# GIT

## git基本命令

### git clone

获取代码 //svn checkout

eg: 获取URL为<http://git.jd.com/cloudwaf/wafapi/tree/dev>的代码

git clone <http://git.jd.com/cloudwaf/wafapi.git>

### git status

当前路径下文件状态，svn diff

新增文件为untracked

### git add

添加内容到下一次提交中

开始跟踪新文件，或把已跟踪的文件放到暂存区

git add 文件名：开始跟踪一个文件。

git add 目录：递归跟踪该目录下所有文件

### git commit

提交，svn commit

git commit –a：自动把所有跟踪过的文件暂存一并提交，从而跳过git add 步骤

### git rm

### git log

git log –p 可以查看log的详细信息

### git pull

自动抓取并合并远程分支到当前分支

Svn update

### git push

git push：将本地库中的最新信息发送给远程库

git commi：将本地修改过的文件提交到本地库

### git branch

git branch klp：自己新建分支klp

注册表修改环境变量GOPATH：

运行-regedit

\HKEY\_USERS\S-1-5-21-1713849901-2797640346-4150151575-784095\Environment

### git reset

Add 多了，用go reset回退

git reset HEAD

### 初始化

进入远程文件夹，git --bare init –shared

Git clone

Touch 文件名

Git add

Git commit

**receive.denyCurrentBranch’ Configuration variable to ‘refuse**

git --bare init --share

## git rm filename

删除文件

git checkout 新分支报错

$ git checkout test

error: pathspec 'test' did not match any file(s) known to git.

