散尽浮华

纵浪大化中，不喜亦不惧；重剑无锋，大巧不工！

昵称：散尽浮华
园龄：3年11个月
粉丝：2494
关注：23
+加关注

Git忽略提交规则 - .gitignore配置运维总结

在使用Git的过程中，我们喜欢有的文件比如日志，临时文件，编译的中间文件等不要提交到代码仓库，这时就要设置相应的忽略规则，来忽略这些文件的提交。简单来说一个场景：在你使用git add .的时候，遇到了把你不想提交的文件也添加到了缓存中去的情况，比如项目的本地配置信息，如果你上传到Git中去其他人pull下来的时候就会和他本地的配置有冲突，所以这样的个性化配置文件我们一般不把它推送到git服务器中，但是又为了偷懒每次添加缓存的时候都想用git add .而不是手动一个一个文件添加，该怎么办呢？很简单，git为我们提供了一个.gitignore文件只要在这个文件中申明那些文件你不希望添加到git中去，这样当你使用git add .的时候这些文件就会被自动忽略掉。

有三种方法可以实现忽略Git中不想提交的文件：

1）在Git项目中定义.gitignore文件

对于经常使用Git的朋友来说，.gitignore配置一定不会陌生。这种方式通过在项目的某个文件夹下定义.gitignore文件，在该文件中定义相应的忽略规则，来管理当前文件夹下的文件的Git提交行为。.gitignore 文件是可以提交到公有仓库中，这就为该项目下的所有开发者都共享一套定义好的忽略规则。在.gitingore 文件中，遵循相应的语法，在每一行指定一个忽略规则。如：

```
*.log
*.temp
/vendor
```

2）在Git项目的设置中指定排除文件

这种方式只是临时指定该项目的行为，需要编辑当前项目下的 .git/info/exclude文件，然后将需要忽略提交的文件写入其中。需要注意的是，这种方式指定的忽略文件的根目录是项目根目录。

3）定义Git全局的 .gitignore 文件

除了可以在项目中定义 .gitignore 文件外，还可以设置全局的git .gitignore文件来管理所有Git项目的行为。这种方式在不同的项目开发者之间是不共享的，是属于项目之上Git应用级别的行为。这种方式也需要创建相应的 .gitignore 文件，可以放在任意位置。然后在使用以下命令配置Git：

```
# git config --global core.excludesfile ~/.gitignore
```

首先要强调一点，这个文件的完整文件名就是".gitignore"，注意最前面有个"."。一般来说每个Git项目中都需要一个".gitignore"文件，这个文件的作用就是告诉Git哪些文件不需要添加到版本管理中。实际项目中，很多文件都是不需要版本管理的，比如Python的.pyc文件和一些包含密码的配置文件等等。这个文件的内容是一些规则，Git会根据这些规则来判断是否将文件添加到版本控制中。

Git忽略文件的原则

- 忽略操作系统自动生成的文件，比如缩略图等；
- 忽略编译生成的中间文件、可执行文件等，也就是如果一个文件是通过另一个文件自动生成的，那自动生成的文件就没必要放进版本库，比如Java编译产生的.class文件；
- 忽略你自己的带有敏感信息的配置文件，比如存放口令的配置文件。

.gitignore文件的使用方法

首先，在你的工作区新建一个名称为.gitignore的文件。
然后，把要忽略的文件名填进去，Git就会自动忽略这些文件。
不需要从头写.gitignore文件，GitHub已经为我们准备了各种配置文件，只需要组合一下就可以使用了。

有时对于git项目下的某些文件，我们不需要纳入版本控制，比如日志文件或者IDE的配置文件，此时可以在项目的根目录下建立一个隐藏文件 .gitignore（linux下以.开头的文件都是隐藏文件），然后在.gitignore中写入需要忽略的文件。

```
[root@kevin ~]# cat .gitignore
*.xml
*.log
*.apk
```

.gitignore注释用'#'，*表示匹配0个或多个任意字符，所以上面的模式就是要忽略所有的xml文件,log文件和apk文件。

< 2020年3月 >

日 一 二 三 四 五 六
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31 1 2 3 4
5 6 7 8 9 10 11

