

# GIT

## 基本使用及管理策略

# 目录

1. Git、GitHub 和 GitLab
2. 常用图形化工具
3. 常用 Git 命令
4. 常用概念
5. Git 管理策略
6. 周报系统
7. 代码风格

# 1. Git、GitHub 和 GitLab

# 1.1 Git

## 1.1.1 Git 是什么？

Git 是一个开源的分布式版本控制系统，可以有效、高速地处理从“很小”到“非常大”的项目版本管理。

- Git 是一个工具，因此没必要把时间浪费在那些“高级但几乎永远不会用到”的命令上；
- 真的到了非用不可时，再自己 Google 或请教专家。

# 1.1 Git

## 1.1.2 Git 资源

- [Pro Git](#)
- [廖雪峰的 Git 教程](#)
- 微信号：Git小助手

# 1.2 GitHub

GitHub 是一个提供给用户空间创建 Git 仓储的网站，可以保存用户的一些数据文档或者代码等：

- 全球开发者公用，所谓的“码农在线交友社区”
- 乐鑫 GitHub: <https://github.com/espressif>
- Fork、Pull Request 和 Issues: [esp-idf](#)

# 1.3 GitLab

GitLab 是一个用于仓库管理系统的开源项目，使用 Git 作为代码管理工具，并在此基础上搭建起来的 Web 服务。

- 内部 GitLab：主要用于内部项目的开发、管理，使用人员为 R&D、AE 及测试人员。
- 外部 GitLab：主要用于与客户进行代码同步，使用人员为 AE 和 FAE 人员。

# 1.4 GitLab 权限

- Private: 只有自己和加入的人员可见
- Internal: 所有有 GitLab 的人员均可见
- Public: 完全开放, 包括外部人员



## 2. 常用图形化工具

## 2. 常用图形化工具

- gitk: Git 原生工具
- SmartGit: 一款跨平台的商用软件, 支持 Windows/Linux/Mac 系统, 我们暂时可以使用 non-commercial 版本
- SourceTree: 免费, 公认的 Git 管理神器, 支持 Windows/Mac 系统

# 3. 常用 Git 命令

命令	描述	命令	描述
git add	添加至暂存区	git fetch	获取远程版本库的提交
git bisect	二分查找	git log	显示提交日志
git branch	分支管理	git merge	分支合并
git checkout	检出到工作区、切换或创建分支	git mv	重命名
git cherry-pick	提交拣选	git pull	拉回远程版本库的提交
git clean	清除工作区未跟踪文件	git push	推送至远程版本库
git clone	克隆版本库	git rebase	分支变基
git commit	提交	git reset	重置改变分支“游标”指向
git stash	保存和恢复进度	git revert	反转提交
git diff	差异比较	git rm	删除文件

# 4. 常用概念

## 4.1 Submodule

一个仓库里需要用到另外一个仓库作为 submodule 的场景，可查看 esp-idf 下的一个隐藏文件 `.gitmodules`，里面包含了当前 idf 使用的 submodule。

- 在项目中添加 submodule:
  - `Git submodule add https://github.com/espressif/esp-idf.git`
- 获取具有 submodule 的仓库:
  - `Git clone --recursive https://github.com/espressif/esp-idf.git`