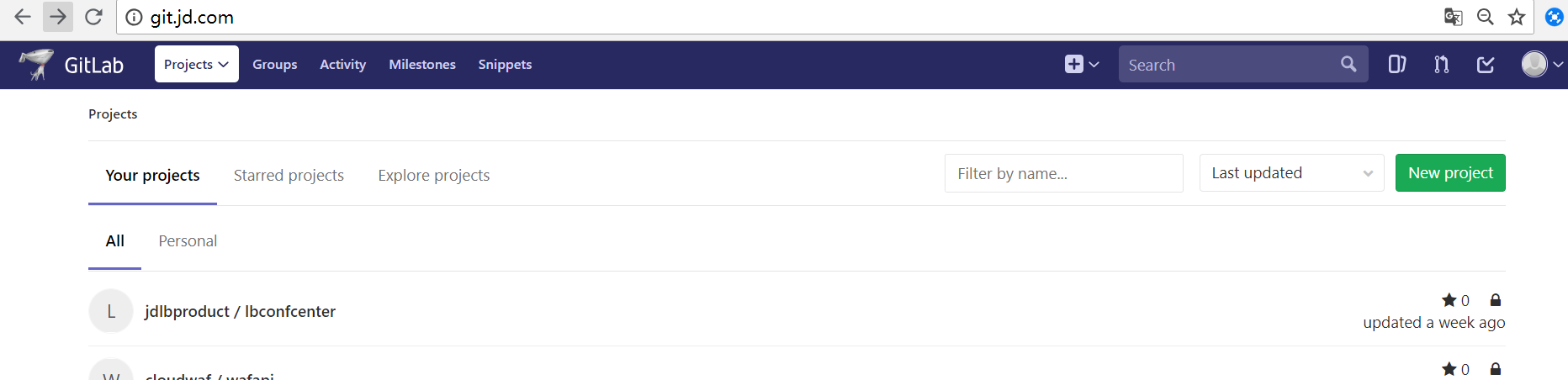
Merge：

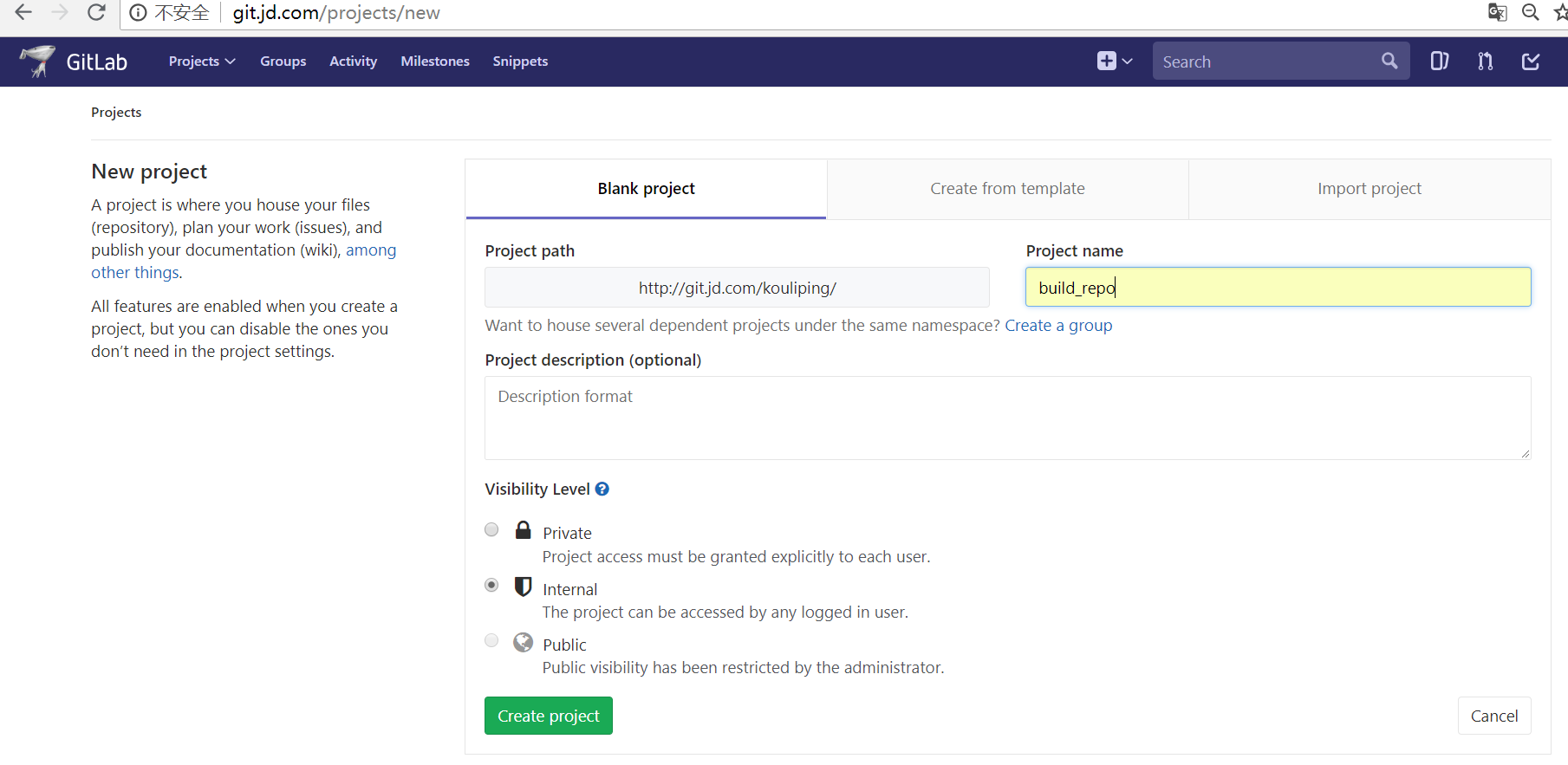
修改要merge到test分支

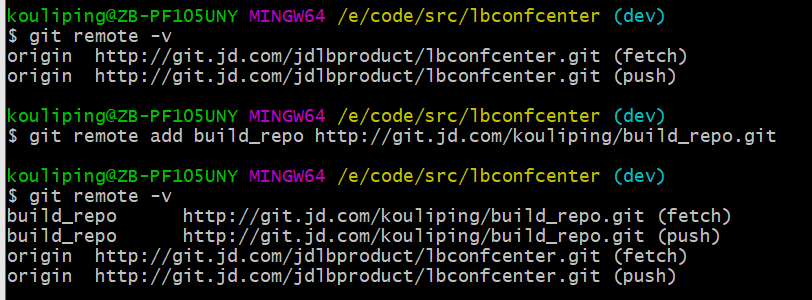
在test分支，Git merge dev，然后git push origin test

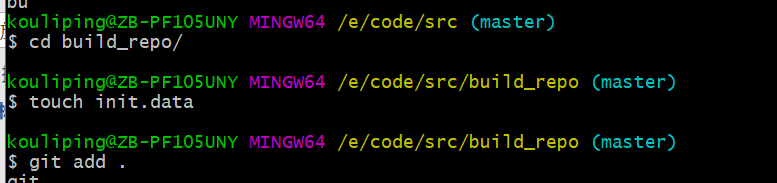
## 新建project：

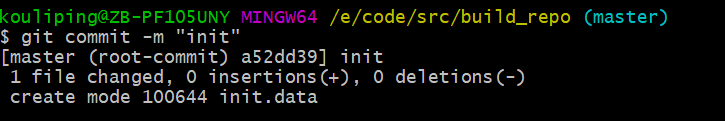
后续直接通过git push build\_repo dev push到build\_repo，然后去linux pull后编译，最后git push origin dev 来提交代码