搜索
找找看
谷歌搜索

常用链接
我的随笔
我的评论
我的参与
最新评论
我的标签

随笔分类
Ansible(5)
Apache(6)
Ceph(4)
ClusterShell(1)
DNS(5)
Docker(30)
DRBD(3)
Elasticsearch(7)
Etcd
Expect(2)
Fabric(1)
FastDFS(1)
FTP(4)
GlusterFS (5)
Haproxy(6)
IP SAN(1)
Iptables(6)
Jenkins(8)
Jira and Confluence(7)
Jumpserver(4)
Kafka(2)
LB/HA高可用(23)
LDAP(2)
LVM(3)
LVS(5)
Maven/Nexus(1)
Memcached(4)
MongoDB(11)
MooseFS(2)

.gitignore配置文件用于配置不需要加入版本管理的文件，配置好该文件可以为版本管理带来很大的便利。

.gitignore忽略规则的优先级

在 .gitingore 文件中，每一行指定一个忽略规则，Git检查忽略规则的时候有多个来源，它的优先级如下（由高到低）：

1）从命令行中读取可用的忽略规则
2）当前目录定义的规则
3）父级目录定义的规则，依次递推
4）$GIT_DIR/info/exclude 文件中定义的规则
5）core.excludesfile中定义的全局规则

.gitignore忽略规则的匹配语法

在 .gitignore 文件中，每一行的忽略规则的语法如下：

1）**空格**不匹配任意文件，可作为分隔符，可用反斜杠转义
2）以"**#**"开头的行都会被 Git 忽略。即#开头的文件标识注释，可以使用反斜杠进行转义。
3）可以使用标准的**glob**模式匹配。所谓的glob模式是指shell所使用的简化了的正则表达式。
4）以斜杠"**/**"开头表示目录；"/"结束的模式只匹配文件夹以及在该文件夹路径下的内容，但是不匹配该文件；"/"开始的模式匹配项目跟目录；如果一个模式不包含斜杠，则它匹配相对于当前 .gitignore 文件路径的内容，如果该模式不在 .gitignore 文件中，则相对于项目根目录。
5）以星号"**"通配多个字符，即匹配多个任意字符；使用两个星号"**" 表示匹配任意中间目录，比如`a/**/z`可以匹配 a/z, a/b/z 或 a/b/c/z等。
6）以问号"**?**"通配单个字符，即匹配一个任意字符；
7）以方括号"**[]**"包含单个字符的匹配列表，即匹配任何一个列在方括号中的字符。比如[abc]表示要么匹配一个a，要么匹配一个b，要么匹配一个c；如果在方括号中使用短划线分隔两个字符，表示所有在这两个字符范围内的都可以匹配。比如[0-9]表示匹配所有0到9的数字，[a-z]表示匹配任意的小写字母）。
8）以叹号"**!**"表示不忽略(跟踪)匹配到的文件或目录，即要忽略指定模式以外的文件或目录，可以在模式前加上惊叹号（!）取反。需要特别注意的是：**如果文件的父目录已经被前面的规则排除掉了，那么对这个文件用"!"规则是不起作用的**。也就是说"!"开头的模式表示否定，该文件将会再次被包含，如果排除了该文件的父级目录，则使用"!"也不会再次被包含。可以使用反斜杠进行转义。

