# Todo

搭建模拟GXPack输入输出环境

网页(浏览器)能下下来，HttpWebRequest下载不下来

## 2018.4.16

### GameBox Hellowworld参考

### 查看其他人提交的接口

### 了解GXPack代码

### VS测试样本

public void TestMethod1()

{

Console.WriteLine("===============================TestMethod1()==============================");

int a = 1, b = 3, expect = 3;

int real = a + b;

Assert.**AreEqual**(real, expect);

}

## submodule:

# \*\*\*\*\*解决问题\*\*\*\*\*

1官方文档

2 google

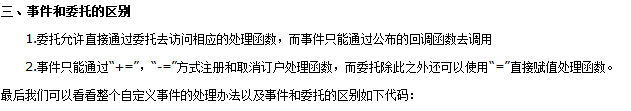
3寻找类似的开源代码、框架、文档 （比如laravel、assetbundlegraph&类红警项目、ai&lua书籍） （github上找排名靠前的项目）

4

# c#

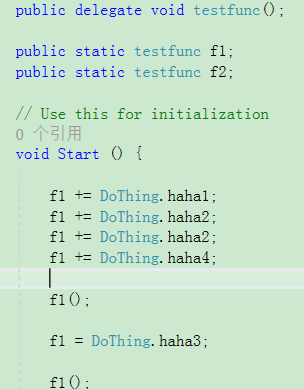
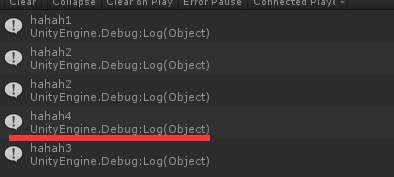
## 事件和委托

<https://www.cnblogs.com/chengxingliang/archive/2013/05/21/3051912.html>



## delegate可以直接用+= 或=初始化

+= 可以赋值多个函数 =只能赋值一个函数

# golang

set GOPATH=F:\project\zeus;F:\project\server

go install Login

go install Center

go install IDIPServer

go install Lobby

go install Match

go install Room

go install Gateway

Lobby.exe.infolog.2017.11.14.23

## net/http

### 处理浏览器的输入

输入：<http://localhost:9090/?url_long=333&url_long=444&wahaha=1234&p4=%2212345%22>

go接收到的：map[url\_long:[333 444] wahaha:[1234] p4:["12345"]]

r.Form

字典

for k, v := range r.Form {

fmt.Println("key:", k)

fmt.Println("val:", strings.Join(v, ""))

}

r.URL.Path

/favicon.ico

r.URL.Scheme

scheme

获取参数

r.Form[“zoneID”]

r.FormValue("url\_long")

### 给浏览器的返回

返回json: 



转跳：

http.Redirect(w, r, "/login/index", http.StatusFound)

## HttpServer

<https://yq.aliyun.com/articles/46439>

## 控制台日志：



log.Println("loginHandler")

# Js



Vue.js

Vue是一套用于构建用户界面的**渐进式框架**

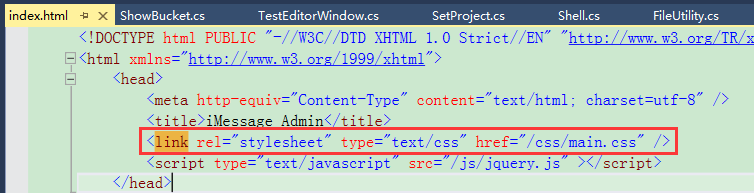
# Css

Sass

Stylelint

LESS

网页引用css:



# Mysql

<http://stading.mysql.gxpack.io/>

gxpack

ztgame@gxpack

## 安装



We've installed your MySQL database without a root password. To secure it run:

mysql\_secure\_installation

MySQL is configured to only allow connections from localhost by default

To connect run:

mysql -uroot

To have launchd start mysql now and restart at login:

brew services start mysql

Or, if you don't want/need a background service you can just run:

mysql.server start

NAVH-WK6A-DMVK-DKW3



## 用户认证

Windows版sql：启用5.x的验证方式

root:ztgame@123

laoguai:ztgame@123

## 客户端软件

EMSSQL ManagerforMySQL

Navicat

phpMyAdmin

MySQL-Front

SQLyog

copy /Y "$(TargetDir)$(TargetName).dll" F:\@@@@@@@@@@@\GameBox\TestNewBucket\Assets\GameBox\Editor\GameBox.Bucket

## 命令行

**mysql.server start**

mysql -u root –p

SELECT User,Host FROM mysql.user;

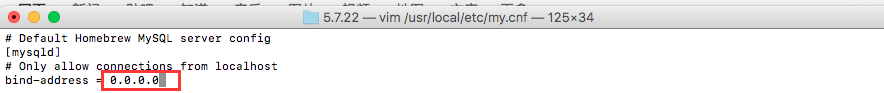
## 允许root远程连mysql

update user set host='%' where user = 'root';

select host from user where user = 'root';

flush privileges;

grant all privileges on \*.\* to root@'%' identified by 'ztgame@123';



# Git

git.exe push --progress "origin" 12-gxpack:12-gxpack

git.exe pull -v --progress "origin"

## submodule:

git.exe submodule add -- "http://gitlab.ztgame.com/tech/Client/GameBox.ThirdLib/Releases/SemVer.git" "lib/test1"

# GameBox

NodeJS

NodeJS.dll

## DLL



# GameBox.Bucket

报错的时候是不会动态加载的，只有重启Unity才起作用



copy /Y "$(TargetDir)$(TargetName).dll" F:\@@@@@@@@@@@\Saturn\code\client\Assets\GameBox\Vendor\GameBox\Bucket\net35\Editor

copy /Y "$(TargetDir)$(TargetName).pdb" F:\@@@@@@@@@@@\Saturn\code\client\Assets\GameBox\Vendor\GameBox\Bucket\net35\Editor

## todo

发现有任何dll更新后重新解析，bucket代码都会重新再执行一次

## 发布

Token

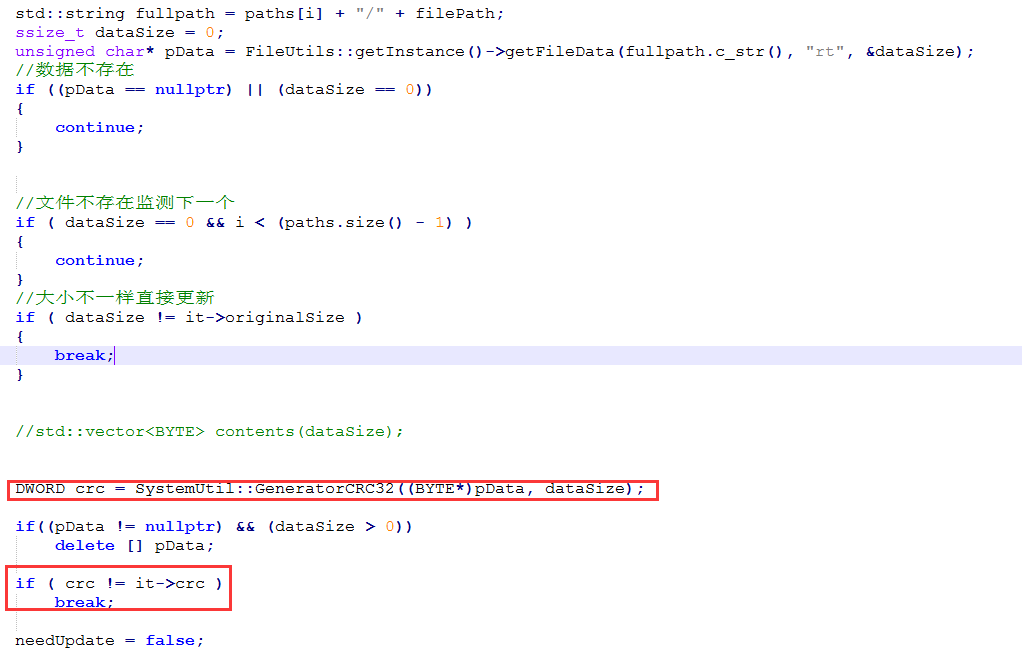
ID

UnityPackage

Host文件

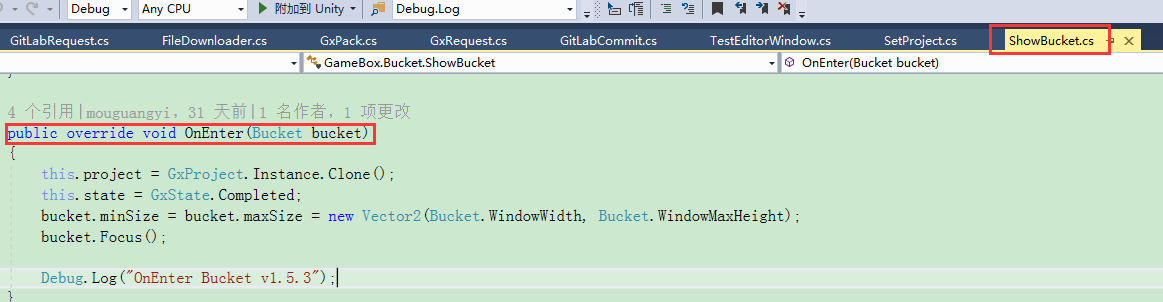
打印192.168.150.221/Gaint

## 遍历更新文件列表计算校验码看要不要更新



## 自更新

从OnEnter处会重新执行



## 获取gitlab指定tag下的所有文件

<http://gitlab.ztgame.com/api/v4/projects/tech%2FClient%2FGameBox%2FReleases%2FGXPack.TestProject.Dep/repository/tree?ref_name=v1.6.2&recursive=true> (**ref\_name XXX**)

[http://gitlab.ztgame.com/api/v4/projects/tech%2FClient%2FGameBox%2FReleases%2FGXPack.TestProject.Dep/repository/tree?**ref**=v1.5.2&recursive=true](http://gitlab.ztgame.com/api/v4/projects/tech%2FClient%2FGameBox%2FReleases%2FGXPack.TestProject.Dep/repository/tree?ref=v1.5.2&recursive=true) （用这个，不要用上面那个）

bucket.ShowNotification(new GUIContent("XXXYYYZZZ"));

private const string BucketJson = @"

{

'status': 0,

'results': {

'name': 'GameBox.Bucket',

'version': '1.0.0',

'url': 'http://gitlab.ztgame.com/tech/Client/GameBox.ThirdLib/Releases/SemVer.git',