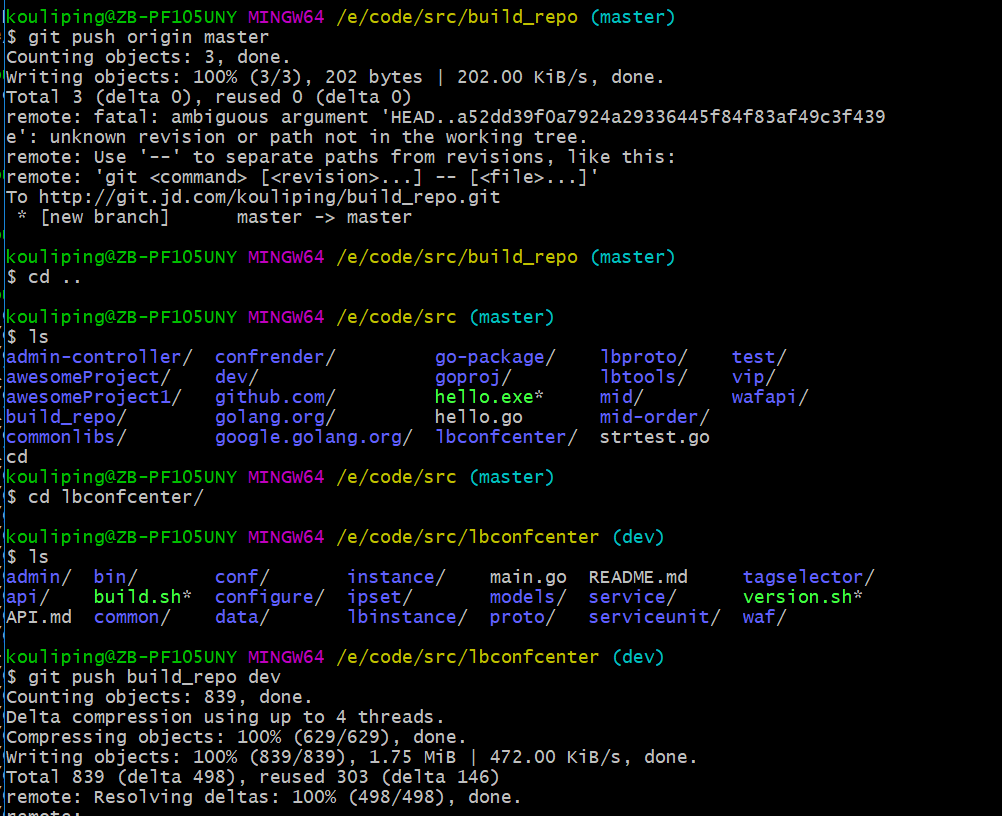












## 上一个git命令中止，无法继续git

今天写完代码git commit -m 一下发现提示

Another git process seems to be running in this repository, e.g.

an editor opened by 'git commit'. Please make sure all processes   
are terminated then try again. If it still fails, a git process   
may have crashed in this repository earlier: 

remove the file manually to continue.

大致意思就是你已经提交一了一个commit，该进程卡在这里了，所以无法继续提交。

上网百度了一下

使用$ rm -f ./.git/index.lock 命令，结束该进程。

即可继续提交。

## 放弃本地所有修改

尚未add

$ git checkout – filename

$ git checkout . 放弃所有修改

已经使用了  git add 缓存了代码。

可以使用  git reset HEAD filepathname （比如： git reset HEAD readme.md）来放弃指定文件的缓存，放弃所以的缓存可以使用 git reset HEAD . 命令。

此命令用来清除 git  对于文件修改的缓存。相当于撤销 git add 命令所在的工作。在使用本命令后，本地的修改并不会消失，而是回到了如（一）所示的状态。继续用（一）中的操作，就可以放弃本地的修改。

三，

已经用 git commit  提交了代码。

可以使用 git reset --hard HEAD^ 来回退到上一次commit的状态。此命令可以用来回退到任意版本：git reset --hard  commitid

你可以使用 git log 命令来查看git的提交历史。git log 的输出如下,之一这里可以看到第一行就是 commitid：

查看用户名

配置用户名

$ git config --global user.name "weiguobing"

$ git config --global user.email "guobingwei@aliyun.com"

git config --global --list

## pull冲突时：

git stash

强制放弃拉取远程分支

*git* *fetch* --*all*

*git* *reset* --*hard* *origin/master*

*git* *pull*

## 撤销某次commit

git log 找到要撤销的id

git reset id 撤销某次commit但不对代码的修改进行撤销

git reset -hard id 撤销某次commit并将代码回复到前一次commit的版本

注意：id为回退到的版本，如本次commit id 为id1，上一次为id2，则应该为git reset id2

## 修改commit的message

<https://blog.csdn.net/sodaslay/article/details/72948722>

git commit –amend

在打开的文件中修改message

git push origin branch

## 新建分支

git checkout –b local\_branch\_name

git push origin localbranch\_name:remote\_branch\_name

## rebase & reset

到自己的开发分支

Git rebase dev，如有conflic，解决完后git rebase –continue

查看dev的git log，得到最新的commit id

在自己的分支git reset id

然后把修改重新add commit push

http://gitbook.liuhui998.com/4\_2.html

## 分支

删除远程分支后，本地git branch -a仍存在已删除的分支

使用命令 git remote show origin，可以查看remote地址，远程分支，还有本地分支与之相对应关系等信息。此时我们可以看到那些远程仓库已经不存在的分支，根据提示，使用 git remote prune origin 命令：删除远程已不存在的分支