需要谨记：git对于.ignore配置文件是按行从上到下进行规则匹配的，意味着如果前面的规则匹配的范围更大，则后面的规则将不会生效；

.gitignore忽略规则简单说明

```
#               表示此为注释,将被Git忽略
*.a             表示忽略所有 .a 结尾的文件
!lib.a          表示但lib.a除外
/TODO           表示仅仅忽略项目根目录下的 TODO 文件，不包括 subdir/TODO
build/          表示忽略 build/目录下的所有文件，过滤整个build文件夹；
doc/*.txt       表示会忽略doc/notes.txt但不包括 doc/server/arch.txt

bin/:           表示忽略当前路径下的bin文件夹，该文件夹下的所有内容都会被忽略，不忽略 bin 文件
/bin:           表示忽略根目录下的bin文件
/*.c:           表示忽略cat.c，不忽略 build/cat.c
debug/*.obj:    表示忽略debug/io.obj，不忽略 debug/common/io.obj和tools/debug/io.obj
**/foo:         表示忽略/foo,a/foo,a/b/foo等
a/**/b:         表示忽略a/b, a/x/b,a/x/y/b等
!/bin/run.sh    表示不忽略bin目录下的run.sh文件
*.log:          表示忽略所有 .log 文件
config.php:     表示忽略当前路径的 config.php 文件

/mtk/           表示过滤整个文件夹
*.zip           表示过滤所有.zip文件
/mtk/do.c       表示过滤某个具体文件

被过滤掉的文件就不会出现在git仓库中（gitlab或github）了，当然本地库中还有，只是push的时候不会上传。

需要注意的是，gitignore还可以指定要将哪些文件添加到版本管理中，如下：
!*.zip
!/mtk/one.txt

唯一的区别就是规则开头多了一个感叹号，Git会将满足这类规则的文件添加到版本管理中。为什么要有两种规则
想象一个场景：假如我们只需要管理/mtk/目录中的one.txt文件，这个目录中的其他文件都不需要管理，那么.g
/mtk/*
!/mtk/one.txt

假设我们只有过滤规则，而没有添加规则，那么我们就需要把/mtk/目录下除了one.txt以外的所有文件都写出来
注意上面的/mtk/*不能写为/mtk/，否则父目录被前面的规则排除掉了，one.txt文件虽然加了!过滤规则，也不

--------------------------------------------------------------------------------
```


随笔档案

Linux加油站

```
还有一些规则如下：
fd1/*
说明：忽略目录 fd1 下的全部内容；注意，不管是根目录下的 /fd1/ 目录，还是某个子目录 /child/fd1/ 目

/fd1/*
说明：忽略根目录下的 /fd1/ 目录的全部内容；

/*
!.gitignore
!/fw/
/fw/*
!/fw/bin/
!/fw/sf/
说明：忽略全部内容，但是不忽略 .gitignore 文件、根目录下的 /fw/bin/ 和 /fw/sf/ 目录；注意要先对b
```

温馨提示：
如果你不慎在创建.gitignore文件之前就push了项目，那么即使你在.gitignore文件中写入新的过滤规则，这些规则也不会起作用，Git仍然会对所有文件进行版本管理。简单来说出现这种问题的原因就是Git已经开始管理这些文件了，所以你无法再通过过滤规则过滤它们。所以大家一定要养成在项目开始就创建.gitignore文件的习惯，否则一单push，处理起来会非常麻烦。

.gitignore忽略规则常用示例

1）示例

比如你的项目是java项目，.java文件编译后会生成.class文件，这些文件多数情况下是不想被传到仓库中的文件。这时候你可以直接适用github的.gitignore文件模板。
https://github.com/github/gitignore/blob/master/Java.gitignore 将这些忽略文件信息复制到你的.gitignore文件中去：

```
# Compiled class file
*.class

# Log file
*.log

# BlueJ files
*.ctxt

# Mobile Tools for Java (J2ME)
.mtj.tmp/

# Package Files #
*.jar
*.war
*.nar
*.ear
*.zip
*.tar.gz
*.rar

# virtual machine crash logs, see http://www.java.com/en/download/help/error_hotspot.xml
hs_err_pid*
```

可以看到github为我们提供了最流行的.gitignore文件配置。保存.ignore文件后我们查看下git status，检查下是否还有我们不需要的文件会被添加到git中去：

```
$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   .gitignore
        new file:   HelloWorld.java

Untracked files:
```