## 4.2 Ignore

查看 esp-idf 目录下的一个隐藏文件 `.gitignore`。

# 5. Git 管理策略

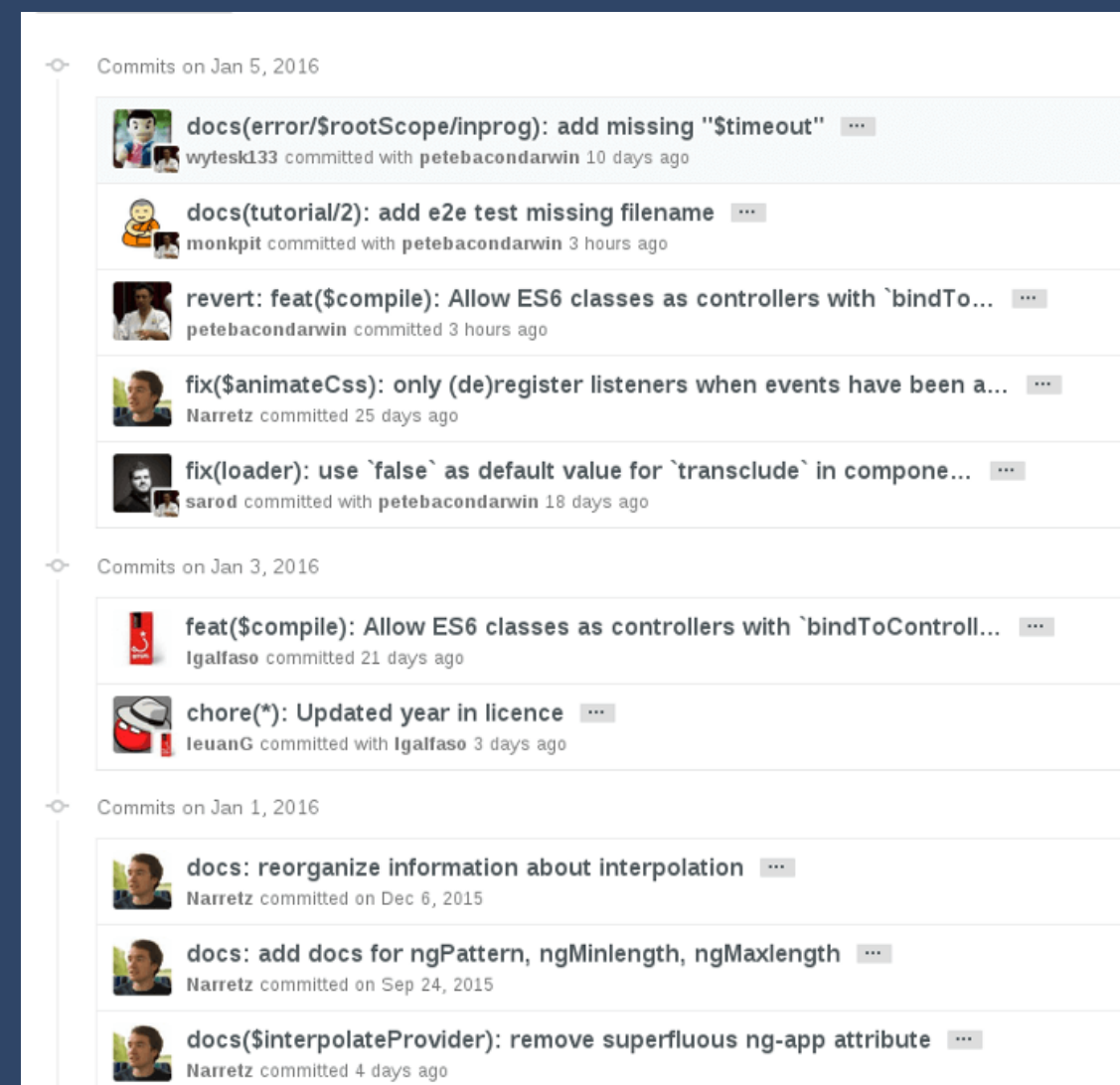


# 5.1 分支管理

- Master 分支
  - 不要直接往 master 分支上提交：
    - 不能保证提交的正确性（包括编译和测试的正确性）
    - 不能看出提交的关联性
- Feature 分支
  - 命名方式：feature/xxxx, bugfix/xxxx, 或以类似组命名
  - 每个分支只干 xxxx 描述的事