https://blog.csdn.net/zhaoguanghui2012/article/details/75127446

删除本地分支：

git branch –d branchname

删除远程分支：

git push origin --delete <BranchName>

https://blog.csdn.net/qq\_32452623/article/details/54340749

## git merge

将两个或以上的开发历史合并

从指定的commit合并到当前分支

1 git pull = git fetch + git merge

2 从一个分支到另一个分支的合并

git merge –abort

抛弃合并过程并尝试重建合并前的状态，但是当合并开始时如存在未commit的文件，则无法重现。

如果commit了，git merge或git pull冲突时，可以这样回退merge

强制覆盖

方法1git push origin test:master -f           //将test分支强制（-f）推送到主分支master

// 注意看清顺序，master被覆盖

方法2（假设当前位于test分支）

git checkout master                          //将当前分支切换到主分支

git reset -hard test                            //将主分支重置为test分支

git push origin master -force             //将重置后的master分支强制推送到远程仓库

参数：

https://blog.csdn.net/t3/article/details/77069523

# 代码

## 下载代码

### Git clone

git clone <http://git.jd.com/cloudwaf/mid.git>

下载下来的是主线，需要切到branch

cd wafapi

git checkout dev

即可

### Ssh方式

git clone git@git.jd.com:jdlbproduct/lbconfcenter.git

# 编译代码

cd mid

./build.sh midserver/main.go

# 上传代码

Git status 查看状态，修改的什么文件，branch分支

Git add：如果都是自己改的，可直接git add .，否则需要git add 文件名将所有文件add

Git commit –m “log”：本地代码上传到本地库

git config --global user.name "kouliping"

git config --global user.email kouliping@jd.com

git push origin klp：本地库到远程库

# lbconfcenter/models

building/src/lbconfcenter/models/admin\_m.go:153:14: undefined: common.XEngPoll

building/src/lbconfcenter/models/admin\_m.go:158:12: undefined: common.XEngPoll

building/src/lbconfcenter/models/admin\_m.go:163:12: undefined: common.XEngPoll

building/src/lbconfcenter/models/admin\_m.go:170:8: undefined: idcMeta

building/src/lbconfcenter/models/admin\_m.go:172:14: undefined: common.XEngPoll

building/src/lbconfcenter/models/admin\_m.go:173:2: too many arguments to return

building/src/lbconfcenter/models/admin\_m.go:178:14: assignment mismatch: 2 variables but 1 values

building/src/lbconfcenter/models/admin\_m.go:211:2: too many arguments to return

have ([]proto.ConfdirMeta, error)

want ()

building/src/lbconfcenter/models/admin\_m.go:256:21: cannot refer to unexported name proto.confdirMeta

building/src/lbconfcenter/models/admin\_m.go:256:21: undefined: proto.confdirMeta

building/src/lbconfcenter/models/admin\_m.go:256:21: too many errors

[kouliping@A01-R04-I17-130-06CEKFK lbconfcenter]$

# Vim

## [vim的跨文件复制粘贴](https://www.cnblogs.com/yoyo-sincerely/p/5866206.html)

1、用vim打开一个文件，例如：a.cpp

2、在普通模式下，输入：":sp"（不含引号）横向切分一个窗口，或者":vsp"纵向切分一个窗口，敲入命令后，你将看到两个窗口打开的是同一个文件

3、在普通模式下，输入：":e b.cpp"，在其中一个窗口里打开另一个文件

4、切换到含有源文件（a.cpp）的窗口，在普通模式下，把光标移到你需要复制内容的起始行，然后输入你想复制的行的数量（从光标所在行往下计算），在行数后面接着输入yy，这样就将内容复制到临时寄存器里 了（在 普通模式下ctrl+w，再按一下w，可以在两个窗口之间切换）