'homepage': 'www.google.com',

'files': {

'README.md' : 'GameBox/Editor/test/README.md',

'net35/SemVer.dll' : 'GameBox/Editor/test/SemVer.dll',

},

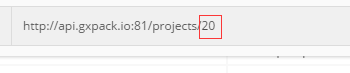
'require': {

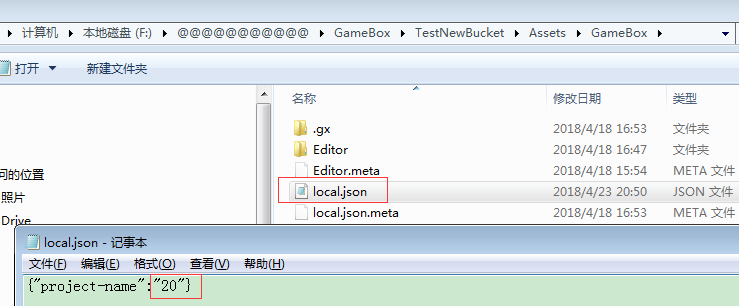
}

}

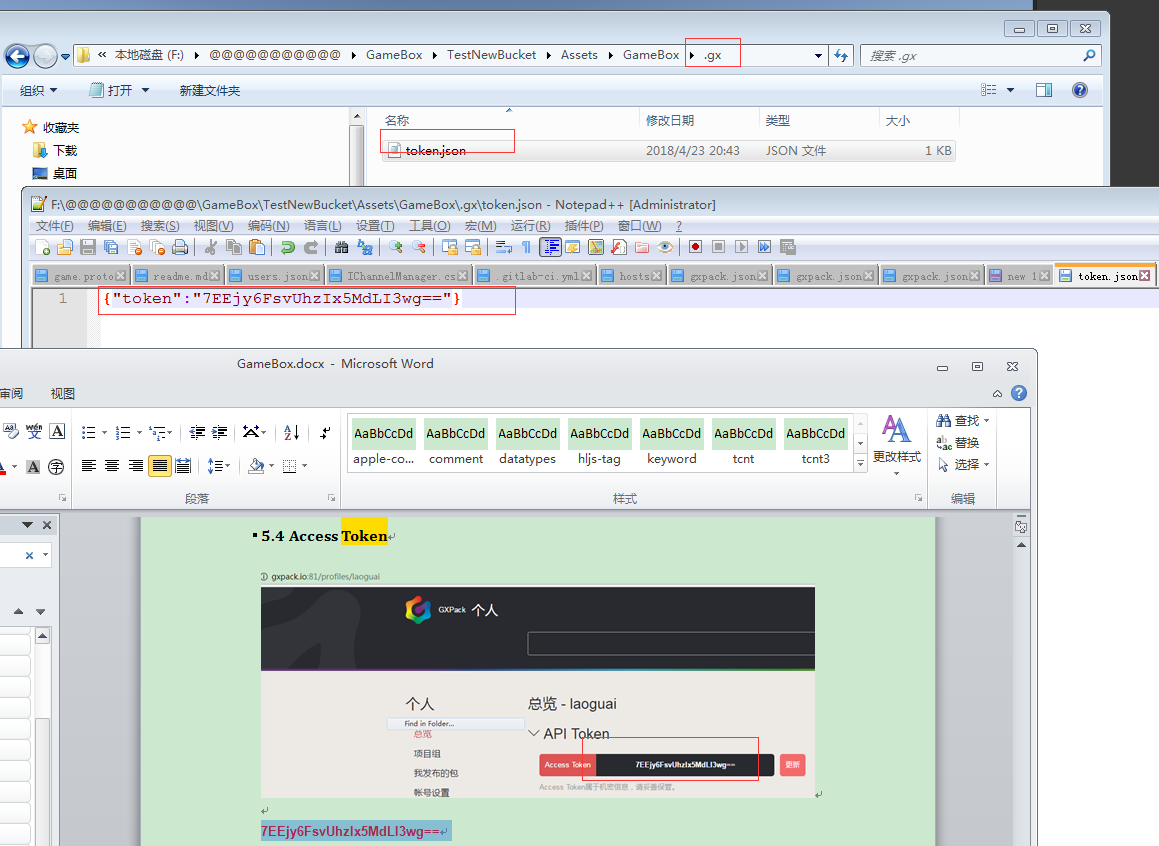
}

## 项目名 （用的id..）





## 配置token:



## seq

Host = <http://api.gxpack.io>

GetPack = "/packages/";

BucketModuleName = "GameBox.Bucket"

<http://api.gxpack.io/packages/GameBox.Bucket>

## gitsubmodule add

## 第三方库

SemVer

Newtonsoft.Json

VS工具-》扩展和更新—》所属NuGet下载安装 (2017好像自带不用下载安装)

VS工具—》NuGet控制台

PM> Install-Package Newtonsoft.Json

git.exe submodule update –init

copy /Y "$(TargetDir)$(TargetName).module" F:\@@@@@@@@@@@\GameBox\TestBucket\Assets\Editor

copy /Y "$(TargetDir)$(TargetName).pdb" F:\@@@@@@@@@@@\GameBox\TestBucket\Assets\Editor

copy /Y "$(TargetDir)$(TargetName).dll" F:\@@@@@@@@@@@\GameBox\TestBucket\Assets\Editor

copy /Y "$(TargetDir)$(TargetName).pdb" F:\@@@@@@@@@@@\GameBox\TestBucket\Assets\Editor

copy /Y "$(TargetDir)$(TargetName).dll" F:\@@@@@@@@@@@\GameBox\TestBucket\Assets\Editor

copy /Y "$(TargetDir)$(TargetName).pdb" F:\@@@@@@@@@@@\GameBox\TestBucket\Assets\Editor

copy /Y "$(TargetDir)$(TargetName).module" F:\@@@@@@@@@@@\GameBox\TestNewBucket\Assets\GameBox\Editor\GameBox.Bucket

copy /Y "$(TargetDir)$(TargetName).pdb" F:\@@@@@@@@@@@\GameBox\TestNewBucket\Assets\GameBox\Editor\GameBox.Bucket

## URL

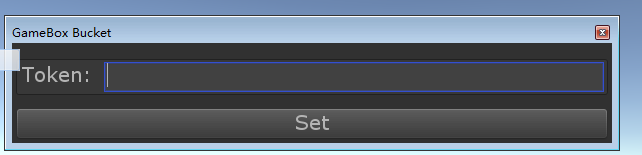
api = <http://api.gxpack.io/packages/GameBox.Bucket/1.1.0>

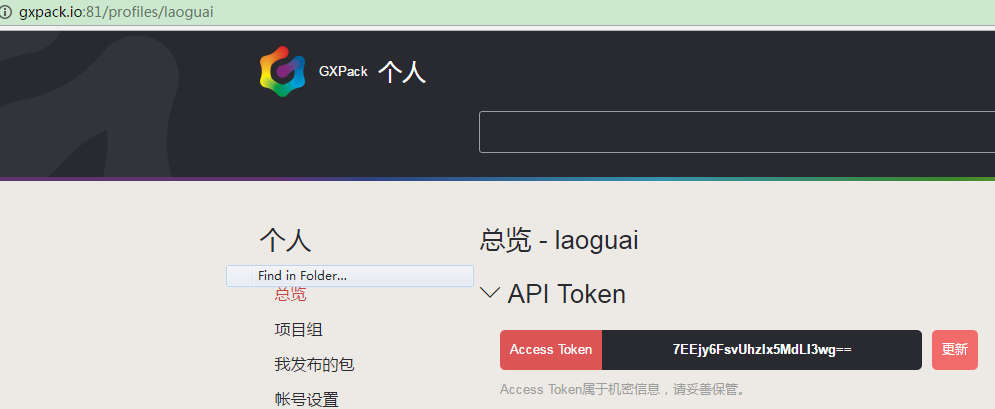
api = "http://api.gxpack.io/packages/Newtonsoft.Json/>=1.0.0"

result = "{\r\n \"status\": 0,\r\n \"results\": {\r\n \"name\": \"GameBox.Bucket\",\r\n \"version\": \"1.1.0\",\r\n \"url\": \"http://192.168.183.174:81/GameBoxAssets/Test.git\",\r\n \"homepage\": \"www.google.com\",\r\n \"files\": {\r\n \"impl...

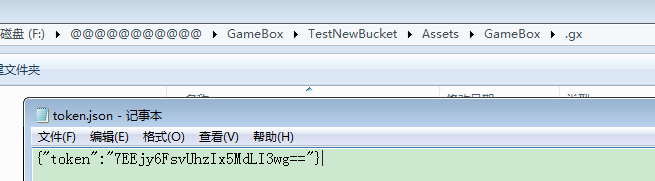
### http://api.gxpack.io/packages/GameBox.Bucket/list

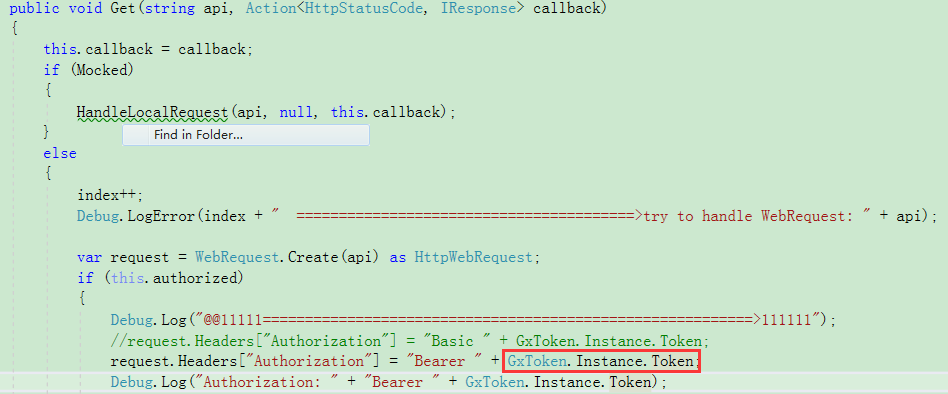
### token



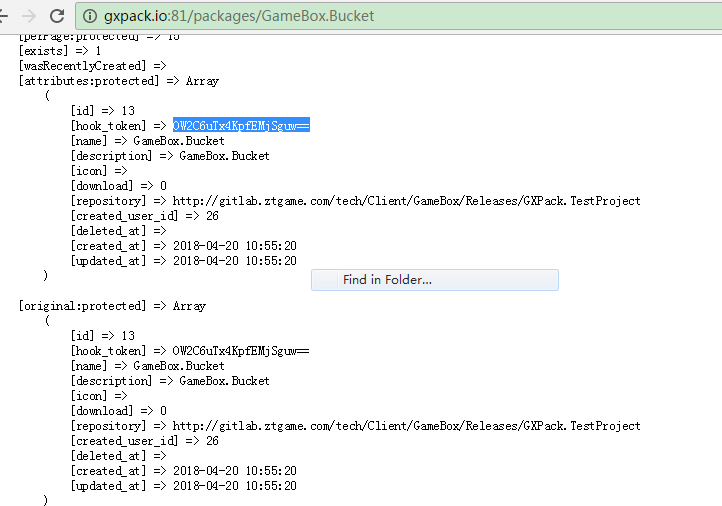


**7EEjy6FsvUhzIx5MdLI3wg==**





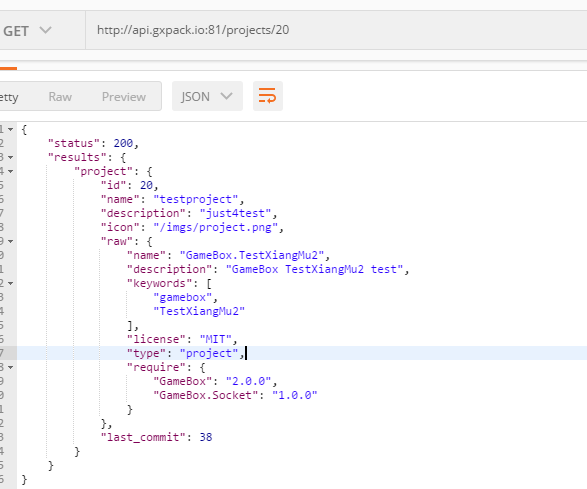
\*\*\*\*\*别跟HookToken混淆了，HookToken直接在gitlab上配置，更新的时候用来调用gxpack上的脚本：



### Project

**get**

<http://api.gxpack.io:81/projects/20>



**Post:**



### Profile

[**http://api.gxpack.io:81/profiles**](http://api.gxpack.io:81/profiles)



### Recommend

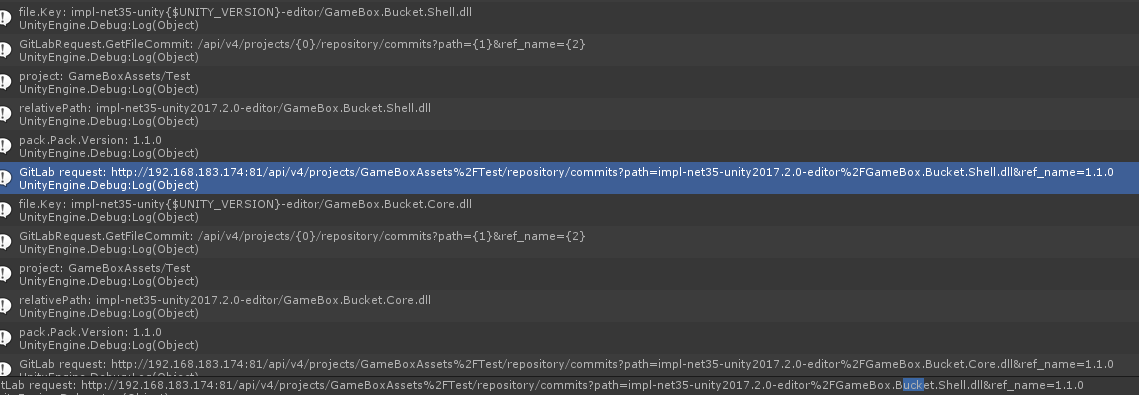
http://api.gxpack.io:81/recommend

## GitLab下载

Api全路径 = <http://192.168.183.174:81/api/v4/projects/GameBoxAssets%2FTest/repository/commits?path=impl-net35-unity2017.2.0-editor%2FGameBox.Bucket.Shell.dll&ref_name=v1.1.0>

