Git 管理规范

由 张华斌创建, 最后修改于大约1小时以前

- 1 为什么
- 2 怎么做
  - 2.1 版本语义化
  - 2.2 分支模型
    - 2.3.1 分支命名
    - 2.3.2 功能分支管理
    - 2.3.3 多版本并行
  - 2.4 提交规范
  - 2.5 错误范例
    - 2.5.1 范例1
    - 2.5.2 范例2
    - 2.5.3 范例3
- 3 开发工具
- 4 参考资料

本文讲述了前端项目的 Git 管理规范，仅适用于前端业务项目，同时也适用于公共库模块。

为什么

1. 降低项目出错的几率；
2. 方便日后 Reviewing Code；
3. 能很好的提升项目整体质量；

怎么做

1. 版本语义化；
2. 统一团队的分支管理方式；
3. 统一团队的提交日志格式；
4. 更有意义的命名方式；

版本语义化

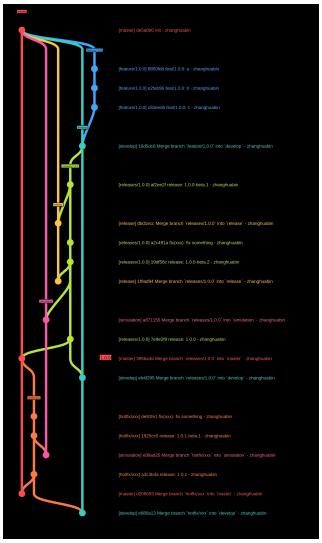
版本格式：主版本号.次版本号.修订号，版本号递增规则如下：

1. 主版本号：当你做了不兼容的 API 修改，
2. 次版本号：当你做了向下兼容的功能性新增，
3. 修订号：当你做了向下兼容的问题修正。

先行版本号及版本编译元数据可以加到“主版本号.次版本号.修订号”的后面，作为延伸先加上一个连接号再加上一连串以句点分隔的标识符来修饰。

范例：见下文的分支模型。

分支模型



对比旧的分支管理策略 Git分支工作流程，新的分支模型主要引入了 release/x.x.x 和 develop 这两个分支，相较而言有以下几点优势：

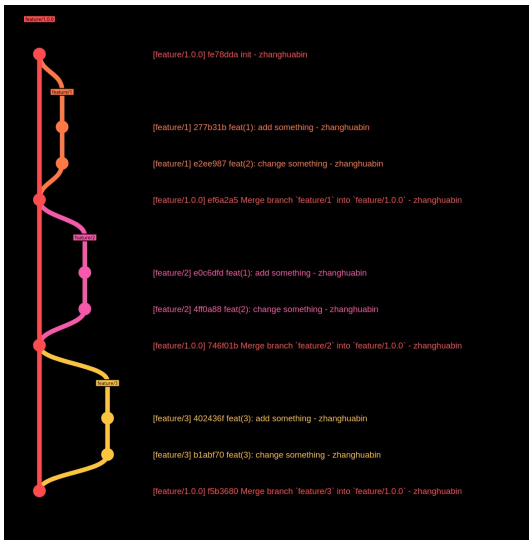
1. 避免在 release，simulation，master 分支上重复构建发布，避免 CI 重复做代码推送；
2. 每一次构建发布且合并到 release，simulation，master 就是一次部署，使得这三个环境的分支变得更有意义（***我是用来部署的***），也使得部署历史更加清晰；
3. 每次发布都伴随着项目的版本升级，使得项目版本语义化，更容易分析项目的变更历史；
4. 在多版本并行开发时，项目更容易管理；

分支命名

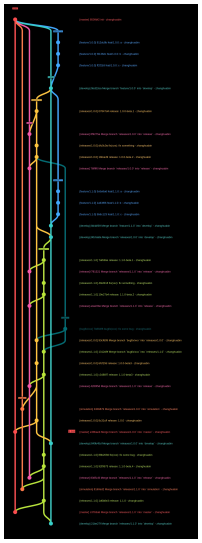
主要参考 Git Flow

- 主分支：master
- 预发分支：simulation
- 测试分支：release
- 开发分支：develop
- 版本分支：feature/x.x.x
- 功能分支：feature/xxx，功能分支的命名禁止使用自己的姓名，尽量使用有意义的名称，可以使用 https://unbug.github.io/codelf 查询相关的命名方式
- 发布分支：releases/x.x.x
- bug 分支：bugfix/xxx，xxx 一般表示 bug id
- 线上问题分支：hotfix/xxx，xxx 一般表示 bug id

功能分支管理



多版本并行



提交规范

推荐遵从 Angular 的 commit 规范 Git Commit Guidelines

```
<type>(<scope>): <subject>
<BLANK LINE>
<body>
<BLANK LINE>
<footer>
```

- type
  - feat: 新功能
  - fix: 修复问题
  - docs: 修改文档
  - style: 修改代码格式，不影响代码逻辑
  - refactor: 重构代码，理论上不影响现有功能
  - perf: 提升性能
  - test: 增加修改测试用例
  - chore: 修改工具相关（包括但不限于文档、代码生成等）
- 修改文件的范围，比如：视图层、控制层、docs、config, plugin
- subject：subject 是 commit 目的的简短描述（用一句话清楚的描述这次提交做了什么），不超过50个字符
- body：补充 subject 添加详细说明，可以分成多行，适当增加原因、目的等相关因素，也可不写
- footer：当有非兼容修改(Breaking Change)时必须在这里描述清楚

错误范例

范例**1**

1. 合并的时候通常提交信息不符合格式规范，合并分支的提交通常为 “Merge branch "xxx" into xxx”，在查看历史的时候更容易识别出是从哪个分支合并到哪里，对应分支的命名可以突显出合并的分支是做什么的；
2. 打包提交信息不合符规范，项目中重复的出现 “某某某打包”，在 Git 提交历史中出现大量的这种提交信息，影响 review 代码；

范例**2**

多人协作开发时尽量创建功能分支，开发好后再合并到主分支或版本分支。上图示例中，一个人在主分支提交了更改要 push 的时候，git 提示不允许提交，远程仓库有更新的提交，这时候 pull 后，有遇到冲突时需要处理冲突，处理完生成一个新的提交“Merge branch feature/x.x.x of gitlab.meiou...”，意思大致就是将远程的分支合并到当前的本地仓库，最后在提交，这种方式导致 Git 提交更加的混乱

范例**3**

在对应分支需要修改或者提交代码时，需要先 pull 在合并和提交。

开发工具

- https://github.com/commitizen/cz-cli
- Git Flow

参考资料

- Git分支工作流程
- A successful Git branching model
- 语义化版本
- 更优雅的使用 Git
- Alpha、Beta、RC、GA版本的区别
- Angular.js Git Commit Guidelines

scope

无标签