5、切换到目标文件( b.cpp )窗口，把光标移到你接收复制内容的起始行，按一下p，就完成复制了。

## 多行复制黏贴的方法有：

1. 光标移到复制的第一行，例如要复制5行：5yy，光标移到要黏贴的位置，输入p或者P（p是在所在位置的后方黏贴，P是在所在位置前方黏贴）

这个方法适合要复制的行数较少情况。

2. 在normal模式下输入：开始行数，结束行数 co(py) 要黏贴的行数

例如要将第5到10行黏贴到第15行后面可以这么写：5,10 copy 15 这个方法适合有大量的行数情况。

3. shift+v进入可视模式，选中要拷贝的行，输入y复制，光标移到奥黏贴的位置，输入p或者P黏贴。这个方法最直观。

## 代码多行注释和删除

1. 光标移到要注释的行首，ctrl+v进入可视块模式，移动上下键选中要注释的行，输入I（大写i），输入要注释的字符（例如python是#），输入两个esc回到normal模式

2. 删除过程一样：先ctrl+v，选中要删除注释的行，输入x即可

3. 如果要在很多行后面加相同的东西也可以用这个方法：ctrl+v，选中要添加的行，输入A，输入要添加的字符，输入两个esc回到normal模式

在可视模式(v),可视行模式(shift+v),可视块模式中，选中行后，可以按d（或者D，删除所选中部分所在的行）删除说选中的行，按c（或者C，删除所选中部分所在的行并进入插入模式）删除所选中的行并进入插入模式。

ESC :%s/1/2/g // 把全文的1替换成2 /末尾加g 是全局替换，不加g 只替换每行第一个

## 命令：

% 这是匹配{}，[]，() 用的，例如您的游标现在在{ 上

只要按%，就会跑到相匹配的} 上。写程式时满好用的。

A 在行尾開始輸入文字。

I 再第一个非空白字符前开始输入文字，若要在第一个地方输入，使用0i

o(小写)：从光标下一行插入

O(大写)：从光标上一行插入

删除：

dd：删除一整行

dw：向后删除一个字

dG：删除至档尾d1G(数字)：删除至档首

^ 移至第一個非空白字元，注意，要 Shift 鍵。

$ 移至行尾，或 End 鍵。要 Shift 鍵。

D 刪至行尾，或 d$（含游標所在處字元）。

d0 刪至行首，或 d^（不含游標所在處字元）。

撤销、恢复

u 這個太重要了，就是 undo，傳統的 vi 僅支援一次 undo，vim

及 elvis 就不只了，vim 是沒有限制的。

U 在游標沒離開本行之前，回復所有編輯動作。

Crtl+r 這個也是很重要，就是 redo 鍵。

# Mysql

## 登陆数据库

mysql -h172.28.13.159 -P3306 -uroot -p1qaz@WSX

use jd\_lb\_db\_test2

show tables // 显示当前数据库都有哪些表

create创建表

drop table table-name删除表

## 授权

 grant all PRIVILEGES on db\_name.\* to 'username'@'xxx.xxx.xx.x' identified by 'password' WITH GRANT OPTION;

上面的语句表示将数据库 db\_name 的所有权限授权给 username 这个用户，允许 username 用户在 xxx.xxx.xx.x 这个 IP 进行远程登陆，并设置 username 用户的密码为 password。

分析参数：

all PRIVILEGES 表示赋予所有的权限给指定用户，这里也可以替换为赋予某一具体的权限，例如：[select](http://www.cnblogs.com/tags.php/select/),insert,update,delete,create,drop 等，具体权限间用“,”半角逗号分隔。

db\_name.\* 表示上面的权限是针对于哪个表的，db\_name指的是数据库名称，后面的 \* 表示对于所有的表，由此可以推理出：对于全部数据库的全部表授权为“\*.\*”，对于某一数据库的全部表授权为“数据库名.\*”，对于某一数据库的某一表授权为“数据库名.表名”。

username表示你要给哪个用户授权，这个用户可以是存在的用户，也可以是不存在的用户。

xxx.xxx.xx.x 表示允许远程连接的 IP 地址，你的IP，如果想不限制链接的 IP 则设置为“%”即可。

password 为用户username的密码。

最后执行了上面的语句后，一般都会立即生效，返回值如下：

Query OK, 0 rows affected (0.01 sec)

如果没有上面的语句那么请执行下面的命令，即可立即生效。

Mysql> flush privileges

[**好文要顶**](javascript:void(0);) [**关注我**](javascript:void(0);) [**收藏该文**](javascript:void(0);) **[https://common.cnblogs.com/images/icon_weibo_24.png](javascript:void(0);)** **[https://common.cnblogs.com/images/wechat.png](javascript:void(0);)**

## 数据

1. 更新记录：

Update 表名 set 字段=\*\* where 条件;

1. 插入记录

insert into ssl\_meta (`ssl`,not\_before,not\_after) value("klp1","2006-09-22","2009-03-12");