Host = "http://192.168.183.174:81";

GetFileCommit = "/api/v4/projects/{0}/repository/commits?path={1}&ref\_name={2}";



令牌：

qzBx3gzzjXRRkcfoddsh

<http://gitlab.ztgame.com/api/v4/projects/12/repository/files?private_token=qzBx3gzzjXRRkcfoddsh&file_path=mydir/myfile&ref=master>

<http://gitlab.ztgame.com/api/v4/users?private_token=qzBx3gzzjXRRkcfoddsh>

[http:// gitlab.ztgame.com:tech/Client/GameBox.ThirdLib/SemVer](http:// gitlab.ztgame.com:tech/Client/GameBox.ThirdLib/SemVer/api/v4/users?private_token=qzBx3gzzjXRRkcfoddsh)

[/api/v4/users?private\_token=qzBx3gzzjXRRkcfoddsh](http:// gitlab.ztgame.com:tech/Client/GameBox.ThirdLib/SemVer/api/v4/users?private_token=qzBx3gzzjXRRkcfoddsh)

**返回所有的项目信息：**

http://gitlab.ztgame.com/api/v4/projects?private\_token=qzBx3gzzjXRRkcfoddsh

<http://gitlab.ztgame.com/api/v4/projects/97/repository/commits?path=SemVer.dll>

<http://gitlab.ztgame.com/api/v4/projects/97/repository/files?private_token=qzBx3gzzjXRRkcfoddsh>&file\_path=builder.json&ref=master

file\_path=myfolder/myfile.txt&ref=master'

<http://gitlab.ztgame.com/api/v3/projects/97/repository/commits?path=build%2FSemVer.dll>

**查看提交记录:**

<http://gitlab.ztgame.com/api/v4/projects/97/repository/commits?path=build%2FSemVer.dll>

&ref\_name=1.1.0

**ref\_name就是tag**

**ref分支**

tech/Client/GameBox.ThirdLib/Releases/SemVer

tech%2FClient%2FGameBox.ThirdLib%2FReleases%2FSemVer

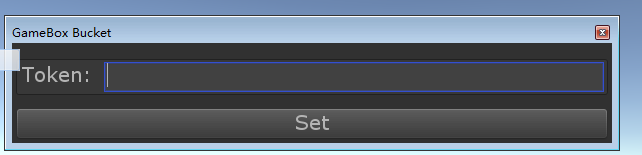
[http://gitlab.ztgame.com/api/v4/projects/**tech%2FClient%2FGameBox.ThirdLib%2FReleases%2FSemVer**/repository/commits?path=net35%2FSemVer.dll](http://gitlab.ztgame.com/api/v4/projects/tech%2FClient%2FGameBox.ThirdLib%2FReleases%2FSemVer/repository/commits?path=net35%2FSemVer.dll) [OK]

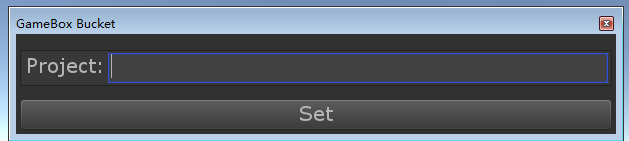
json = "[{\"id\":\"07c744fdc44d51d2021ccaafabfd9c49e13825c8\",\"short\_id\":\"07c744fd\",\"title\":\"tool update\",\"created\_at\":\"2018-04-10T17:23:40.000+08:00\",\"parent\_ids\":[\"064d7e6b49a3465736ef23b20dee7e5786a1a7ce\"],\"message\":\"tool update\\n\",\"author...

http://gitlab.ztgame.com/tech/Client/GameBox.ThirdLib/Releases/SemVer/raw/1.0.0/ net35/SemVer.dll

<http://gitlab.ztgame.com/tech/Client/GameBox.ThirdLib/Releases/SemVer/raw/1.0.0/net35/SemVer.dll>

## Token&Project





**YiBKOjpDMISIWWf2TG7ylA==**

## GxRequest顺序

1==>try to handle: api.gxpack.io/packages/GameBox.Bucket

2==>try to handle: api.gxpack.io/packages/GameBox.Bucket/1.0.0 （Preload-）CollectPackInfo）

3==>try to handle: api.gxpack.io/projects/testproject (SetProject-> OnUpdate)

4==>try to handle: api.gxpack.io/profiles

(ShowBucket🡪OnupdateGxRequest.Compose(GxRequest.GetUser)

5==>try to handle: **api.gxpack.io/recommend (推荐包)**

**C:\Users\wumingsheng\Desktop\gitllab**

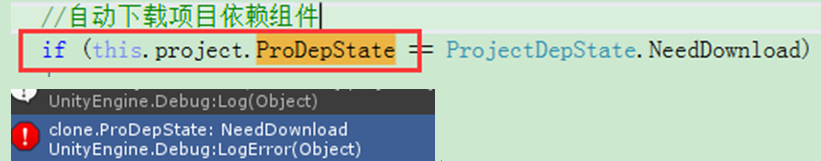
## 个人访问令牌

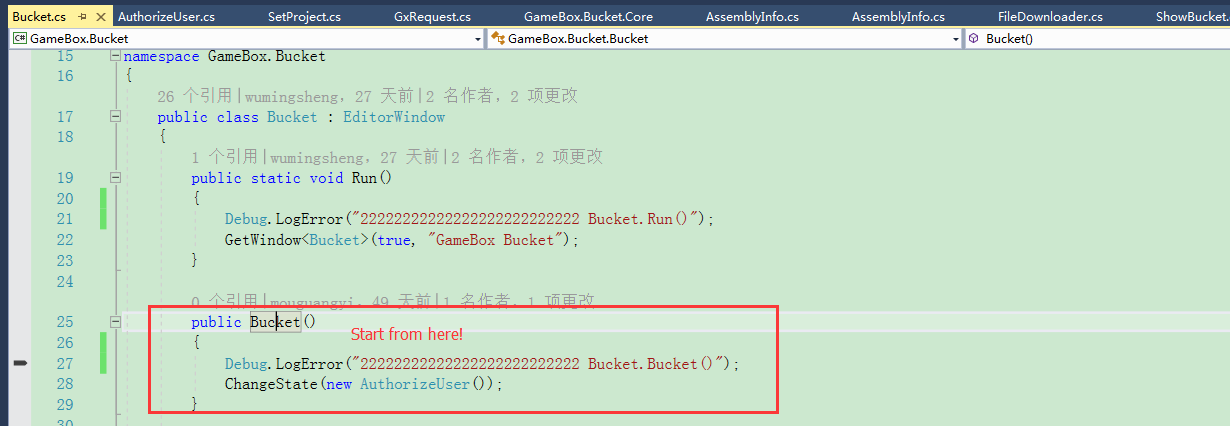
qzBx3gzzjXRRkcfoddsh

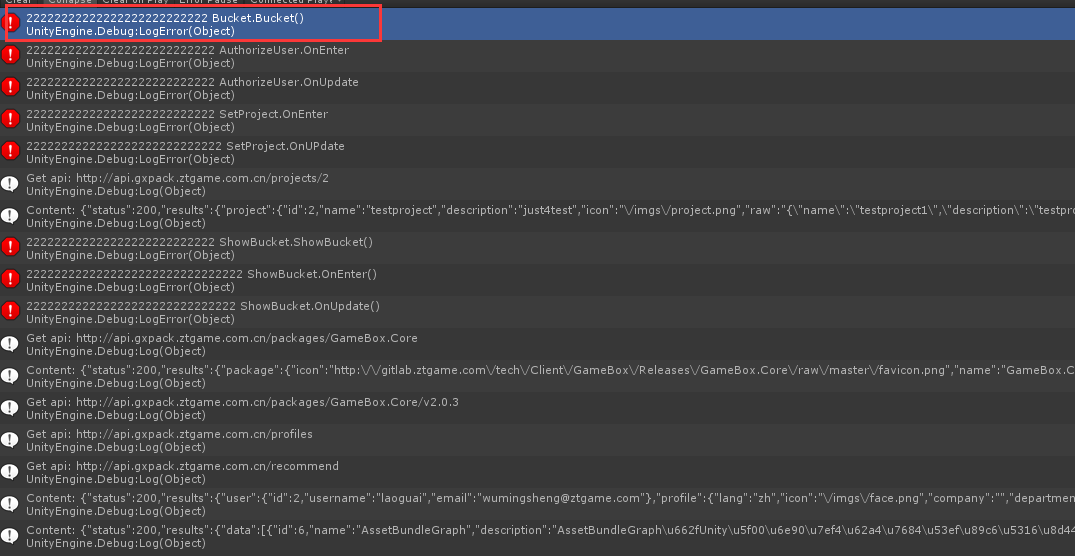
2kxfrnoQVn61ZJxtPZL6

## 自更新

替换完dll后还会执行如下代码： (this.project数据**也会清理掉**？很神奇！！)







## 项目信息

有一份全局的

GxProject.Instance

还有一份在ShowBucket状态上用来临时操作的

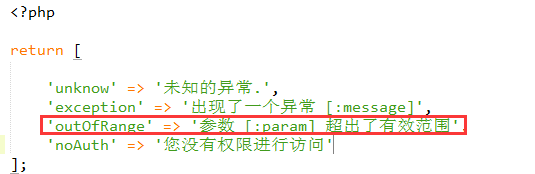
this.project

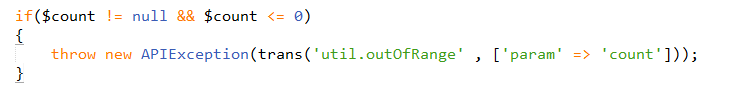
GxProject.Instance🡺 this.project (进入showbucket)

this.project🡺 GxProject.Instance (各种操作成功后)

