

从所有教程的词条中查询...

首页 > 慕课教程 > Go工程师体系课全新版 > 2. git的简单规范

全部开发者教程

1. go最常用的设计模式 - 函数选项

2. 单例模式和懒加载

3. 测试金字塔

第23周 protoc插件开发、cobra命令行

1. protoc调试源码

2. protoc自定义gin插件

第24周 log日志包设计

日志源码

第25周 ast代码生成工具开发

错误码

第26周 三层代码结构

通用app项目启动



bobby · 更新于 2022-11-03

上一节 1. 有哪些规范我... 3. go代码规范 下一节

1. git想要使用好，最好是理解git的大概原理，git的规范使用基本上可以出一门课，所以我们只给大家简单介绍git的规范，各个规范我会给出网上查询到的比较好的文章，请轻喷
2. git本身学习很枯燥，所以最好的办法就是强迫自己平时遵循这些规范，养成好习惯
3. 项目开发中最常用的是用git进行管理，git对于多人开发中有一套成熟的分支管理

1. 现在有5个人

1. a和c突然要开发一个功能A
2. b和d这个时候业开发一个功能D
3. e和a也要合作开发一个功能C
4. 微服务不适合太多人开发一个微服务 一个微服务10个人开发

2. 绝大部分小公司都仍然会采用svn式的管理，无可厚非
3. 多人开发的大项目都会采用标准的开发管理，减少代码冲突，大家进入以后直接遵循即可
4. 如果一两个人开发一个项目简单化就可以了
5. 参考这篇：<https://zhuanlan.zhihu.com/p/23478654>

4. commit规范

1. commit规范很重要，不论是多人开发还是自己开发，否则后期你自己看代码都得晕， 而且还能帮我们生成changelog， commit的log生成changelog的步骤可以参考：  
<https://cloud.tencent.com/developer/article/1755709>
2. 很多时候为了方式代码丢失本地会进行进行commit， 要学会merge commitlog

1. 举个例子说明一下：

1. 我们现在开发了一个订单接口服务，但是开发中，我们commit了一个log：feature-order：增加生成订单接口，接下来我们发现了一个bug，然后继续commit：feature-order：修复了bug1，然后又发现了一个bug，继续commit：feature-order：修复了bug2，然后又增加了log记录，生成commit：feature-order：增加日志打印，然后就成这样了：

1. feature-order：增加生成订单接口
2. feature-order：修复了bug1
3. feature-order：修复了bug2
4. feature-order：增加日志打印

2. 我们发布的版本是V1.2，生成changelog：

1. feature-order：增加生成订单接口
2. feature-order：修复了bug1

意见反馈

收藏教程

标记书签

4. feature-order: 增加日志打印

其实这就是一个低质量的commit log，这些都可以合并成一个commit，我们并不关心你中途的所有commit log

合并commit log可以参考：<https://cloud.tencent.com/developer/article/1690638>

commit规范可以参考：<https://blog.csdn.net/xmcy001122/article/details/123372325>

goland的插件：



1. 有哪些规范我们应该遵循 ◀ 上一节      下一节 ▶ 3. go代码规范

我要提出意见反馈