// 以分号结尾

1. 显示数据

select \*from table name where 筛选条件

1. 插入数据

insert into ssl\_meta (`ssl`,not\_before,not\_after) value("klp1","2006-09-22","2009-03-12");

insert into命令用于向表中插入数据。

insert into命令格式：insert into <表名> [(<字段名1>[,..<字段名n > ])] values ( 值1 )[, ( 值n )];

注：ssl是关键字，需要加反引号

复制

https://www.cnblogs.com/lxboy2009/p/7234535.html

## 表结构

删除某field

MySQL [jd\_lb\_db\_test2]> alter table vip\_meta drop column utime, drop route\_cluster;

Query OK, 0 rows affected (0.18 sec)

Records: 0 Duplicates: 0 Warnings: 0

显示表结构

MySQL [jd\_lb\_db\_test2]> desc vip\_meta;

增加某field

MySQL [jd\_lb\_db\_test2]> alter table vip\_meta add conf\_dir varchar(20) NOT NULL ;

修改fieldming

alter table test rename test1; --修改表名

alter table test add column name varchar(10); --添加表列

alter table test drop column name; --删除表列

alter table test modify address char(10) --修改表列类型

||alter table test change address address char(40)

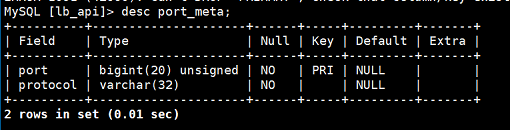
alter table test change column address address1 varchar(30)--修改表列名

https://www.cnblogs.com/jiangxiaobo/p/6110679.html

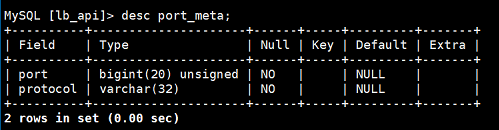
修改键属性：

alter table port\_meta drop key port\_UNIQUE;

修改前：



修改后：



Show create table 表名

修改表名

RENAME TABLE old\_table\_name TO new\_table\_name;

## Mysql备份

mysqldump -hm2836m.mysql.jddb.jcloud.com -P3358 -uwaf\_mid\_rw -pJdnWRQYEq2OqeVAFJAtx waf\_mid eslog\_download\_meta --skip-lock-tables > eslogbak.sql;

mysql -hm2836m.mysql.jddb.jcloud.com -P3358 -uwaf\_mid\_rw -pJdnWRQYEq2OqeVAFJAtx waf\_mid < eslogbak.sql

文件里是drop create insert语句

加条件的导出

--where='id < 10'

只导出表结构不导出数据：

mysqldump　-d　-A　--add-drop-table　-uroot　-p　>xxx.sql

只导出数据不导出结构

mysqldump　-d　-A　--add-drop-table　-uroot　-p　>xxx.sql

## Excel导入到mysql

load data local infile "/export/home/kouliping/files/region.csv" into table `dn\_region` FIELDS TERMINATED BY ',' LINES TERMINATED BY '\r\n';

文件里要有id

## Mysql切换调整字段顺序，不改变索引哦，完美解决方案！

mysql> alter table student modify id int(10) unsigned auto\_increment first;

mysql> alter table student modify name varchar(10) after id;

alter table qq\_admin add unique key(admin\_name);

## 拷贝数据库

跨服务器：

服务器A,B都安装了mysql，然后需要把服务器A上的数据库DB1都拷贝到服务器B上。

1st：在服务器A上操作

mysqldump -u 用户名 -p 密码 数据库名 [表名] > 导出的文件名

// 不输入表名参数则把该数据库下的所有表都拷贝

2nd: 在服务器B上，scp 把导出的文件拷贝过来

3rd： 服务器B上把数据导入数据库

mysql –u 用户名 p 密码

mysql>use 数据库

mysql> source temp.sql;// 文件名

## 字符集

show variables like '%char%';

php不加set names `charname`时所使用的编码

show global variables like '%char%';

字符集：character set，不同字符集规定了不同的字符的编码方式，一个character set 是一组符号和编码eg：ascii 编码方式是7bit表示一个字符，非英文不能使用ascii编码。Utf8对世界上所有的语言文字做了编码，

字符集级别：

服务器

数据库

数据表

表列

自下向上继承

最终使用字符集的是存储字符的列，服务器级 数据库级 数据表级 的字符集都是列的字符集的默认选项

1. 在配置文件my.cnf中 [client] 增加 default-character-set =utf8 ,会立即对**本机上**的新创建连接生效  
     
   2.在配置文件my.cnf中 [mysqld] 增加 default-character-set =utf8 ，待mysqld服务重新启动后生效  
     
   3. 执行SET语句修改字符集，对非本机新创建的连接也会生效  
     
   SET GLOBAL character\_set\_clinet=utf8;  
   SET GLOBAL character\_set\_connection=utf8;  
   SET GLOBAL character\_set\_database=utf8;  
   SET GLOBAL character\_set\_results=utf8;  
   SET GLOBAL character\_set\_server=utf8;  
     