## 5.2 Commit 策略

目前，社区有多种 Commit message 的写法规范。推荐使用 Angular 规范，这是目前使用最广的写法，比较合理和系统化。



## 5.2.1 Commit Message 的好处

- 首先，提供更多的历史信息，方便快速浏览
- 比如，下面的命令显示上次发布后的变动，每个 commit 占据一行。你只看行首，就知道某次 commit 的目的

```
$ git log <last tag> HEAD --  
pretty=format:%s
```

```
fix(ng-bind-html): watch string value instead of wrapper (4 days ago) <Chirayu Krishnappa>  
chore(sce): remove unused function (4 days ago) <Chirayu Krishnappa>  
fix(ngInclude): don't break attribute bindings on ngInclude-ed element  
fix(ngView): IE8 regression due to expando on non-element nodes (4 days ago) <Dave Peticolas>  
docs(ngClass): fix grammar (4 days ago) <Dave Peticolas>  
docs(ngCloak): fix grammar, clarity (4 days ago) <Dave Peticolas>  
docs(ngModelController): clarify issue with isolated scope directive (5 days ago) <Dave Peticolas>  
docs(input): fix spelling error and reword for clarity (5 days ago) <Dave Peticolas>  
doc(ngApp): fix grammar (5 days ago) <Dave Peticolas>  
(origin/g3_v1_2) docs($exceptionHandler): add an example of overriding  
docs(FAQ): update jQuery compatibility (5 days ago) <Pete Bacon Darwin>  
docs(guide/services): rewording of explanation (5 days ago) <Pete Bacon Darwin>  
docs(guide/services): explain services in plain language (5 days ago) <Pete Bacon Darwin>  
docs(guide/i18n): change non-existent de-ge to de-de (5 days ago) <Maarten de Groot>  
docs(ngForm): fix grammar and improve explanation (5 days ago) <Dave Peticolas>  
docs: update sticker ordering info in FAQ (6 days ago) <naomiblack>  
docs(ngResource): fix typo (6 days ago) <Tim Statler>  
docs(ngShowHide): fix typo (6 days ago) <Roberto Bonvallet>  
docs(guide/$location): describe workaround for collocated apps (6 days ago) <Dave Peticolas>  
docs(tutorial/step_03): add info about karma-ng-scenario plug-in (6 days ago) <Dave Peticolas>  
fix(scenario): include "not " in error messages if test is inverted (6 days ago) <Dave Peticolas>  
fix($parse): disallow access to window and dom in expressions (7 days ago) <Dave Peticolas>
```

## 5.2.1 Commit Message 的好处

- 其次，可以过滤某些 commit（比如文档改动），便于快速查找信息

```
$ git log <last release> HEAD  
--grep feature
```

- 此外，还可以直接从 commit 生成 Change log，也就是用于说明与上一个版本差异的文档

```
fix(ng-bind-html): watch string value instead of wrapper (4 days ago) <  
chore(sce): remove unused function (4 days ago) <Chirayu Krishnappa>  
fix(ngInclude): don't break attribute bindings on ngInclude-ed element  
fix(ngView): IE8 regression due to expando on non-element nodes (4 days  
docs(ngClass): fix grammar (4 days ago) <Dave Peticolas>  
docs(ngCloak): fix grammar, clarity (4 days ago) <Dave Peticolas>  
docs(ngModelController): clarify issue with isolated scope directive (5  
docs(input): fix spelling error and reword for clarity (5 days ago) <ts  
doc(ngApp): fix grammar (5 days ago) <Dave Peticolas>  
(origin/g3_v1_2) docs($exceptionHandler): add an example of overriding  
docs(FAQ): update jQuery compatibility (5 days ago) <Pete Bacon Darwin>  
docs(guide/services): rewording of explanation (5 days ago) <Pete Bacon  
docs(guide/services): explain services in plain language (5 days ago) <  
docs(guide/i18n): change non-existent de-ge to de-de (5 days ago) <Maar  
docs(ngForm): fix grammar and improve explanation (5 days ago) <Dave Pe  
docs: update sticker ordering info in FAQ (6 days ago) <naomiblack>  
docs(ngResource): fix typo (6 days ago) <Tim Statler>  
docs(ngShowHide): fix typo (6 days ago) <Roberto Bonvallet>  
docs(guide/$location): describe workaround for collocated apps (6 days  
docs(tutorial/step_03): add info about karma-ng-scenario plug-in (6 day  
fix(scenario): include "not " in error messages if test is inverted (6  
fix($parse): disallow access to window and dom in expressions (7 days a
```

## 5.2.2 Commit Message 的格式

每次提交，Commit message 都包括三个部分：

- Header (必须)
- Body (可省略)
- Footer (可省略)
- 无论哪一个部分，任何一行都不得超过 72 个字符（或 100 个字符），从而避免自动换行，影响美观

## 5.2.2 Commit Message 的格式

### 5.2.2.1 Header

Header 部分只有一行，包括：

- type（必需）
- scope（可省略）
- subject（必需）

```
<type>(<scope>): <subject>
```

```
// blank
```

```
<body>
```

```
// blank
```

```
<footer>
```

# 5.2.2 Commit Message 的格式

## 5.2.2.1 Header

**Type:** 用于说明 commit 的类别，只允许使用下面7个标识。

- feat: 新功能 (feature)
- fix: 修补 bug
- docs: 文档 (documentation)
- style: 格式 (不影响代码运行的变动)
- refactor: 重构 (即不是新增功能，也不是修改 bug 的代码变动)
- test: 增加测试
- chore: 构建过程或辅助工具的变动