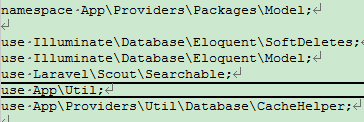
# PHP

## String array dictionary?





## Namespace&lib



## 继承(实现？)

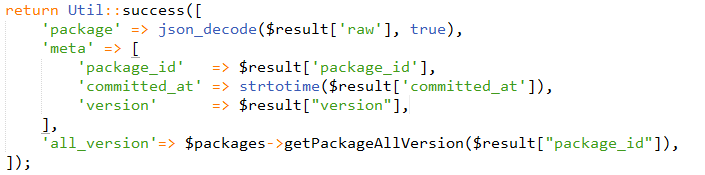


class Packages implements IPackages

## 默认参数



## 返回JSON



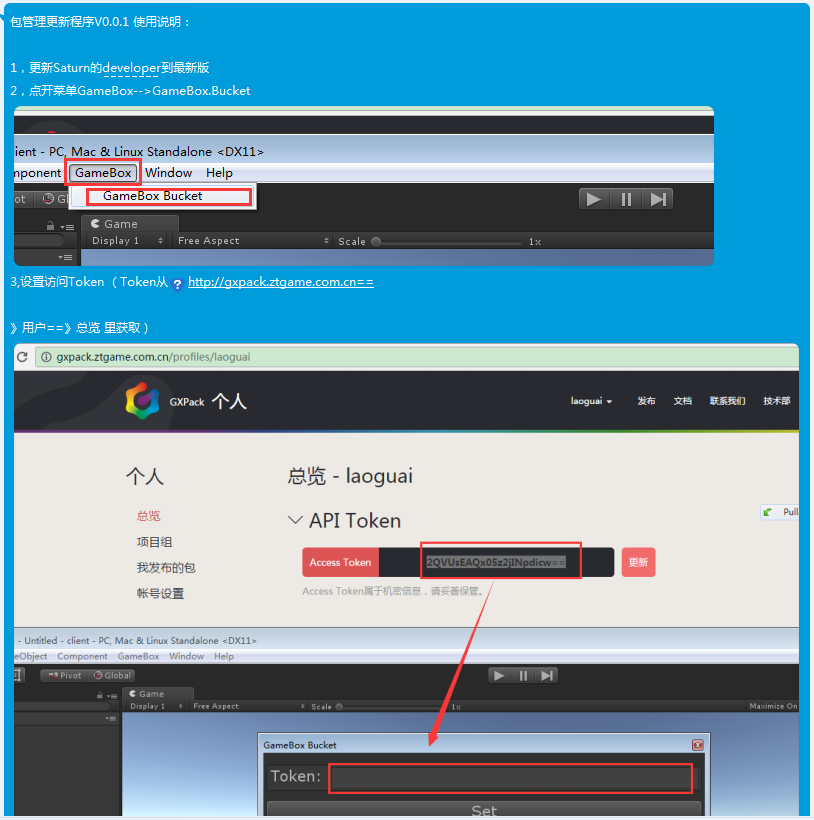
# GXPack

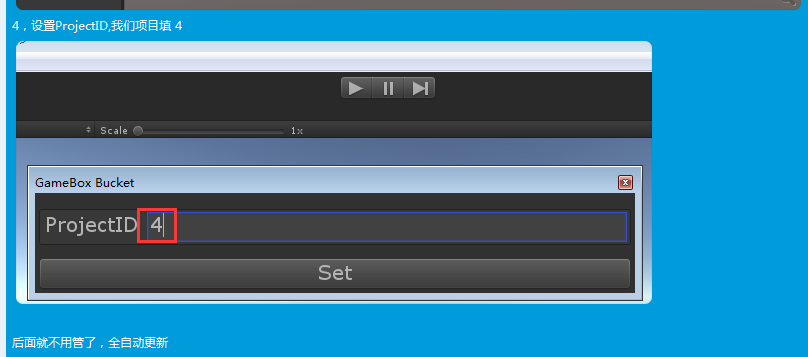
<http://gitlab.ztgame.com/api/v4/projects/tech%2FClient%2FGameBox%2FReleases%2FGXPack.TestProject/repository/tree?&ref=v0.0.13&recursive=true&per_page=1000000>

<http://gitlab.ztgame.com/api/v4/projects/tech%2FClient%2FGameBox.ThirdLib%2FReleases%2FAssetBundleGraph/repository/tree?&ref=v1.4.1&recursive=true>

<http://gitlab.ztgame.com/api/v4/projects/tech%2FClient%2FGameBox%2FReleases%2FGXPack.TestProject/repository/tree?&ref=v0.0.3&recursive=true>

使用说明：





192是牛B机子

193弱鸡

## auth路由



## 发布

walle.gxpack.io

walle 重启 （在mac机上）

nginx和mysql和php-fpm

[wumingsheng@ztgame.com](mailto:wumingsheng@ztgame.com)

ztgame@123

## 监控

<http://stading.cesi.gxpack.io/>

admin:admin

gxpack:gxpack

## webhook

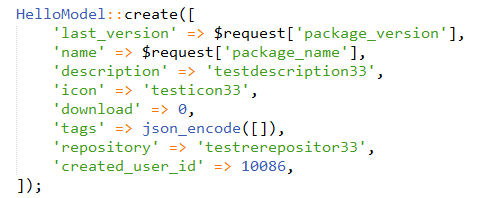
<http://api.gxpack.ztgame.com.cn/webhook>

http://192.168.93.247



## Eloquent ORM & cache

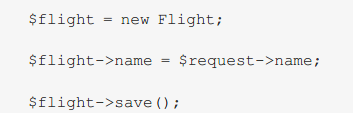
### 创建(添加)



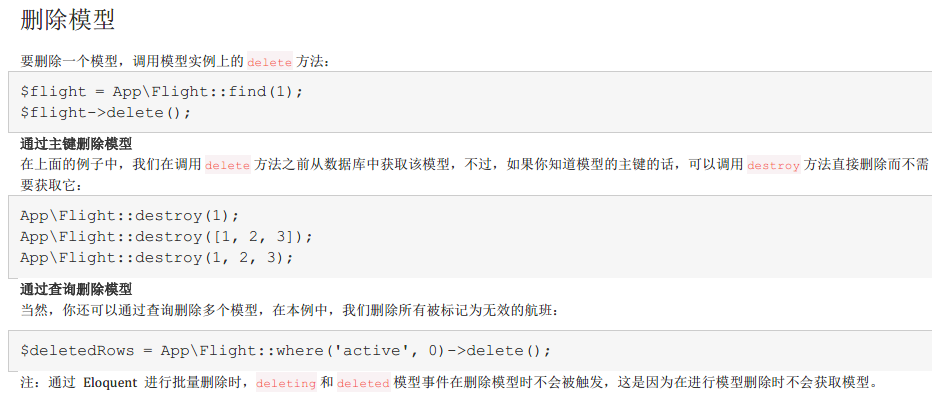
### 修改 where好像返回的集合类型，不能只能调用trashed()接口，但好像可以直接调用restore()接口

$package = $this->update($data);





### 删除



### 软删除



* 之前没使用软删除后来又追加软删除，数据表中需要添加deleted\_at字段

添加字段可以通过php的数字迁移来完成

php artisan make:migration alter\_posts\_deleted\_at --table=hellomodel

会生成AlterPostsDeletedAt文件，修改如下：



最后执行php artisan migrate 完成数据表的结构修改

REF：<https://www.cnblogs.com/ghjbk/p/6638125.html>

{{ Auth::user()->username }}

### 恢复

HelloModel::withTrashed()->where ('last\_version', $request['package\_version'])->restore();

### 强制删除

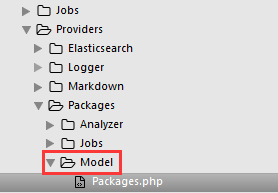
forceDelete

## Mark

### 路径

Namespace设置成文件夹的路径





调用只能用相对路径？？

### Php self和static

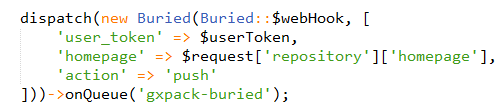
PHP官方也说过，大概意思是说self调用的就是本身代码片段这个类，而static调用的是从堆内存中提取出来，访问的是当前实例化的那个类，那么 static 代表的就是那个类，例子比较容易明白些。

### 埋点

dispatch(new Buried(Buried::$webHook, […

dispatch(new Buried(Buried::$webHookFaild, […

### 数组



protected $fillable = [

'name', 'description', 'created\_user\_id', 'icon', 'repository' , 'download', 'last\_version', 'tags' , 'visitor'

];

### 获取客户请求信息

头信息：

$request->headers->get("X-Gitlab-Token")

Body信息：

$request['repository']['homepage']



{

"object\_kind": "tag\_push",

"ref": "refs/tags/v1.0.0",

"project\_id": 1,

"repository":{

"url": "ssh://git@example.com/jsmith/example.git",

"homepage": "http://example.com/jsmith/example",

"git\_http\_url":"http://example.com/jsmith/example.git",

"git\_ssh\_url":"git@example.com:jsmith/example.git",

"visibility\_level":0

},

"commits": [],

"total\_commits\_count": 0

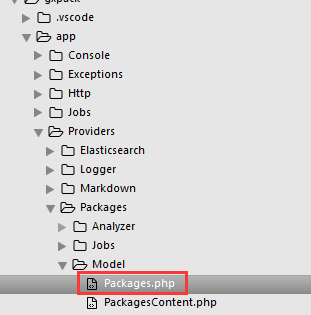
}

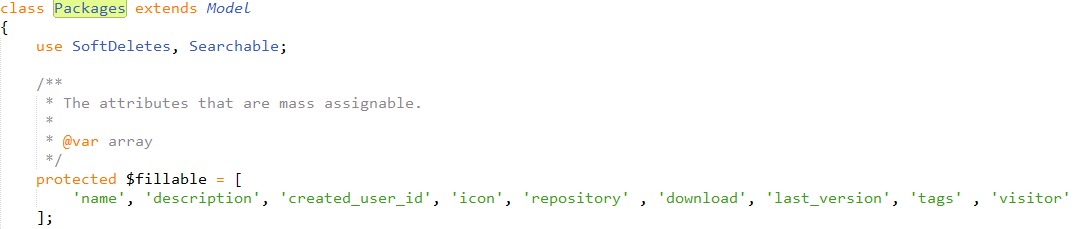
### 软删除

use SoftDeletes, Searchable;

### Package相关

数据字段：







## 功能模块

### 注册

Validator🡺 create🡺 guard()->login 🡺registered

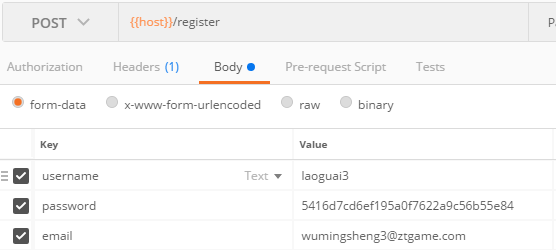
$users->register

users.php

public function register($username, $password, $email)

mode/users.php

public static function register($username, $password, $email)



RegisterController.showRegistrationForm🡺

RegisterController. makeNav🡺

RegisterController. Validator🡺

RegisterController.Create

### 登录

AuthenticatesUsers.login🡺

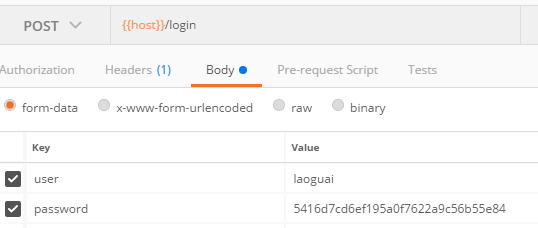
LoginController.validateLogin🡺

LoginController.attemptLogin



protected function **attemptLogin**(Request $request)

密码用md5(32位)加密:



### 获取AccessToken

get

{{host}}/profiles/token

### 更新AccessToken

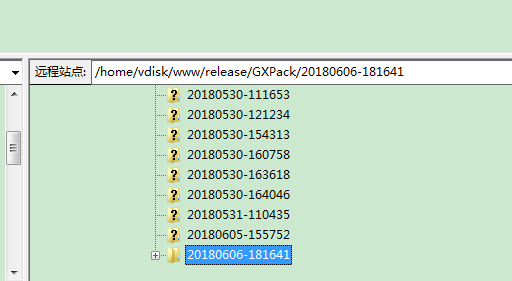
post

{{host}}/profiles/token

### 创建项目组

post

{{host}}/projects/create



{\"name\":\"test25\",\"description\":\"hello description\"}

{

"name": "testproject",

"description": "testproject haha"

}

//$raw = '{\"name\":\"' . $data['project\_name'] . '\",\"description\":\"' . $data['project\_description'] . '\"}';

### 获得项目操作日志

{{host}}/projects/{{project\_id}}/logs/0

### 创建tag时webhook

WebHookController.php

public function onWebHook(Request $request)

获取http头部的token:

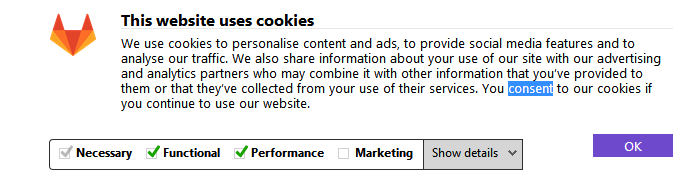
$userToken = $request->headers->get("X-Gitlab-Token");

https://docs.gitlab.com/ee/user/project/integrations/webhooks.html

X-Gitlab-Token HTTP header.

