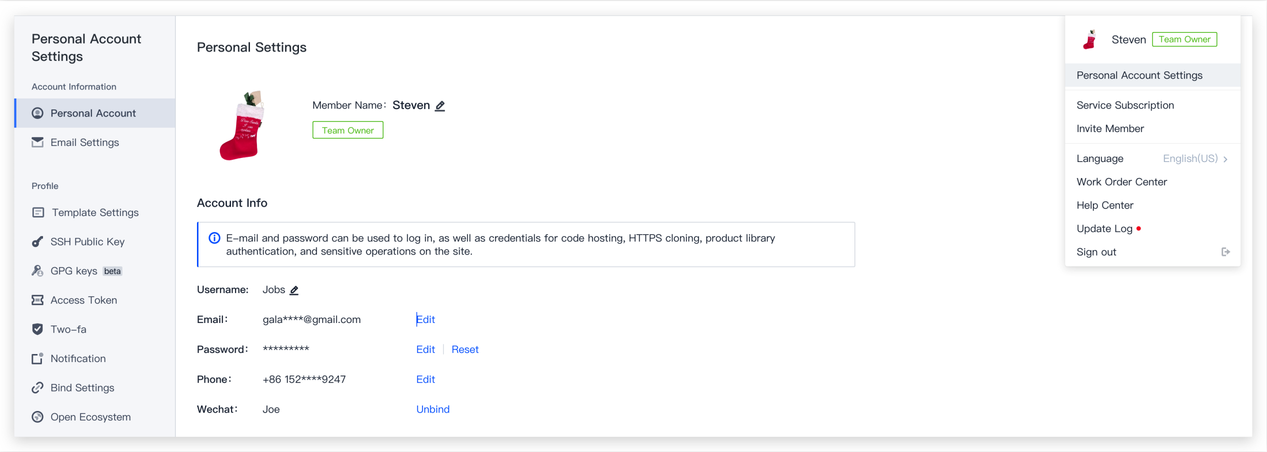
app	Steven	Initial commit	
gradle/wrapper	Steven	Initial commit	
.gitignore	Steven	Initial commit	3 days ago
Jenkinsfile	Steven	Initial commit	3 days ago
README.md	Steven	Update filesREADME.md	3 days ago
build.gradle	Steven	Initial commit	3 days ago
gradle.properties	Steven	Initial commit	3 days ago
gradlew	Steven	Initial commit	3 days ago
gradlew.bat	Steven	Initial commit	3 days ago
settings.gradle	Steven	Initial commit	3 days ago

README.md

build passing

Try sample projects
Document explanation
quick start

When pulling for the first time, you must enter your credentials. Enter the email and password you used to register a CODING account. You can also open the dropdown menu in the upper-right corner and go to **Personal Account Settings** to modify your credential information.



Personal Settings

Member Name: Steven

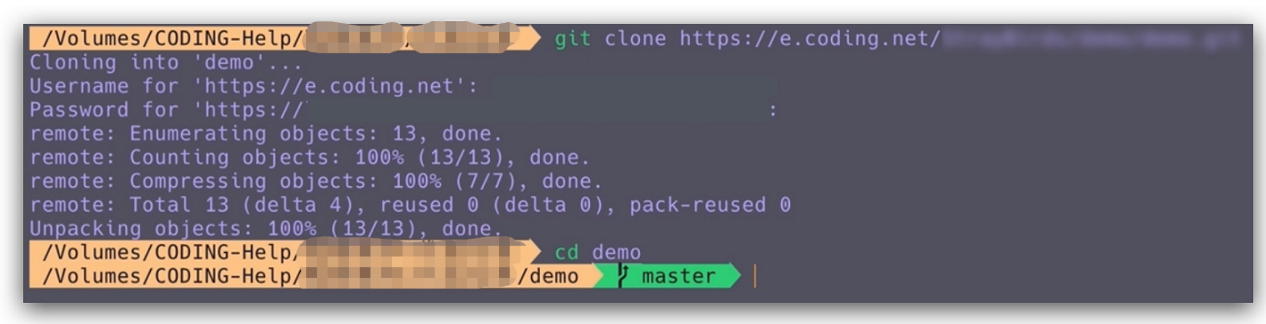
Team Owner

Account Info

E-mail and password can be used to log in, as well as credentials for code hosting, HTTPS cloning, product library authentication, and sensitive operations on the site.

Username:	Jobs	
Email:	gala****@gmail.com	Edit
Password:	*********	Edit | Reset
Phone:	+86 152****9247	Edit
Wechat:	Joe	Unbind

When the operation is successful, you can modify the code in your local code repository.



