



TJTT乒乓球一体化小程序开发流程（技术端工作规范）

写给TJTT工作室技术端

✎ 撰写：@Charlie. BU 日期：2024年8月3日


！ 注意：本文档为TJTT Studio内部资料，受严格的保密协议保护，未经允许不得外传、截图传播！

准备工作

熟悉相关平台账号

- **Coding**： [DevOps_DevOps 解决方案_一站式 DevOps_开发者工具](#) | [腾讯云 CODING DevOps](#)
 - 工作室管理
 - 工作安排与协同
 - 知识库
 - 代码管理
 - CI/CD

💡 注册后更换头像，方便区分

- **阿里云**： [阿里云登录 - 欢迎登录阿里云，安全稳定的云计算服务平台](#)
 - 服务器
 - 数据库
 - **tjtt.top**、**tjtt.asia**域名及备案
 - OSS数据存储

🔑 账号：TJTTGroup

密码：*****

- 宝塔：🔗 <https://120.55.171.190:8888/4b7af04e>
 - 服务器管理
 - 项目部署
 - 数据库操作

🔑 账号：TJTT

密码：*****

- 微信公众平台：👤 微信公众平台
 - 小程序管理
 - 代码审核、部署、上线
 - 域名管理
 - 小程序备案
 - 小程序数据分析
- 工作室邮箱：📧 163网易免费邮-你的专业电子邮局
 - 用户密码找回
 - 用户反馈

🔑 账号：TJTTGroup@163.com

密码：*****

需求收集流程

- TJTT工作室成员有新的需求产生时，可将需求描述后写入TJTT Platform项目—创建事项—需求
- 定期check TJTT邮箱TJTTGroup@163.com，收集用户需求；用户有新的需求产生时，由收集者将需求描述后写入TJTT Platform项目—创建事项—需求
- 技术端成员定期评估需求是否能够满足，若不能满足，上报@Charlie. BU后删除需求

Bug收集流程

- 定期check TJTT邮箱TJTTGroup@163.com，收集用户反馈Bug；由收集者将Bug描述后写入TJTT Platform项目—创建事项—缺陷

- TJTT工作室成员发现系统Bug，及时上报@Charlie. BU或其他技术端成员，由发现者将Bug描述后写入TJTT Platform项目—创建事项—缺陷

开发流程

开发准备

- 前端IDE：HbuilderX
- 后端IDE：PyCharm或VS Code
- 数据库GUI：Navicat
- 服务器：阿里云轻量应用服务器—华东2（上海）：120.55.171.190

可用ssh远程连接：

```
ssh root@120.55.171.190
```

处理任务

- 在TJTT Platform项目—项目协同中查找处理人为“我”的事项
- 将状态标记为开发中
- 发现有block，在描述中填写，并告知@Charlie. BU

首次开发git配置

- 配置SSH密钥

参考：[🔗 如何配置SSH密钥及自定义SSH认证密钥的路径_云效\(Apsara Devops\)-阿里云帮助中心](#)

SSH密钥对一般存放在本地用户的根目录下。

Linux、Mac请直接使用以下命令查看已存在的公钥，Windows 用户在WSL（需要 windows10 或以上）或 [Git Bash](#)下使用以下命令查看已生成公钥：

```
cat ~/.ssh/id_rsa.pub
```

将公钥复制进入Coding个人账户设置—SSH公钥，新增公钥

- 在本地新建目录FE-TJTT或BE-TJTT，cmd进入该路径下进行 `git clone <代码仓库地址>`
- 进入IDE进行开发

Git Flow

参考：[🔗 Git之GitFlow工作流 | Gitflow Workflow\(万字整理，已是最详\)_gitflow工作流程-CSDN博客](#)

分支说明

1. master/production：生产分支

用途：主要分支上存放的是最稳定、最新的发布版本，该分支的代码应该是在**生产环境**中使用的代码。

！ master分支只能从其他分支合并，不能直接修改

合并方向：release/* → master ; hotfix/* → master ;