X-Gitlab-Event: Push Hook



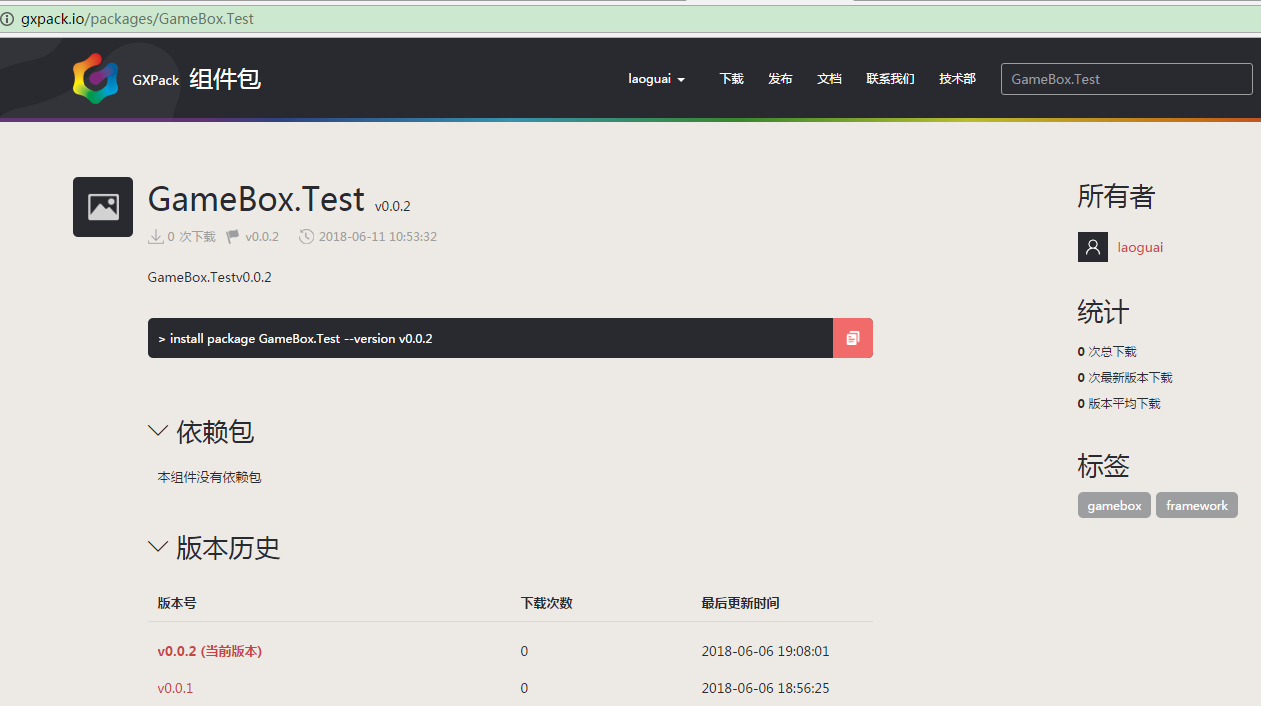


### 删除tag时webhook、埋点

### create package

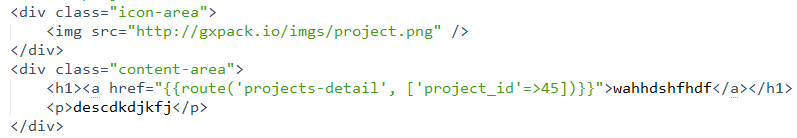
public function submitPackage($userId, $repository)





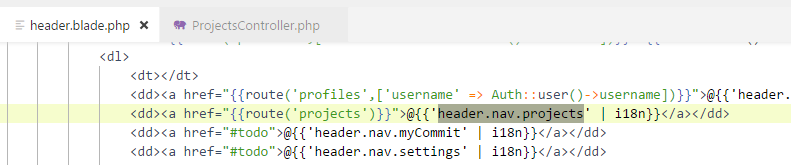
http://gxpack.io/imgs/project.png

### create project



### 包管理

发起



### 设置

Auth::user()->email

Auth::user()->username

### 邮件功能



php artisan make:mail ForgetPassword

**kiksdehfeoaocabd**



<https://www.jianshu.com/p/8ccb2820df23>

php artisan make:controller MailController

## 备用监控

192.168.23.193:9001

gxpack:gxpack

## 压力测试

ab -n 30000 -c 20 <http://gxpack.ztgame.com.cn/packages/GameBox.Core>

Queue没消费掉

动态加载负载均衡：

SuporVisor?

调整队列消费：

Php-fpm配置里调？

**fpm.sock出错**

那是fpm来不及消费了

**出现大批502错误是fpm队列满了**

**Rabbitmq队列是数据是代码直接传给rabbitmq的**

**访问极限：250次每秒**

**php nginx**

## .env

QUEUE\_DRIVER=sync



## php

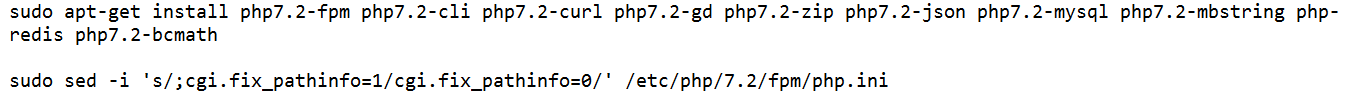
入口：？

<http://127.0.0.1:81/public/index.php>

.env

只关注功能的话

**nginx php mysql rabitmq**



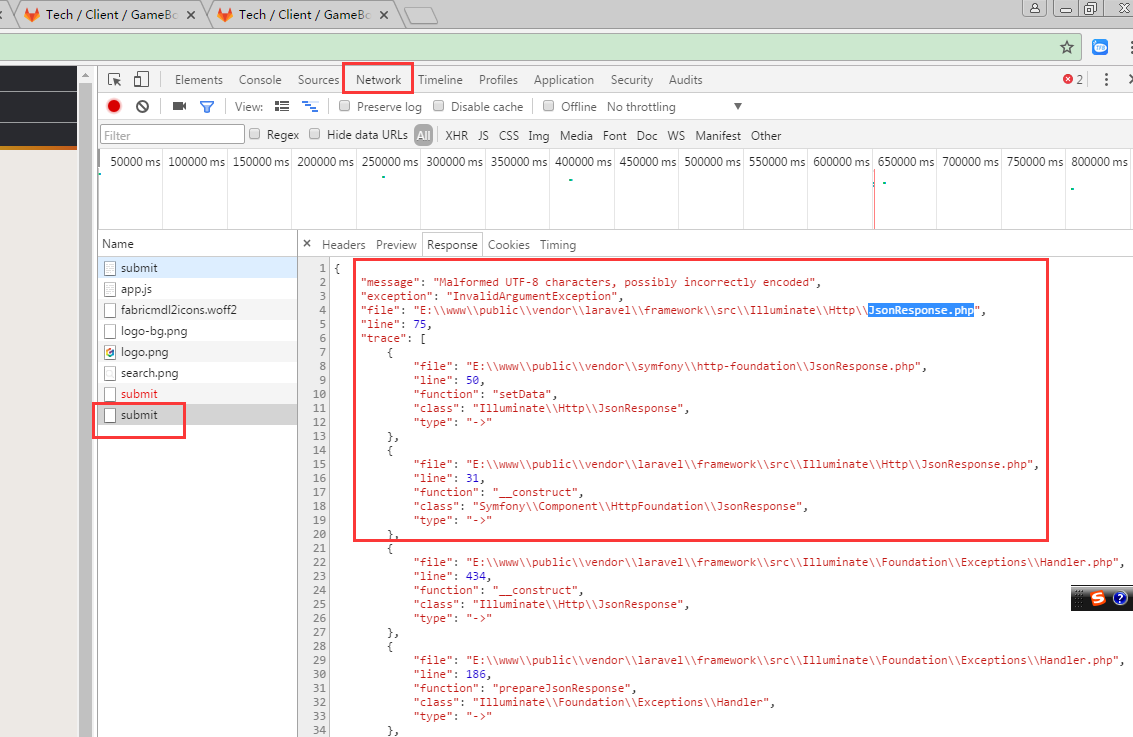
**上面对应的配置要打开**



[**http://pecl.php.net/package/redis/4.0.2/windows**](http://pecl.php.net/package/redis/4.0.2/windows)

### 调试：

<h1><a href="{{route('package', ['package\_name'=>$package["name"])}}">{{$package["name"]}}</a></h1>



[**如何在sublime+chrome中调试php代码？**](https://www.cnblogs.com/jewave/p/5721103.html)

https://www.cnblogs.com/jewave/p/5721103.html

<http://pecl.php.net/package/xdebug/2.6.0/windows>

**提交组件：**

public function submitPackageAsync($userId, $repository)

### 执行出错代码：

$array =debug\_backtrace();

unset($array[0]);

foreach($array as $row)

{

$html .=$row['file'].':'.$row['line'].'行,调用方法:'.$row['function']."<p>";

}

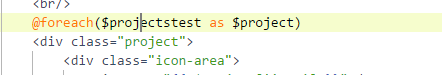
echo $html;

可以查看出错堆栈

### 数据结构

* 数组 $packageVersionsShow[]





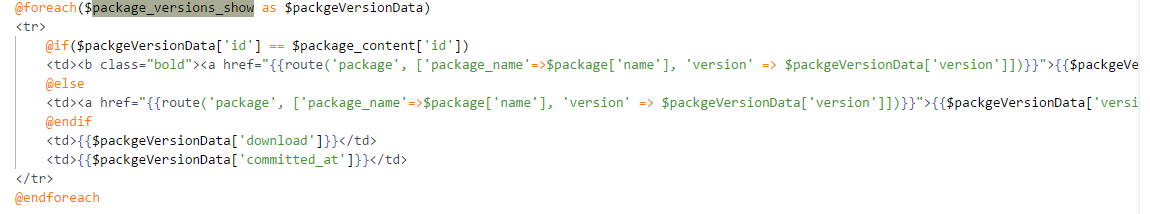
* array



* Collection







## nginx

tasklist|findstr "nginx"

taskkill /f /t /im nginx.exe

## npm

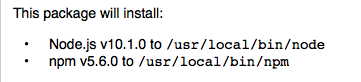
npm run watch

npm install

npm update

## Node.JS

<http://nodejs.cn/>



<http://nodejs.cn/download/>

node install

## Composer

安装好后在根目录 （代码的根目录，配置文件composer.json所在的目录）

composer install

更新的依赖包在vendor目录下

composer config -g repo.packagist composer <https://packagist.phpcomposer.com>

composer update

## Laravel

### 开发环境

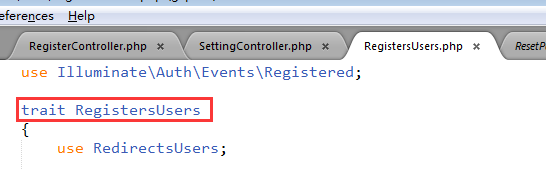
<http://laravelacademy.org/?p=8016>

### 奇怪的路由

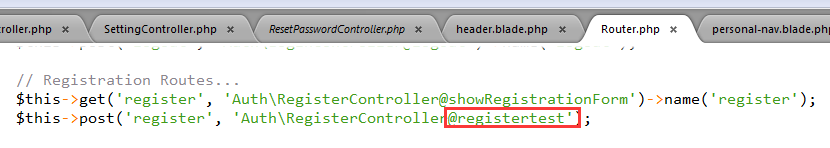
Validator🡺 create🡺 guard()->login 🡺registered

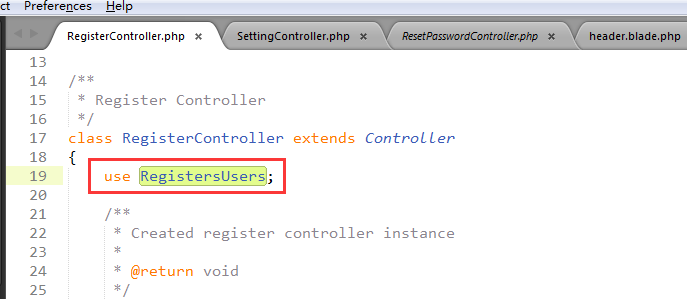
$users->register

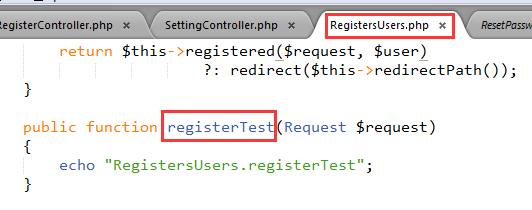
RegisterController@registertest==>RegisterUsers@registertest