   4.对于之前的连接线程，则没有办法，除非他们自己设置为utf8 或者等待其断开重新连接

## 修改databases名称

<https://blog.csdn.net/tjcwt2011/article/details/79451764>

## 删除database

 drop database databasename

## 设置字符集

set character\_set\_server=utf8;

set character\_set\_database=utf8;

show variables like '%char%';

# linux常用命令

## 压缩解压缩：

https://blog.csdn.net/freeking101/article/details/51480295

解压：

1. tar –xvf file.tar *// 解压 tar包*
2. tar -zxvf file.tar.gz *// 解压tar.gz*
3. tar -jxvf file.tar.bz2 *// 解压 tar.bz2*
4. tar –Zxvf file.tar.Z *// 解压tar.Z*

压缩：

1. tar –cvf jpg.tar \*.jpg *// 将目录里所有jpg文件打包成tar.jpg*
2. tar –czf jpg.tar.gz \*.jpg *// 将目录里所有jpg文件打包成jpg.tar后，并且将其用gzip压缩，生成一个gzip压缩过的包，命名为jpg.tar.gz*
3. tar –cjf jpg.tar.bz2 \*.jpg *// 将目录里所有jpg文件打包成jpg.tar后，并且将其用bzip2压缩，生成一个bzip2压缩过的包，命名为jpg.tar.bz2*
4. tar –cZf jpg.tar.Z \*.jpg *// 将目录里所有jpg文件打包成jpg.tar后，并且将其用compress压缩，生成一个umcompress压缩过的包，命名为jpg.tar.Z*

tgz文件：

tar zxvf nginx-11.8.tgz 解压

tar czvf nginx-11.8.tgz nginx-11.8 压缩

tar参数

c 新建打包文件，cv一起使用查看过程中打包文件名

x 解压文件，-C 指定解压到的文件目录

f 指定要处理的文件

j bzip2方式压缩或解压，以.tar.br2为后缀

z gzip方式，.tar.gz后缀

v 压缩或解压过程，显示

t 查看打包文件中的内容

u 更新打包文件中的内容

p 保留绝对路径

P 保留数据原来的权限和属性

1、把/home目录下面的mydata目录压缩为mydata.zip  
zip -r mydata.zip mydata #压缩mydata目录  
2、把/home目录下面的mydata.zip解压到mydatabak目录里面  
unzip mydata.zip -d mydatabak  
3、把/home目录下面的abc文件夹和123.txt压缩成为abc123.zip  
zip -r abc123.zip abc 123.txt  
4、把/home目录下面的wwwroot.zip直接解压到/home目录里面  
unzip wwwroot.zip  
5、把/home目录下面的abc12.zip、abc23.zip、abc34.zip同时解压到/home目录里面  
unzip abc\\*.zip  
6、查看把/home目录下面的wwwroot.zip里面的内容  
unzip -v wwwroot.zip  
7、验证/home目录下面的wwwroot.zip是否完整  
unzip -t wwwroot.zip  
8、把/home目录下面wwwroot.zip里面的所有文件解压到第一级目录  
unzip -j wwwroot.zip

https://www.cnblogs.com/wangkongming/p/4305962.html

zip -q -r html.zip /home/html

[kouliping@A01-R04-I17-130-06CEKFK bin]$ sed -i 's/^M//g' ./supervise.logs-manager

^M输入的方法是ctrl+v然后m

[kouliping@A01-R04-I17-130-06CEKFK bin]$ cat -A supervise.logs-manager

## 查找文件

find / -name *filename*

locate *filename*

## 查看进程总数

ps -ef |grep ping |wc -l

# protoc

## 安装下载

下载源码：

<https://github.com/google/protobuf/releases/tag/v3.6.1>

解压后得到 bin、include、readme

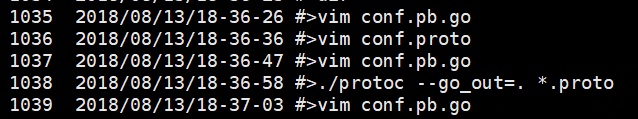


go get github.com/golang/protobuf/proto

得到bin和src，拷贝到linux中安装protoc



修改gobin为src所在路径



## 生成配置文件：

./protoc --go\_out=. \*.proto

# 正则表达式

## 元字符

. : 匹配除换行符以外的任意字符

\w : 匹配字母、数字、下划线、汉字

\s : 匹配任意的空白符

\d : 匹配数字

\b : 匹配单词的开始或结束, 由空格、标点符号、换行来分隔

^ : 匹配字符串的开始

$ : 匹配字符串的结束

\b ^ $ 只匹配一个位置，如果选中了处理多行，则^ $ 匹配行的开始和结束

|  |  |  |  |
| --- | --- | --- | --- |
|  | 12345 | 12345,12345 | 12345 12345 |
| ^\d{5,12}$ | T | F | F |
| \b\d{5,12}\b | T | T | T |