其中，如果 *type* 为 *feat* 和 *fix*，则该 *commit* 将肯定出现在 *Change log* 之中。其他情况 (*docs*、*chore*、*style*、*refactor*、*test*) 由你决定要不要放入 *Change log* (建议不要)。



# 5.2.2 Commit Message 的格式

## 5.2.2.1 Header

- **Scope**
  - 用于说明 commit 影响的范围，比如数据层、控制层、视图层等等，视项目不同而不同
- **Subject**
  - 是 commit 目的的简短描述，不超过 50 个字符
  - 以动词开头，使用第一人称现在时，比如 change，而不是 changed 或 changes
  - 第一个字母大写
  - 结尾不加句号 (.)



## 5.2.2 Commit Message 的格式

### 5.2.2.2 Body

Body 部分是对本次 commit 的详细描述，应该说明代码变动的动机，以及以前行为的对比。具体：

- 可以分成多行，使用列表形式
- 尽可能限制在 72 个字符
- 使用悬挂式缩紧
- 以动词开头，使用第一人称现在时，比如 change，而不是 changed 或 changes

## 5.2.2 Commit Message 的格式

### 5.2.2.3. Footer

- 不兼容变动：如果当前代码与上一个版本不兼容，则 footer 部分以 BREAKING CHANGE 开头，后面是对变动的描述、以及变动理由和迁移方法。

**BREAKING CHANGE:** isolate scope bindings definition has changed.

To migrate the code follow the example below:

**Before:**

```
scope: {  
  myAttr: 'attribute',  
}
```

**After:**

```
scope: {  
  myAttr: '@',  
}
```

The removed `inject` wasn't generally useful for directives so there should be no code using it.

## 5.2.2 Commit Message 的格式

### 5.2.2.3. Footer

- 关闭单个 issue
- 关闭多个 issue

其中 *teamwork* 中的问题采用 *TW1234* 的命名方式，*GitHub* 或 *GitLab* 中的问题，采用 *#123* 的命名方式。

**Closes TW1234**

**or**

**Closes #123**

**Closes #123, #245, #992, TW1234**

# 5.2.2 Commit Message 的格式

## 5.2.2.4. Revert

- 如果当前 commit 用于撤销以前的 commit, 则必须以 `revert:` 开头, 后面跟着被撤销 commit 的 header;
- Body 部分的格式是固定的, 必须写成 `This reverts commit <hash>.`, 其中的 hash 是被撤销 commit 的 SHA 标识符。

```
revert: feat(pencil): add 'graphiteWidth' option
```

```
This reverts commit 667ecc1654a317a13331b17617d973392f415f02.
```

注意:

1. 如果当前 *commit* 与被撤销的 *commit*, 在同一个发布 (*release*) 里面, 那么它们都不会出现在 *Change log* 中;
2. 如果两者在不同的发布, 那么当前 *commit*, 会出现在 *Change log* 的 *Reverts* 小标题下面。

# 5.3 MR 策略

## 5.3.1 Submit MR

当一个正在开发的 feature 或 bugfix 分支完成后：

- 需要在 GitLab 提交 Merge Request, 请求合并入 master 或其他分支；
- 需要注意提 MR 时填写的信息。

## 5.3.2 Review MR

当一个 MR Submit 后，进入审阅流程：

- 鼓励每个参与项目的开发者进行审阅，并进行评论；
- 如果觉得 OK，给个 "Thumb's up"，当至少有 2 个 "Thumb's up"，并且讨论结束，即可接受 merge。

## 5.3.3 Update MR

在 review 过程中，需要对提的意见进行修改：

- 只需在原分支上进行修改，一定不要删除分支，重新提交；
- 最终合入 master 的提交记录要清晰明了。



## 5.3.4 解决冲突

在解决冲突时：

- 将 master 分支合入到当前分支；
- rebase 到 master 分支，解决冲突。

# 6. 周报系统

[https://glab.espressif.cn/applicationteam/  
applicationteamweeklyreport](https://glab.espressif.cn/applicationteam/applicationteamweeklyreport)

# 7. 代码风格

<https://gitlab.espressif.cn:6688/idf/esp-idf/blob/master/docs/contribute/style-guide.rst>