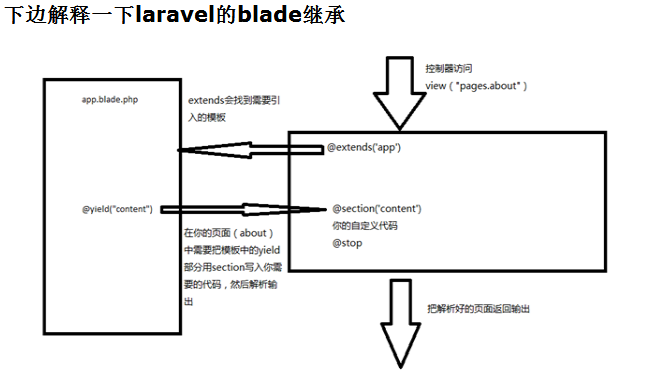
Trait可以看做类的部分实现 (RegisterUsers可以看做是RegisterController的一部分)



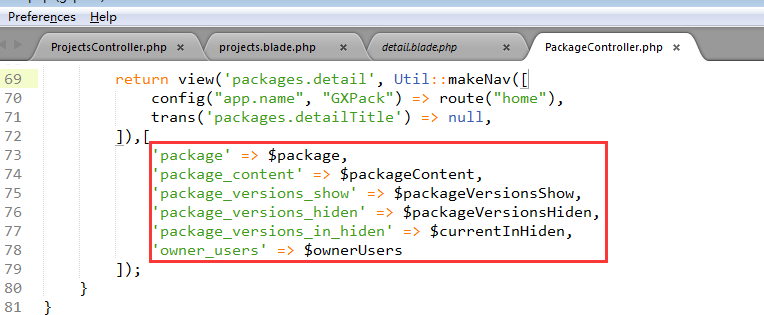


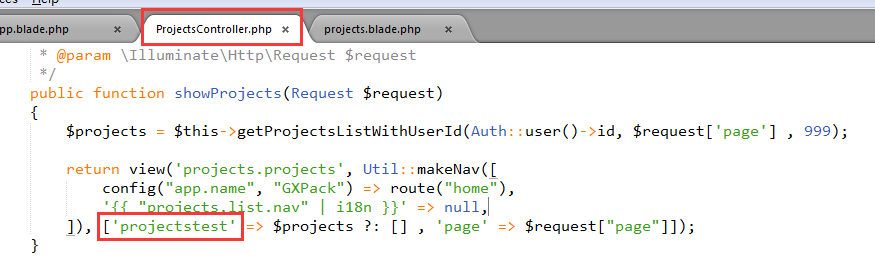


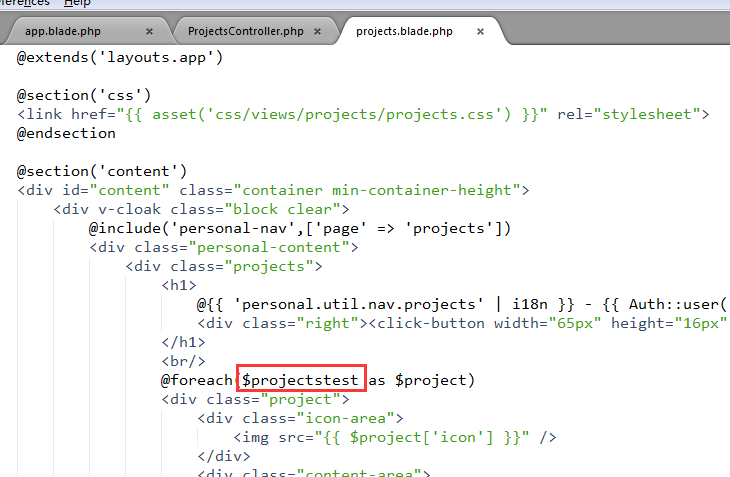
## Blade



### 参数传递：





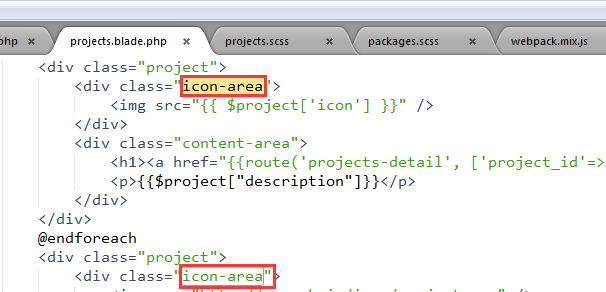


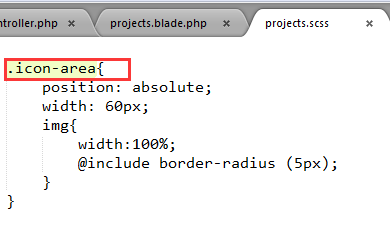
## CSS

### 编译sass (.scss)

npm run dev

### 实例





## Html

### foldable-area





### 标签

thead

tbody

## Vue

### Vbind



### v-model (输入)



### v-on (事件处理)

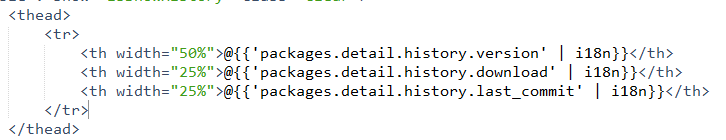


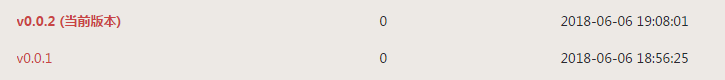
### v-show (根据条件展示元素的选项)

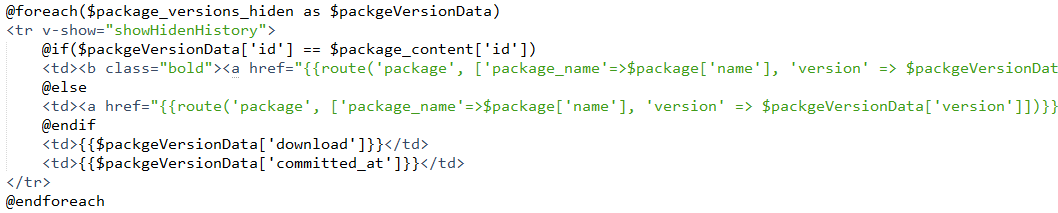


### 实例









## Rabbitmq

## Redis

Mac下用如下指令启动：

sudo redis-server /etc/redis.conf

进入redis:

 redis-cli -p 6379

查询当前redis有没设置密码:

 redis-cli -p 6379

设置密码：

config set requirepass ztgame@123

**auth 123**

**配置文件路径：**

/etc/redis.conf

重启：

brew services restart redis

**查询：**

keys \*

193 redis

ztgame@gxpack

## Kibana

/usr/local/Cellar

/usr/local/Cellar/kibana/6.2.3

## Sentinl

1，下载与kibana相同版本的sentinl压缩包

2，

/usr/local/Cellar/kibana/6.2.3/bin/kibana-plugin install file:///Users/Shared/Jenkins/Downloads/sentinl-v6.2.3.zip

## grafana

stading.grafana.gxpack.io

用户名和密码

gxpack:gxpack

安装：

brew reinstall grafana

/usr/local/etc/grafana/grafana.ini

/usr/local/var/log/grafana/grafana.log

/usr/local/var/lib/grafana/plugins

/usr/local/var/lib/grafana(sqlite db)

初始用户名和密码:

admin

admin

日志：

/usr/local/var/log/grafana/grafana.log

## filebeat

Mac：

查看filebeat信息：

brew info filebeat

安装路径：

/usr/local/Cellar/filebeat/6.2.4

配置文件：

/usr/local/etc/filebeat/ filebeat.yml

## MetriBeat

Mac:

curl -L -O https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-6.2.4-darwin-x86\_64.tar.gz

tar xzvf metricbeat-6.2.4-darwin-x86\_64.tar.gz

/Users/Shared/Jenkins/metricbeat-6.2.4-darwin-x86\_64

启动Metricbeat

sudo nohup ./metricbeat -e -c metricbeat.yml &

sudo ps -ef | grep metricbeat

## Logstash

配置文件

/usr/local/Cellar/logstash/6.2.3/libexec/config/logstash.yml

logstash.conf文件要自己创建

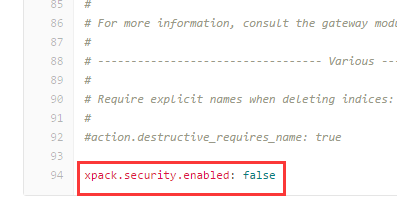
~/logstash/logstash.conf

## Elasticsearch

Version: 6.2.3

<http://stading.elasticsearch.gxpack.io>

加了下面的配置会报错：



/usr/local/Cellar

配置文件：

/usr/local/etc/elasticsearch/elasticsearch.yml

重启：

brew services restart elasticsearch

### 命令

* 查询Index

GET {{elasticsearch\_host}}/\_cat/indices?v

* 删除Index

DELETE {{elasticsearch\_host}}/gxpack.packages (gxpack.packages是Index名字)

* 创建Index

PUT {{elasticsearch\_host}}/gxpack.packages

* 创建Index的mapping

{{elasticsearch\_host}}/gxpack.packages/\_mapping/gxpack.packages

Header: 

Body:



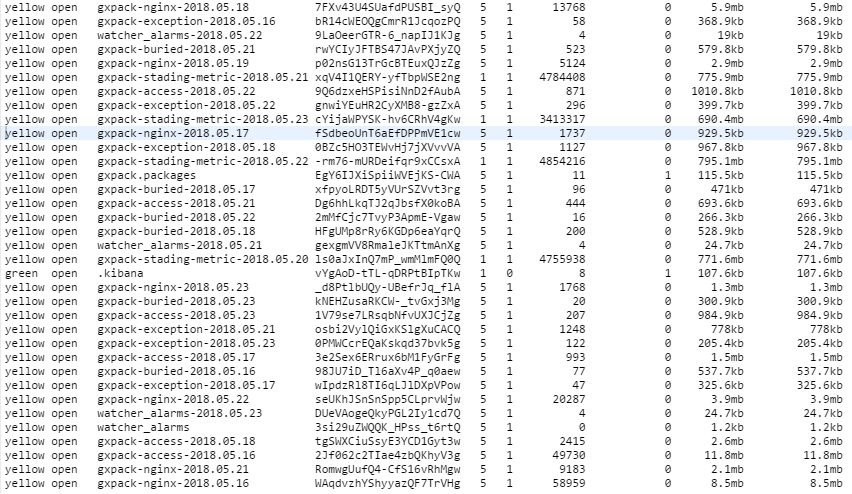
* 获取Index的mapping

GET {{elasticsearch\_host}}/gxpack.packages

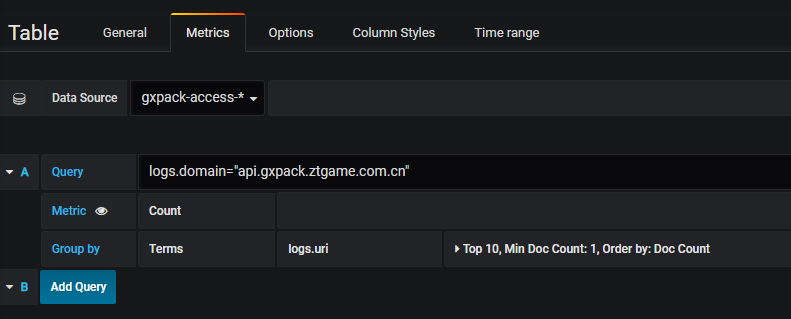
* 获取Index的搜索结果(搜索Index)

GET {{elasticsearch\_host}}/gxpack.packages/\_search?q=\*&pretty

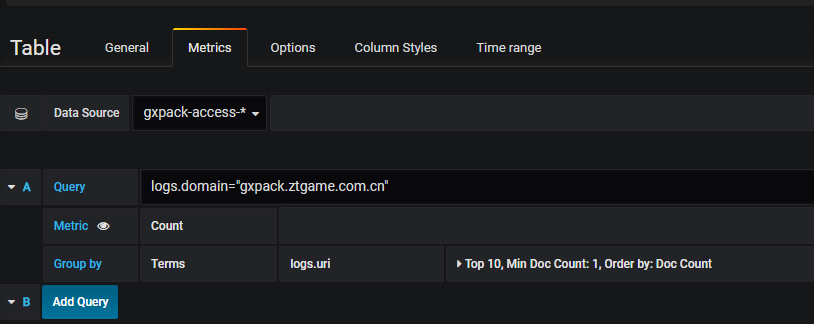
### Index索引 192.168.23.192 (牛鸡)