Git基础教程
Prometheus中文手册
Zabbix监控配置教程
Centos的epel源下载
Nginx官方文档
Mysql源码下载
Python自动化运维-案例源码
阿里开源镜像
Docker镜像-hub.docker
Python自动化运维之路
Docker基础学习
Redis命令参考
Django基础教程
RUNOOB.COM - 编程学习
Elasticsearch 中文社区
Swarm管理-Linux运维日志
Tengine参考文档
HTTP Status Codes
Linux监控专注
Puppet版本下载
Open-falcon社区文档
每天一个Linux命令
Python学习- 推荐网站
攻防安全指南
Squid中文权威指南
Ceph开源社区
Linux运维日记本
AWK使用手册
Docker学习汇总
Kubernetes 中文社区
Technology blog1 of learning
精通Python自动化脚本
Shell传递参数
Rancher 部署文档
Kubernetes 部署手册
ElasticSearch性能监控指标
Go语言学习速查手册
Go语言中文网
运维派
优雅姿态监控Kubernetes
K8S TLS bootstrapping
Prometheus 操作指南
Supervisor教程
Prometheus 高可用
Ansible Tower 实战
Kubernetes 中文文档
Kubernetes 学习手册
Elasticsearch 权威指南
Technology blog2 of learning
Harbor 使用手册
Kubernetes 实践指南

最新评论

1. Re:监控某个目录是否被更改
不错，赞一个
--怀岭以南
2. Re:Gitlab利用Webhook实现Push代码后的jenkins自动构建
Hook executed successfully but returned HTTP 404，查阅了许久，未曾解决之前能正常触发，配置Nginx代理，webhook URL地址变为域名，过后尝试t...
--情郎
3. Re:Redis哨兵模式（sentinel）学习总结及部署记录（主从复制、读写分离、主从切换）
您好前辈！我配置哨兵模式的过程中，主从都在运行，为什么哨兵相互之间会发现不了对方，最后以master sdown的状态结束了。
--zc0530
4. Re:Nginx+keepalived 双机热备（主从模式）
@ 看尽浮华缺少lib文件导致。使用"ldd $(which/app/nginx/sbin/nginx)"查看

```
  (use "git add <file>..." to include in what will be committed)

        Config.ini
```

比如我的项目目录下有一个Config.ini文件，这个是个本地配置文件我不希望上传到git中去，我们可以在gitignore文件中添加这样的配置：

```
Config.ini
```

或者你想忽略所有的.ini文件你可以这样写：

```
*.ini
```

如果有些文件已经被你忽略了，当你使用git add时是无法添加的，比如我忽略了*.class，现在我想把HelloWorld.class添加到git中去：

```
$ git add HelloWorld.class
The following paths are ignored by one of your .gitignore files:
HelloWorld.class
Use -f if you really want to add them.
```

git会提示我们这个文件已经被我们忽略了，需要加上-f参数才能强制添加到git中去：

```
$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   .gitignore
        new file:   HelloWorld.class
        new file:   HelloWorld.java
```

这样就能强制添加到缓存中去了。如果我们意外的将想要忽略的文件添加到缓存中去了，我们可以使用rm命令将其从中移除：

```
$ git rm HelloWorld.class --cached
rm 'HelloWorld.class'
```

如果你已经把不想上传的文件上传到了git仓库，那么你必须先从远程仓库删了它，我们可以从远程仓库直接删除然后pull代码到本地仓库这些文件就会本删除，或者从本地删除这些文件并且在.gitignore文件中添加这些你想忽略的文件，然后再push到远程仓库。

2）示例

```
下面是曾经线上使用过的一个gerrit里项目代码的.gitignore的配置（在项目中添加.gitignore过滤文件，在g
[wangshibo@gerrit-server hq_ios]$ cat .gitignore
#Built application files
*.apk
*.ap_

# Files for the Dalvik VM
*.dex

# Java class files
*.class

# Generated files
*/bin/
*/gen/
*/out/

# Gradle files
.gradle/
build/
*/build/
gradlew
gradlew.bat

# Local configuration file (sdk path, etc)
```

确认。再使用find命令找libfastcommon.so的位置，做软链接到/usr/lo...