2. release/{版本号}：预发布分支

用途：基于develop分支**创建（不是合并）**，发布即合并到master分支

创建方向：develop → release/*（是**创建**不是合并）

3. develop：开发分支

用途：平时开发用的主分支，永远是功能**最全最新且没有Bug**。当全部新功能测试完毕/Bug修复完毕决定上线时，直接合并到release分支

合并方向：feature/* → develop ; hotfix/* → develop

4. feature/{功能描述}：功能分支

用途：用于开发即将发布版本或未来版本的新功能或者探索新功能，未开发完成切勿合并到develop分支

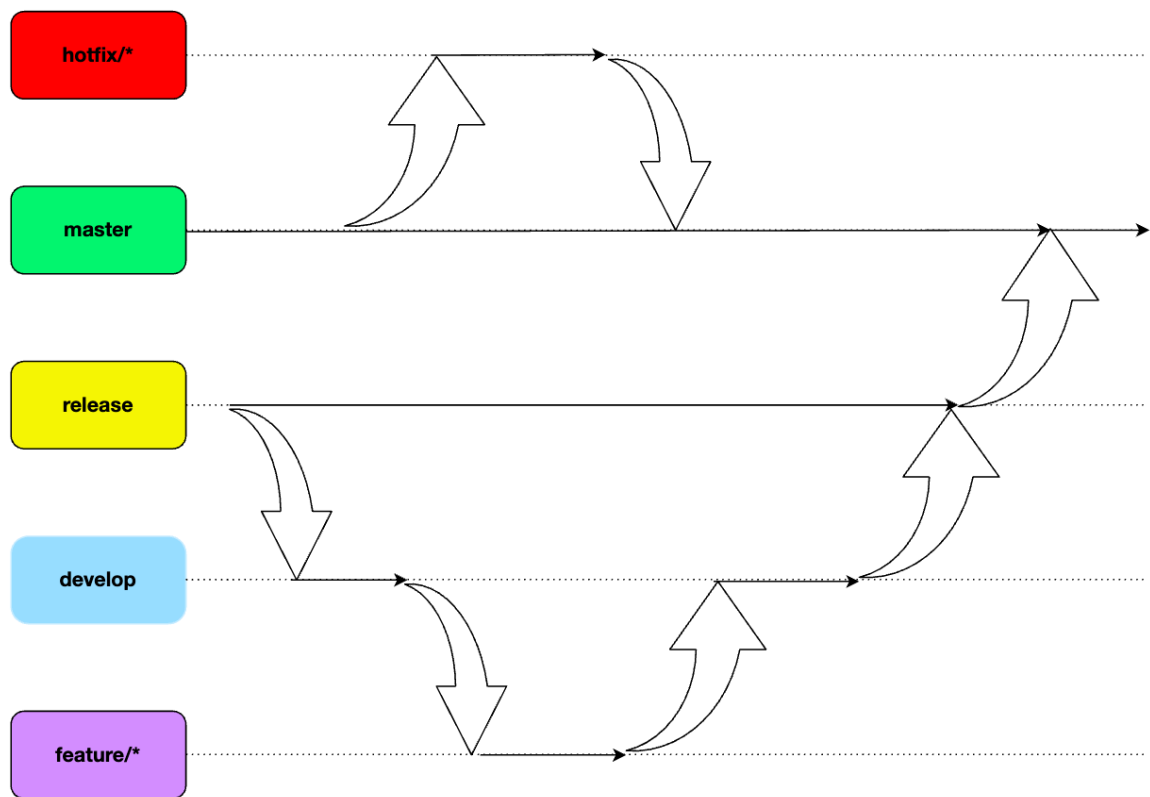
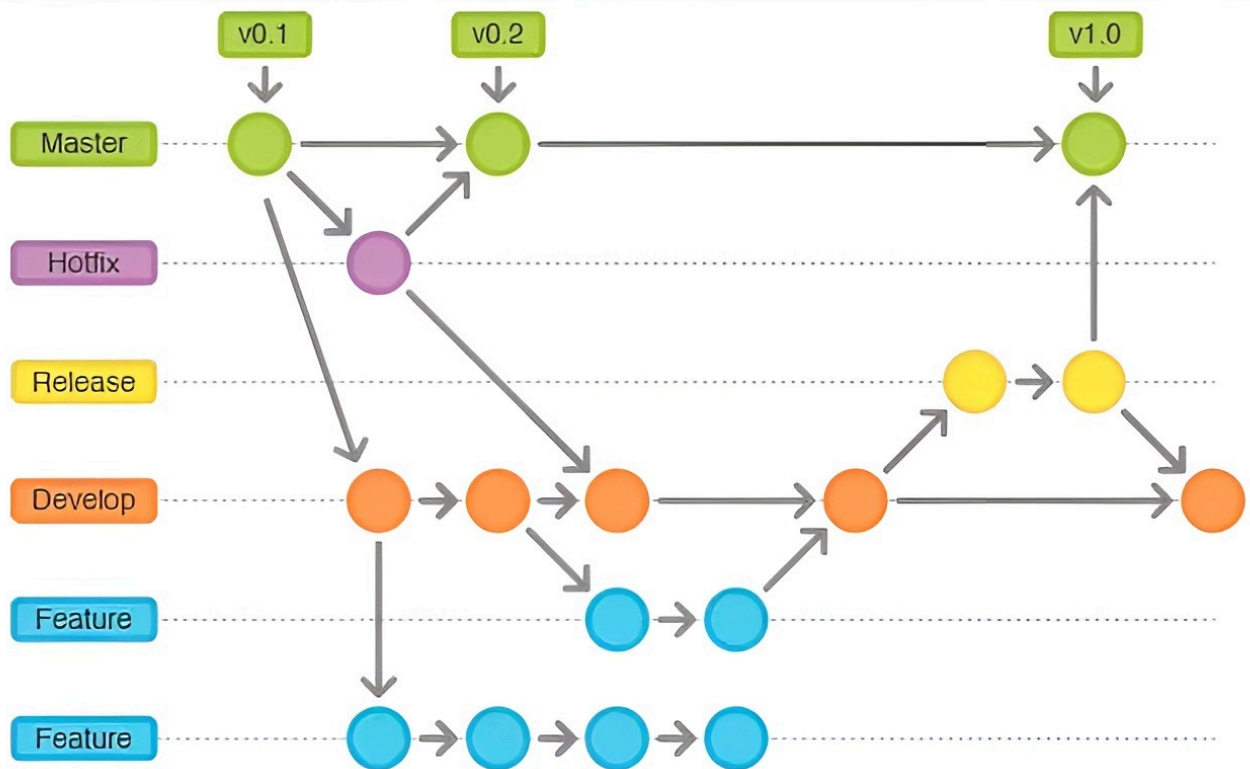
合并方向：develop → feature/*