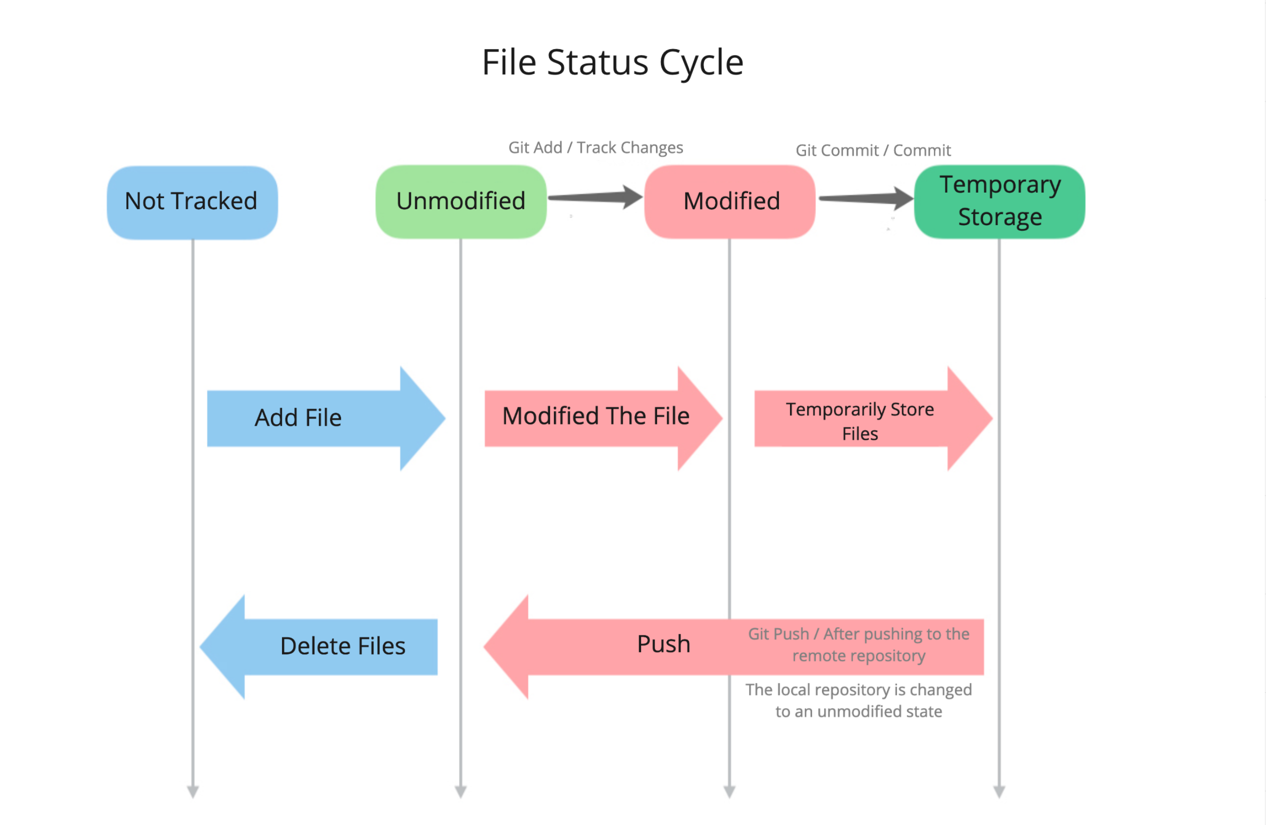
```
/Volumes/CODING-Help/[redacted] git clone https://e.coding.net/[redacted]
Cloning into 'demo'...
Username for 'https://e.coding.net': [redacted]
Password for 'https://[redacted]:
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 13 (delta 4), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (13/13), done.
/Volumes/CODING-Help/[redacted] cd demo
/Volumes/CODING-Help/[redacted]/demo  master
```

Edit files

In a folder, create the `readme.txt` and `learn-git.txt` files. In one of the files, write `I'm learning git.` (or customize as needed) and save it.

Flowchart

Before committing code, you can refer to this flowchart to understand the status lifecycle Git uses when tracking files.



Track files (git add)

After creating or editing a file, run the `git add` command to add the file to the staging area (Index Stage) of the local Git repository.