## 字符转义

\ : 转义符，\\ 查找\

## 重复

\* : 重复0次或更多

+ : 重复1次或更多

? : 重复0次或1次

{n} : 重复n次

{n,} : 重复n次或更多

{n,m} : 重复n到m次

## 字符串

[]:

[aeiou]: 任一个aeiou

[0-9]: 即\d, 任一个数字

[a-z0-9A-Z] : 英文字母或数字

\(?0\d{2}[) -]?\d{8}

一个或0个( 0 2个数字 1个或0个)- 8个数字

## 分支条件

有多个规则，满足任一个则认为匹配，用 | 把不同规则分开

分支条件时，从左到右测试，如有满足，不再继续测试后续分支条件

## 分组

使用小括号指定子表达式，即分组。

(\d{1,3}\.){3}\d{1,3} 1~3个数组+. 重复三次，1~3个数字

Eg：匹配ipv4

((2[0-4]\d|25[0-5]|[01]?\d\d?)\.){3}(2[0-4]\d|25[0-5]|[01]?\d\d?)。

## 反义

\W : 匹配任意非字母、数组、下划线、汉字的字符

\S : 任意非空白的字符

\D : 任意非数字的字符

\B : 非单词开始或结束的位置

[^x] : 匹配除了x意外的任意字符

[^aeiou] : 匹配非aeiou的任意字符

\S+ : 不包含空白符的字符串

<a[^>]\*> : 匹配用尖括号括起来的以a开头的字符串

## 后向引用

小括号()指定的子表达式后，匹配该子表达式的文本可以进一步使用，子表达式会自动拥有一个组号，自左向右默认组号为1 2 3·············……

\b(\w+)\b\s+\1\b : 匹配重复的单词

这个表达式首先是一个单词，也就是单词开始处和结束处之间的多于一个的字母或数字(\b(\w+)\b)，这个单词会被捕获到编号为1的分组中，然后是1个或几个空白符(\s+)，最后是分组1中捕获的内容（也就是前面匹配的那个单词）(\1)。

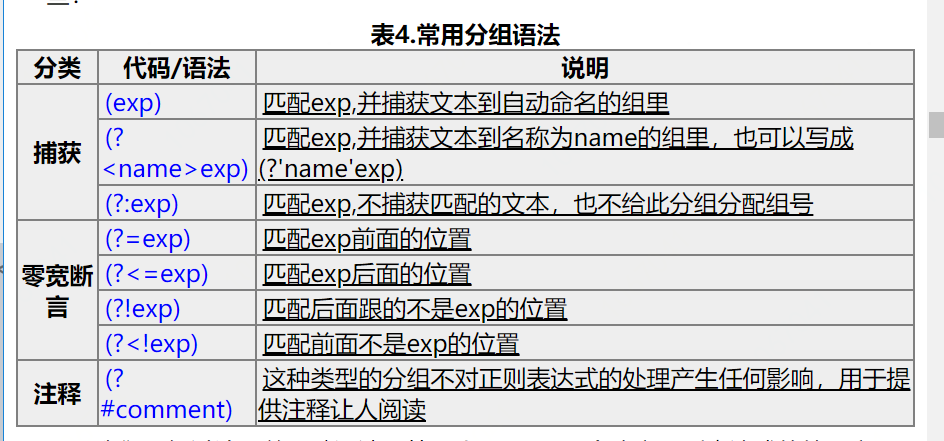
* 分组0对应整个正则表达式
* 实际上组号分配过程是要从左向右扫描两遍的：第一遍只给未命名组分配，第二遍只给命名组分配－－因此所有命名组的组号都大于未命名的组号
* 你可以使用(?:exp)这样的语法来剥夺一个分组对组号分配的参与权．

指定组名：

(?<word>\w+) 或 (?’word’\w+) ，使用： \k<word>

\b(?<Word>\w+)\b\s+\k<Word>\b

小括号用法：



For more information

<https://www.jb51.net/tools/zhengze.html>

# openapi

1st：修改package.ini，只剩下waf

2nd：./pre-gen.sh

3rd：

java -jar openapi-codegen-1.0-jar-with-dependencies.jar -i "/export/home/kouliping/code/jcloud-open-api/package\_tmp" -o "/export/home/kouliping/code/git.jd.com/jcloud-api-gateway/jcloud-sdk-go/services" -l go -a

# 查看磁盘空间

cannot create temp file for here-document: No space left on device

df –sh /\* // 逐级查看占用

环境变量

打开cmd，set PATH=%PATH%;\*\*