5. hotfix/{bug描述}：热修复分支

用途：当生产环境的代码（master分支上代码）遇到严重到必须立即修复的Bug时，需要从master分支拉取hotfix分支进行代码的紧急修复，通常修复并测试通过后可直接合并到master分支，亦可合并到develop分支

合并方向：master → hotfix/*

• Git Flow示意图



• 版本号说明

版本号通常为 **x.y.z**

序号	格式要求	说明
x	非负整数	主版本号(major), 进行不向下兼容的修改时, 递增主版本号
y	非负整数	次版本号(minor), 保持向下兼容, 新增特性时, 递增次版本号
z	非负整数	修订号(patch), 保持向下兼容, 修复问题但不影响特性时, 递增修订号



0.y.z 表示开发阶段，一切可能随时改变，非稳定版。

1.0.0 界定此版本为初始稳定版，后面的一切更新都基于此版本进行修改。

前端



鉴于HBuilderX的git版本管理功能不规范，拟使用命令行进行git管理

1. 从代码仓库拉取代码

```
git remote add origin <代码仓库地址>          # 若本地未关联远程代码仓库需执行
git pull -u origin master
# 或
git pull -u origin develop
# 或
git pull --all          # 拉取远程库的全部分支
# -u表示与远程库的该分支建立跟踪关系，之后git pull和git push无需指定仓库和分支
```