Command to **track a specific file**:

```
git add readme.txt
```

Command to **add multiple files**:

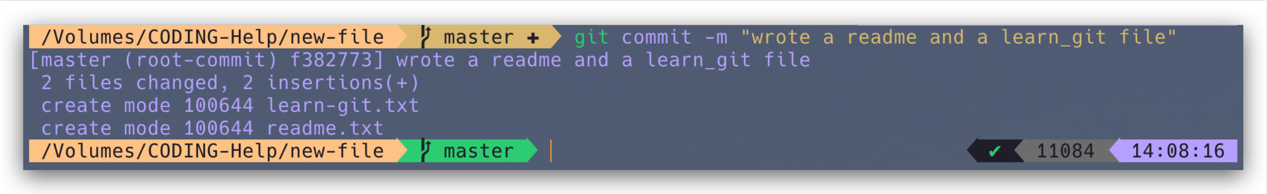
```
git add readme.txt learn_git.txt
```

To track **all files** at once, you can directly enter `git add` in the terminal.

Commit files (git commit)

After files to commit are added to the staging area, run `git commit` to commit the files to the local repository. This command commits all files in the staging area:

```
git commit -m "wrote a readme and a learn_git file"
```



```
/Volumes/CODING-Help/new-file  master + git commit -m "wrote a readme and a learn_git file"
[master (root-commit) f382773] wrote a readme and a learn_git file
 2 files changed, 2 insertions(+)
 create mode 100644 learn-git.txt
 create mode 100644 readme.txt
/Volumes/CODING-Help/new-file  master                                    ✓ 11084  14:08:16
```

The content enclosed in quotes following `-m` is your commit description. The following lines are the result returned. Make sure that you add a change description to each commit to provide a clear description of what changes were committed.

In addition to the `git commit` command, you can also use the standard plugin to standardize commit messages in the repository for easy backtracking.

```
// Step 1: Install yarn

brew install yarn
```

```
// Step 2: Install plugin

yarn add -D standard-version

// Step 3: Commit code with the plugin

git cz
```

Associate issues automatically

When you commit code, `# Issue ID` is included in the commit message. You can also directly associate issues in the project.

If the Development Convention function is enabled for this repository and a commit convention (such as check associated issues) is specified in the Branch Convention, make sure issues of the corresponding type are associated in the commit message. Otherwise, this branch will violate the convention.

For example, for an issue (requirement) with the ID 630, add `#630` in the commit message to automatically associate the requirement. Add `project-1#630` to associate issue 630 in `project-1`.

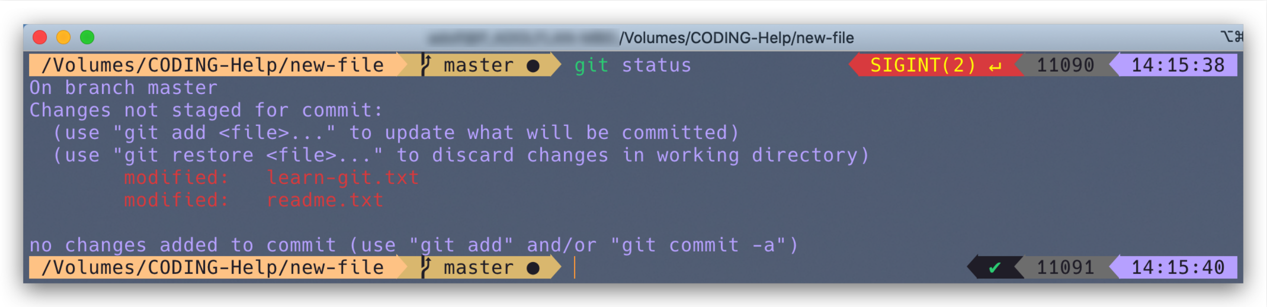
View file status (git status)

If you are not sure that Git is precisely tracking file changes and want to confirm the file status, use the `git status` command to see the file status.

If no files are tracked in the current repository, the result returned is as follows:

```
$ git status
On branch master
Your branch is up-to-date with 'origin/master'
nothing to commit, working directory clean
```

If a file has changed but the changes were not tracked, the result returned is as follows:

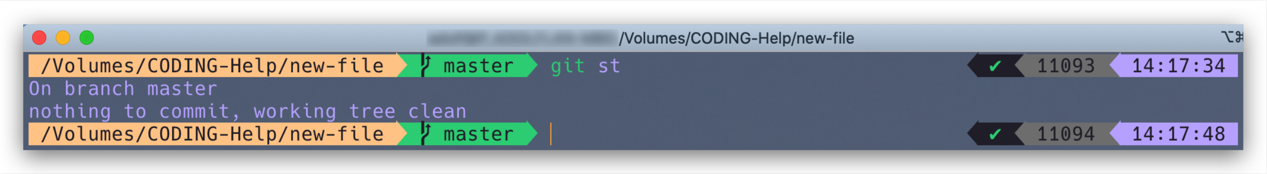


```
/Volumes/CODING-Help/new-file  master ●  git status                    SIGINT(2) ↵  11090  14:15:38
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   learn-git.txt
        modified:   readme.txt

no changes added to commit (use "git add" and/or "git commit -a")
/Volumes/CODING-Help/new-file  master ●                                      ✔  11091  14:15:40
```

Run `git add` to track the file. After successful tracking, the font color will change from red to green.