--散尽浮华

5. Re:Nginx+keepalived 双机热备（主从模式）

大神，nginx之前启动的时候用了root权限，现在改成普通用户启动，但是会报错/app/nginx/sbin/nginx: error while loading shared libraries:...

--看尽浮华

6. Re:LVM常规操作记录梳理 [扩容、缩容、快照等]

@ z寒江雪对的...

--散尽浮华

7. Re:LVM常规操作记录梳理（扩容/缩容/快照等）

如果要使用卷组剩下的所有空间，直接使用lvcreate -n lv_name -l 100%FREE vg_name 命令参数就可以。

--z寒江雪

8. Re:Tomcat通过Redis实现session共享的完整部署记录

博主你好，能否分享下RSM for Tomcat8的源码，学习一下，感谢。

--beneshady

9. Re:linux下sendmail邮件系统安装操作记录

博主，你好，我用第一种方式发送邮件成功了，但是发件人是root@xxxx，我想把root改成其它名称，怎么样才能在不切换到其它用户的前提下，发现人显示我想设置的名称呢?

--小豹子加油

10. Re:MySQL+MGR 单主模式和多主模式的集群环境 - 部署手册 (Centos7.5)

博主，关于客户端连接MGR的方式是什么样的呢?

--JK-chen

阅读排行榜

评论排行榜

```
local.properties

# Proguard folder generated by Eclipse
proguard/

# Log Files
*.log

# Android Studio Navigation editor temp files
.navigation/

# Android Studio captures folder
captures/

# Intellij
*.iml
*/*.iml

# Keystore files
#*.jks
#gradle wrapper
gradle/

#some local files
*/.settings/
*/.DS_Store
.DS_Store
*/.idea/
.idea/
gradlew
gradlew.bat
unused.txt
```

3）示例

```
[wangshibo@gerrit-server hq_ios$ cat .gitignore
# Lines that start with '#' are comments.
# IntelliJ IDEA Project files
.idea
*.iml
*.ipr
*.iws
out

# Eclipse Project files
.classpath
.project
.settings/

bin/
gen/
local.properties

.DS_Store
Thumbs.db

*.bak
*.tem
*.temp
#.swp
*.*~
~*.*
```

.gitignor忽略规则查看

如果你发下.gitignore写得有问题，需要找出来到底哪个规则写错了，可以用git check-ignore命令检查：

```
$ git check-ignore -v HelloWorld.class
```

```
.gitignore:1:*.class    HelloWorld.class
```

可以看到HelloWorld.class匹配到了我们的第一条*.class的忽略规则所以文件被忽略了。

简单来说，要实现过滤掉Git里不想上传的文件，如上介绍三种方法能达到这种目的，只不过适用情景不一样：

```
============第一种方法============
针对单一工程排除文件，这种方式会让这个工程的所有修改者在克隆代码的同时，也能克隆到过滤规则，而不用自
这就能保证所有修改者应用的都是同一份规则，而不是张三自己有一套过滤规则，李四又使用另一套过滤规则，个

配置步骤如下：
在工程根目录下建立.gitignore文件，将要排除的文件或目录 写到.gitignore这个文件中，其中有两种写入方

a)使用命令行增加排除文件
排除以.class结尾的文件 echo "*.class" >.gitignore (>> 是在文件尾增加,> 是删除已经存在的内容再增
生成一个.gitignore的文件。排除bin目录下的文件 echo "bin/" >.gitignore

b)最方便的办法是，用记事本打开，增加需要排除的文件或目录，一行增加一个，例如：
*.class
*.apk
bin/
gen/
.settings/
proguard/


===========第二种方法===========
全局设置排除文件，这会在全局起作用，只要是Git管理的工程，在提交时都会自动排除不在控制范围内的文件或
比较省事，只要一次全局配置，不用每次建立工程都要配置一遍过滤规则。但是这不保证其他的开发者在克隆你的
的是一样的，这就带来了代码提交过程中的各种冲突问题。
配置步骤如下：
a）像方法（1）一样，也需要建立一个.gitignore文件，把要排除的文件写进去。
b）但在这里，我们不规定一定要把.gitnore文件放到某个工程下面，而是任何地方，比如我们这里放到了Git默
c）使用命令方式可以配置全局排除文件：
   # git config --global core.excludesfile ~/.gitignore
   你会发现在~/.gitconfig文件中会出现excludesfile = /home/wangshibo/hqsb_ios/.gitignore。
   说明Git把文件过滤规则应用到了Global的规则中。


===========第三种方法===========
单个工程设置排除文件，在工程目录下找到.git/info/exclude，把要排除的文件写进去：
*.class
*.apk
bin/
gen/
.settings/
proguard/

这种方法就不提倡了，只能针对单一工程配置，而且还不能将过滤规则同步到其他开发者，跟方法一和方法二比较
```