2. 在cmd新建分支

```
git checkout develop
git checkout -b feature/*
```

或

```
git checkout master
git checkout -b hotfix/*
```

3. 在HBuilderX中进行开发

4. 初步开发完成后

```
git add .
git commit -m '<本次提交message>'
```



提交messgae规范

- feat: 功能描述
- hotfix: Bug描述
- chore: 描述

5. 在TJTT Platform项目—项目协同将需求状态标记为测试中，进行测试；若测试过程存在代码变动，需重复步骤4

6. 测试无误后，将代码推至远程仓库

```
git push -u origin feature/*
```

或

```
git push -u origin hotfix/*
```

！ 注意


- 切勿直接将本地master、develop分支推入远程库
- 切勿执行git push --all
- 切勿私自在本地对代码进行merge
- 之后每次正式开发前，都需重新拉取远程库的master、develop分支

7. 在Coding代码仓库创建MR（合并请求）

严格遵循Git Flow中分支合并规则

8. MR经@Charlie. BU审核后，合并到预发布分支release/*
9. 预发布分支release/*合并到生产分支master
10. master分支在微信开发者工具发版

后端

 推荐使用PyCharm或VS Code的git版本管理

1. 在PyCharm或VS Code中新建feature/*分支或hotfix/*分支、进行开发
2. 初步开发完成后，在PyCharm—Git或VS Code—源代码管理中提交代码至本地git仓库（不push）

提交messgae规范

- feat: 功能描述
- hotfix: Bug描述
- chore: 描述

3. 在TJTT Platform项目—项目协同将需求状态标记为测试中，进行测试；若测试过程存在代码变动，需重复步骤2
4. 测试无误后，在PyCharm—Git或VS Code—源代码管理中代码推至远程仓库

！ 注意

- 切勿直接将本地master、develop分支推入远程库

- 切勿执行git push --all
- 切勿私自在本地对代码进行merge
- 之后每次正式开发前，都需重新拉取远程库的master、develop分支

5. 在Coding代码仓库创建MR（合并请求）

严格遵循Git Flow中分支合并规则

6. MR经@Charlie. BU审核后，合并到预发布分支release/*
7. 预发布分支release/*合并到生产分支master
8. 在服务器（120.55.171.190）拉取远程仓库master分支进行部署

前后端联调与上线

1. 前端代码提交微信公众平台后，在版本管理—开发版本中选为体验版本，交由技术端成员测试
2. 若发现前端Bug：添加事项至TJTT Platform项目—项目协同，执行上述Bug热修复流程
3. 若发现后端Bug：服务器及时回滚至上一稳定版，添加事项至TJTT Platform项目—项目协同，执行Bug热修复流程
4. 测试通过，在微信公众平台将体验版本提审，通过后发布线上版本，在TJTT Platform项目—项目协同将需求状态标记为已完成
5. 在 TJTT Studio 2.0 重大工作节点—微信小程序TJTT乒乓球一体化平台迭代中添加新版本介绍
6. 若微信代码审核未通过，添加事项至TJTT Platform项目—项目协同，执行上述Bug热修复流程

数据库操作

！ 高危操作，务必上报@Charlie. BU！

• 方式一

Navicat连接远程数据库。可修改数据内容

Database: MySQL
Host: 120.55.171.190
Port: 3306
User Name: Charlie
Password: *****

• 方式二

PyCharm、VS Code等IDE中连接远程数据库。可修改数据内容

- 方式三

宝塔终端使用phpMyAdmin对数据库进行操作。可修改**数据内容、表结构、字段、属性**，操作前务必在Coding**创建事项**，并通过@[Charlie. BU](#)审批

上海同乒体育科技有限公司

TJTT Studio