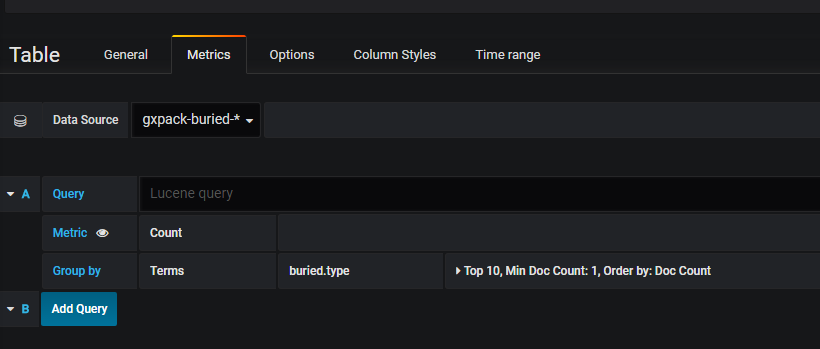
api.gxpack.io



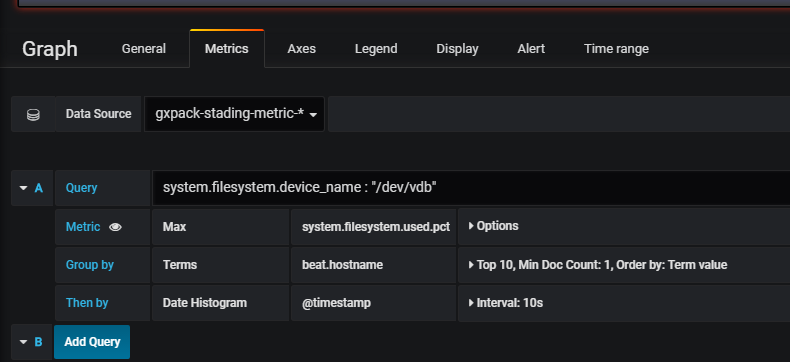
gxpack.io



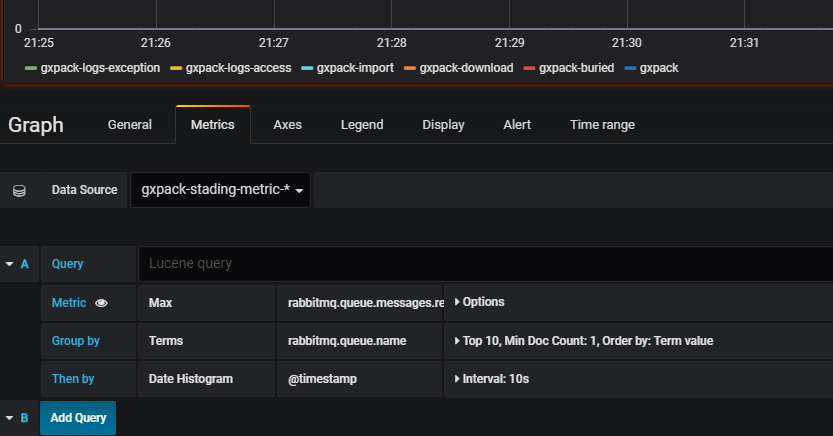
Buried



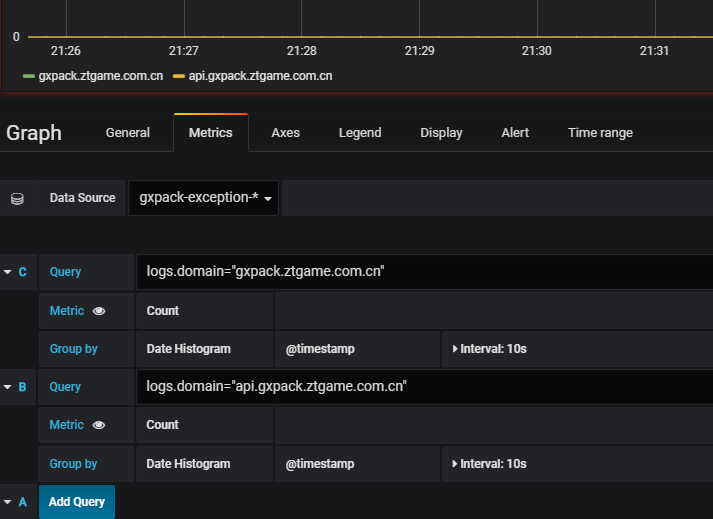
Diskused



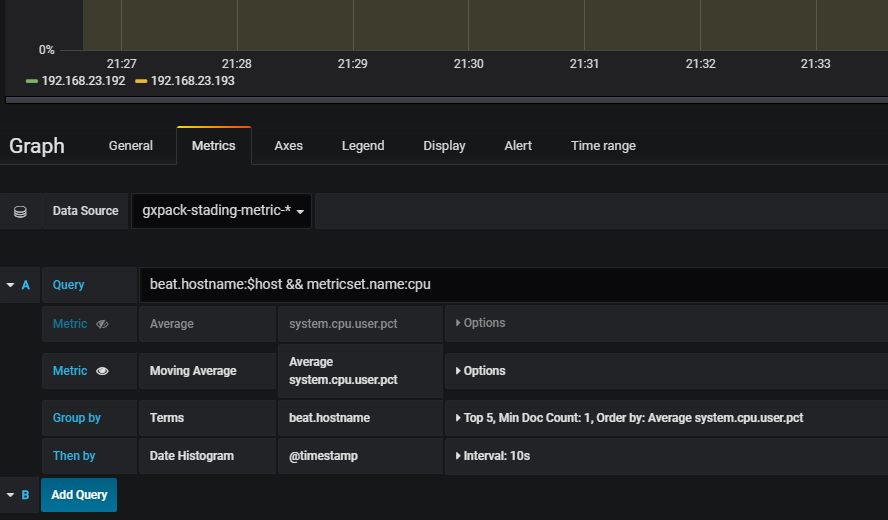
Queue



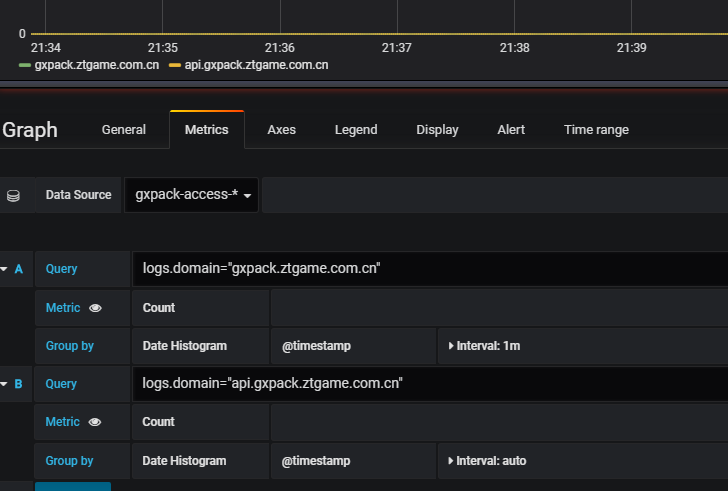
Exception



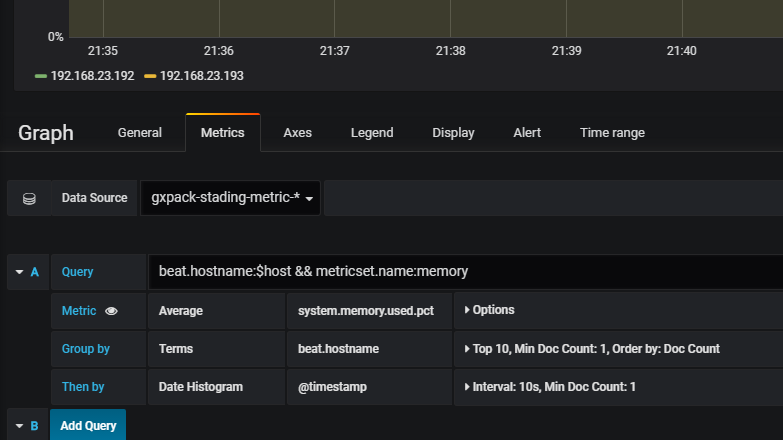
CPU



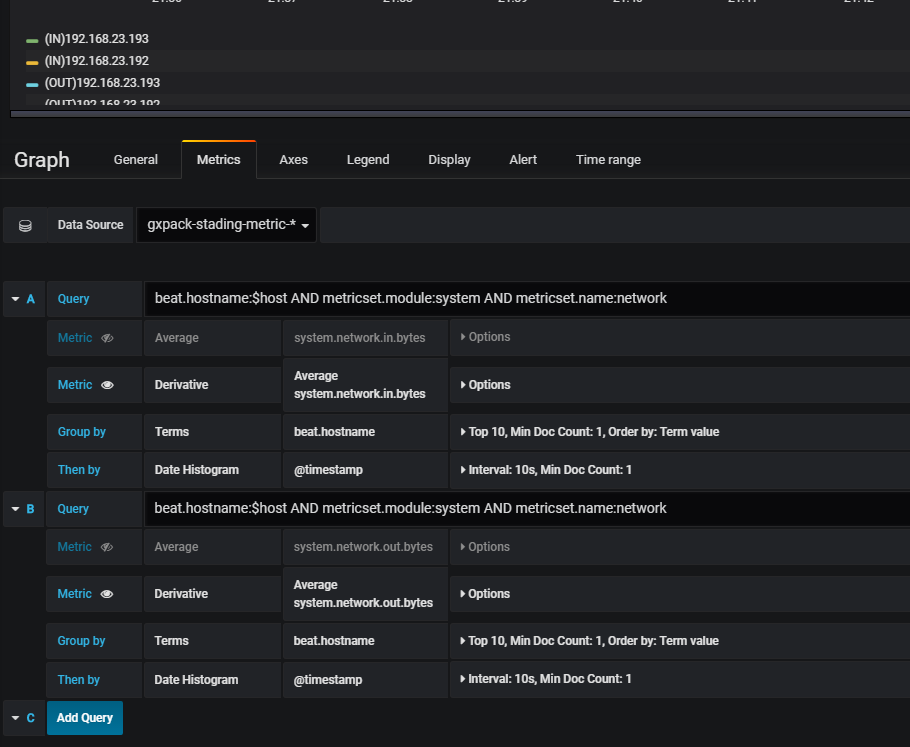
QPM



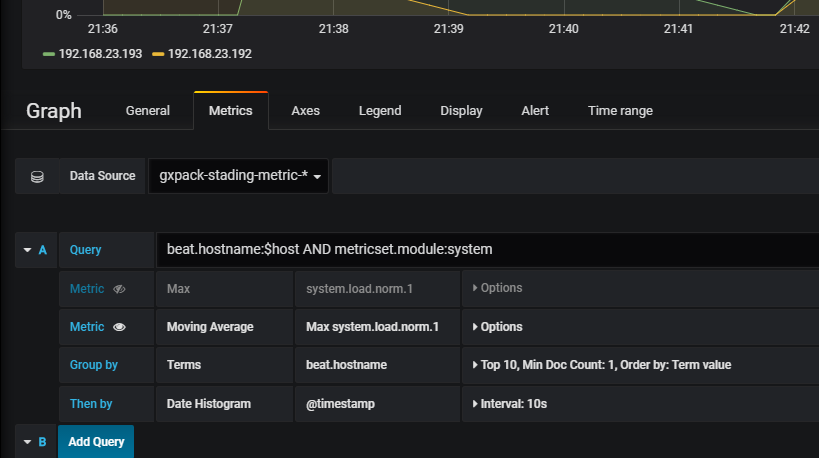
Memory



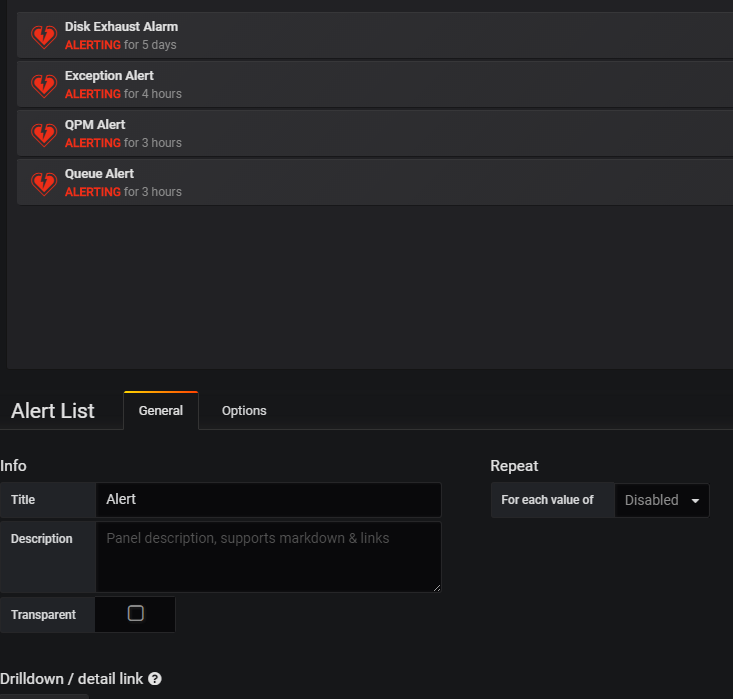
Network

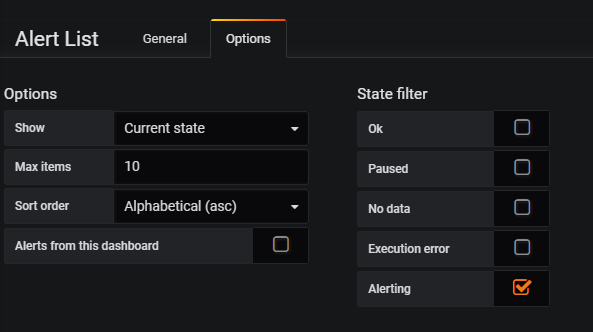


1Min Average load

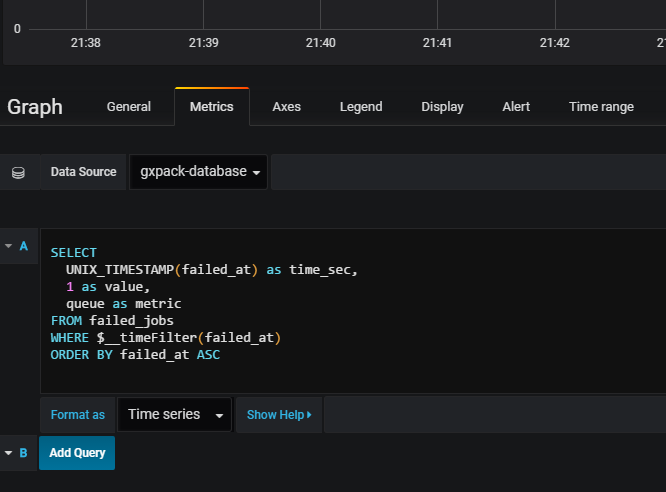