Git忽略规则(.gitignore配置）不生效原因和解决

```
第一种方法：
.gitignore中已经标明忽略的文件目录下的文件，git push的时候还会出现在push的目录中，或者用git statu
原因是因为在git忽略目录中，新建的文件在git中会有缓存，如果某些文件已经被纳入了版本管理中，就算是在.
这时候我们就应该先把本地缓存删除，然后再进行git的提交，这样就不会出现忽略的文件了。

解决方法：git清除本地缓存（改变成未track状态），然后再提交：
[root@kevin ~]# git rm -r --cached .
[root@kevin ~]# git add .
[root@kevin ~]# git commit -m 'update .gitignore'
[root@kevin ~]# git push -u origin master

需要特别注意的是：
1）.gitignore只能忽略那些原来没有被track的文件，如果某些文件已经被纳入了版本管理中，则修改.gitign
2）想要.gitignore起作用，必须要在这些文件不在暂存区中才可以，.gitignore文件只是忽略没有被staged(c
   对于已经被staged文件，加入ignore文件时一定要先从staged移除，才可以忽略。
```

```
16
17  第二种方法:（推荐）
18  在每个clone下来的仓库中手动设置不要检查特定文件的更改情况。
19  [root@kevin ~]# git update-index --assume-unchanged PATH                    //在PATH处输入要忽
```

在使用.gitignore文件后如何删除远程仓库中以前上传的此类文件而保留本地文件

在使用git和github的时候，之前没有写.gitignore文件，就上传了一些没有必要的文件，在添加了.gitignore文件后，就想删除远程仓库中的文件却想保存本地的文件。这时候**不可以直接使用"git rm directory"**，这样会删除本地仓库的文件。可以使用"**git rm -r –cached directory**"来删除缓冲，然后进行"**commit**"和"**push**"，这样会发现远程仓库中的不必要文件就被删除了，以后可以直接使用"**git add -A**"来添加修改的内容，上传的文件就会受到.gitignore文件的内容约束。

额外说明：**git库所在的文件夹中的文件大致有4种状态**

```
1   Untracked:
2   未跟踪，此文件在文件夹中，但并没有加入到git库，不参与版本控制. 通过git add 状态变为Staged.
3
4   Unmodify:
5   文件已经入库，未修改，即版本库中的文件快照内容与文件夹中完全一致. 这种类型的文件有两种去处，如果它被
6   而变为Modified. 如果使用git rm移出版本库，则成为Untracked文件
7
8   Modified:
9   文件已修改，仅仅是修改，并没有进行其他的操作. 这个文件也有两个去处，通过git add可进入暂存staged状
10  使用git checkout 则丢弃修改过，返回到unmodify状态，这个git checkout即从库中取出文件，覆盖当前修
11
12  Staged:
13  暂存状态. 执行git commit则将修改同步到库中，这时库中的文件和本地文件又变为一致，文件为Unmodify状
14  执行git reset HEAD filename取消暂存，文件状态为Modified
15
16  Git 状态 untracked 和 not staged的区别
17  1）untrack     表示是新文件，没有被add过，是为跟踪的意思。
18  2）not staged  表示add过的文件，即跟踪文件，再次修改没有add，就是没有暂存的意思
```

********** 当你发现自己的才华撑不起野心时，就请安静下来学习吧！**********

分类: 版本控制

散尽浮华
关注 - 23
粉丝 - 2494
+加关注

36 3

« 上一篇: 常用rsync命令操作梳理
» 下一篇: Git常用命令梳理

posted @ 2016-07-21 01:25 散尽浮华 阅读(415291) 评论(14) 编辑 收藏

评论

#1楼 2017-02-16 17:50 | lglong519

DS_Store
.AppleDouble
.LSOverride

类似的这几个东西有什么用?