When a file has been tracked and already committed to the repository, the result returned is as follows:



```
/Volumes/CODING-Help/new-file  master  git st                                 ✔  11093  14:17:34
On branch master
nothing to commit, working tree clean
/Volumes/CODING-Help/new-file  master                                         ✔  11094  14:17:48
```

Push files to remote repository (git push)

On the terminal, run the command:

```
git push
```

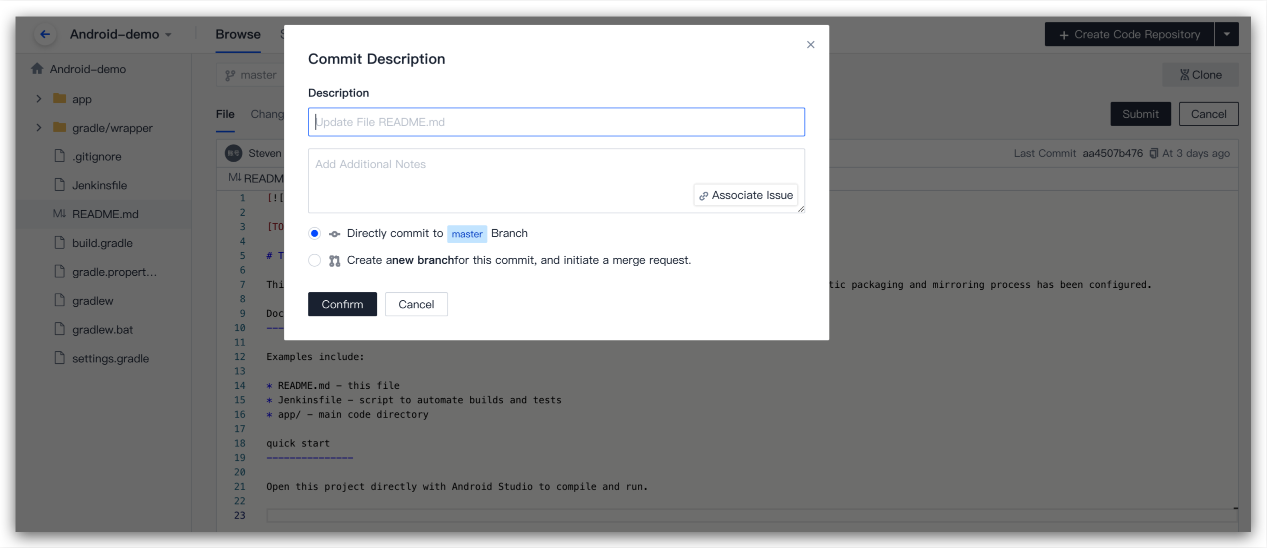
To automatically create a merge request and associate issues upon commit, use the following command:

```
git push origin local-branch:mr/target-branch/local-branch
```

The `git push` is the push command. It pushes the local branch to the remote repository, which is equivalent to creating a remote backup. Go to the CODING code repository to view pushed files. If multiple people collaborate in maintaining the remote repository, you need to run the `git pull` command to synchronize your local repository with the remote repository to get any code committed by other users.

View and Edit Remote Repositories

After a file is pushed to the CODING code repository, you can edit, save, and commit code on the webpage. Let's take the `README.md` file as an example. After editing and committing the changes, you can add a simple description of the modified content. If you do not add a description, the default commit description is "File xxx updated".



Commit Description

Description

Update File README.md

Add Additional Notes

Associate Issue

● Directly commit to master Branch

○ Create anew branchfor this commit, and initiate a merge request.

Confirm Cancel

```
[![
[TO
# T
Thi                                                    tic packaging and mirroring process has been configured.
Doc
---

Examples include:

* README.md - this file
* Jenkinsfile - script to automate builds and tests
* app/ - main code directory

quick start
---------------

Open this project directly with Android Studio to compile and run.

```