支持(0) 反对(0)

#2楼 [楼主] 2017-02-17 18:11 | 散尽浮华

@ lglong519
抱歉，没看懂你的问意。
匹配忽略的字符而已~

支持(0) 反对(0)

#3楼 2018-09-10 17:23 | huochong

挺详细的

支持(0) 反对(0)

#4楼 2018-11-15 14:35 | .NET开发菜鸟

楼主，按照文章里面试的.gitignore不生效的原因和解决，再次status的时候，.gitignore文件里面的规则还是不起作用

支持(0) 反对(0)

#5楼 [楼主] 2018-11-15 16:57 | 散尽浮华

@ .NET开发菜鸟
试试第二种方法

支持(0) 反对(0)

#6楼 2018-12-11 19:13 | Ethan Shan

mark.

支持(0) 反对(0)

#7楼 2018-12-19 14:57 | 大限

谢谢楼主分享，#**/packages/*如果忽略或重新下载包很多很慢，#*.[Cc]ache是一个类库，如果忽略无法及时更新；所以我全都注释掉；

支持(0) 反对(0)

#8楼 2019-02-22 10:03 | Lasper

修改:
博主的说法没问题，自己的理解有偏差。
博主说的是过滤目录，而我说的是过滤目录和文件。

博主您好，非常感谢您详细的整理。
发现其中的一个问题
原文：fd1/* 说明：忽略目录 fd1 下的全部内容；注意，不管是根目录下的 /fd1/ 目录，还是某个子目录 /child/fd1/ 目录，都会被忽略；

如果过滤某个文件或者目录，而不管是在根目录还是子目录，应该写作 **/fd1 或者 fd1。

支持(1) 反对(0)

#9楼 2019-02-23 11:50 | eagle_wolf

.idea是不是写错了啊 不是应该写成/.idea/吗?
/.idea/是代表过滤整个文件夹啊

支持(0) 反对(0)

#10楼 2019-02-23 11:53 | eagle_wolf

@ Lasper
谢谢你 看你的回复我明白我的问题了

支持(0) 反对(0)

#11楼 2019-02-23 12:08 | eagle_wolf

不对 还是没明白 请问/fd1/ 和/fd1 有区别吗?

支持(0) 反对(0)

#12楼 2019-04-27 16:43 | wengweng

很详细，学习了

支持(0) 反对(0)

#13楼 2019-06-25 14:14 | iceFroooooog

好的，谢谢啦

支持(0) 反对(0)

#14楼 2019-10-15 14:16 | WaterYuan

忽略app模块build文件夹下除generated/source/apt/debug/com/apt/demo/AudoClass.java之外所有文件、文件夹，该怎么匹配?

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 登录 或 注册， 访问 网站首页。

相关博文：
· Git忽略规则.gitignore梳理-散尽浮华-博客园
· Git 忽略提交 .gitignore
· Git 忽略规则 .gitignore文件 MD
· Git忽略提交.gitignore
· git正确的删除远程仓库的文件并用.gitignore忽略提交此文件
» 更多推荐...

最新 IT 新闻:
· 梁建章：疫情全球蔓延应防一刀切隔离 警惕脱钩风险
· 我在东北当主播：父母最早坚决反对 看到年入百万后就啥都不说了
· 与程序员相关的CPU缓存知识
· 西雅图确认美国首例冠状病毒死亡病例 微软可能会取消Build 2020会议
· 百程CEO回应清算传闻：并非官方文件 没有破产也没欠钱
» 更多新闻...