AlertList

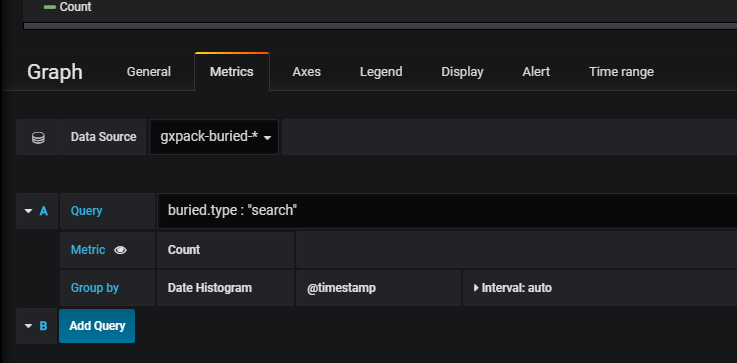




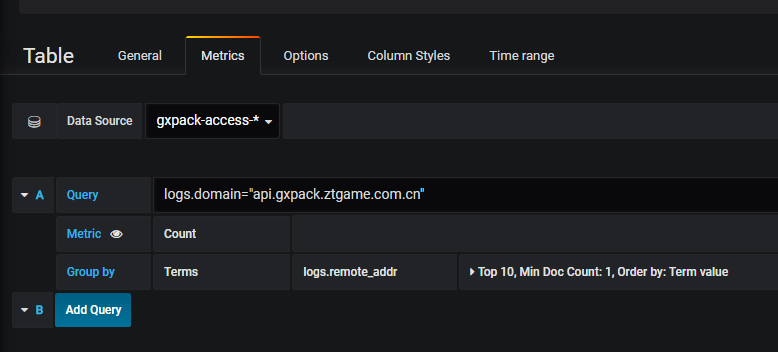
Failed Jobs



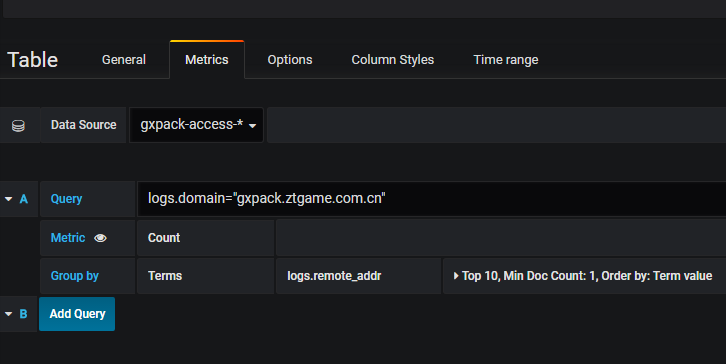
SearchCount



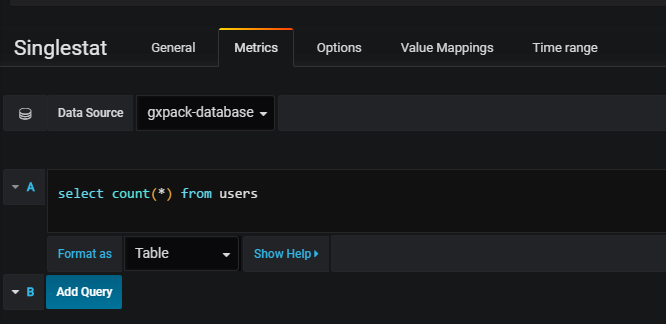
API Access IP



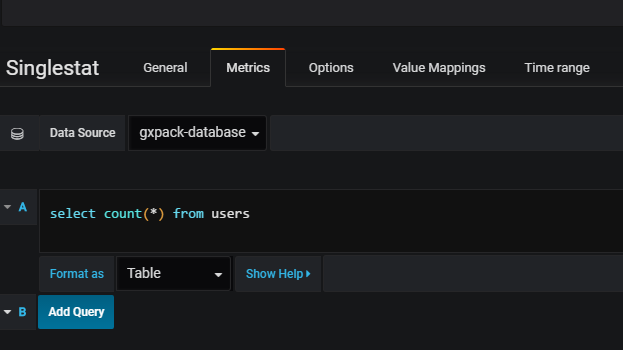
Web Access IP



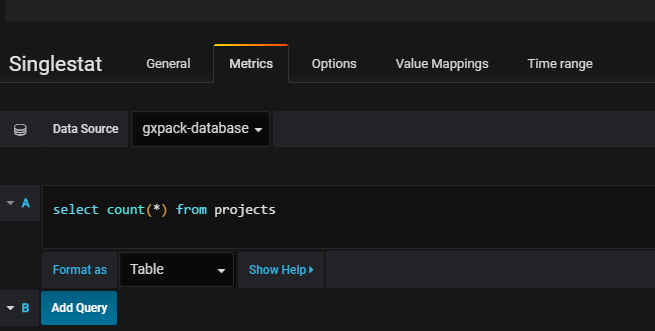
User Count



Project Count

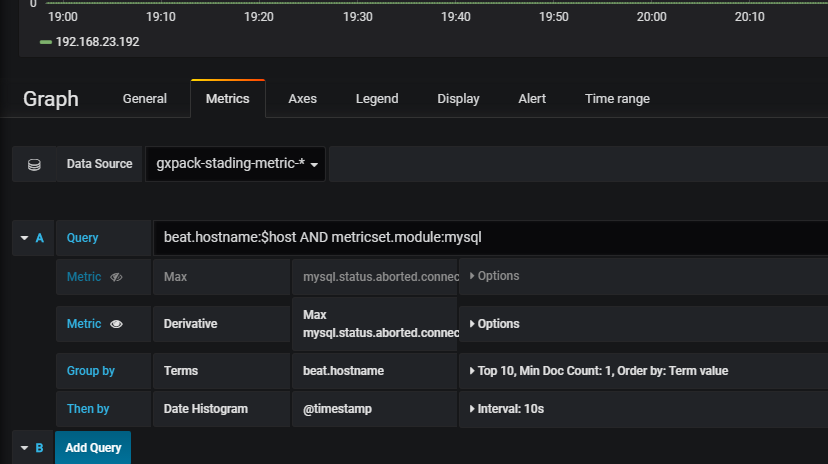


Package Count

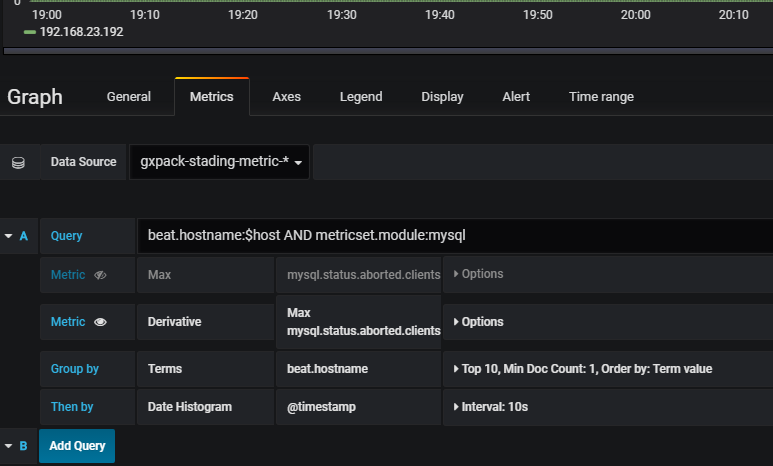


===============================================================================mysql

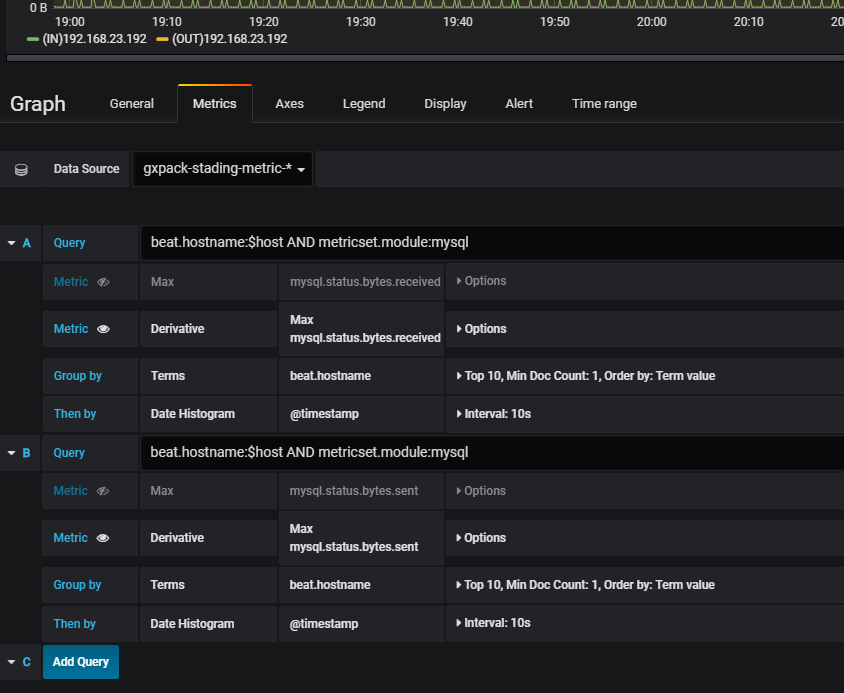
Aborted Connects



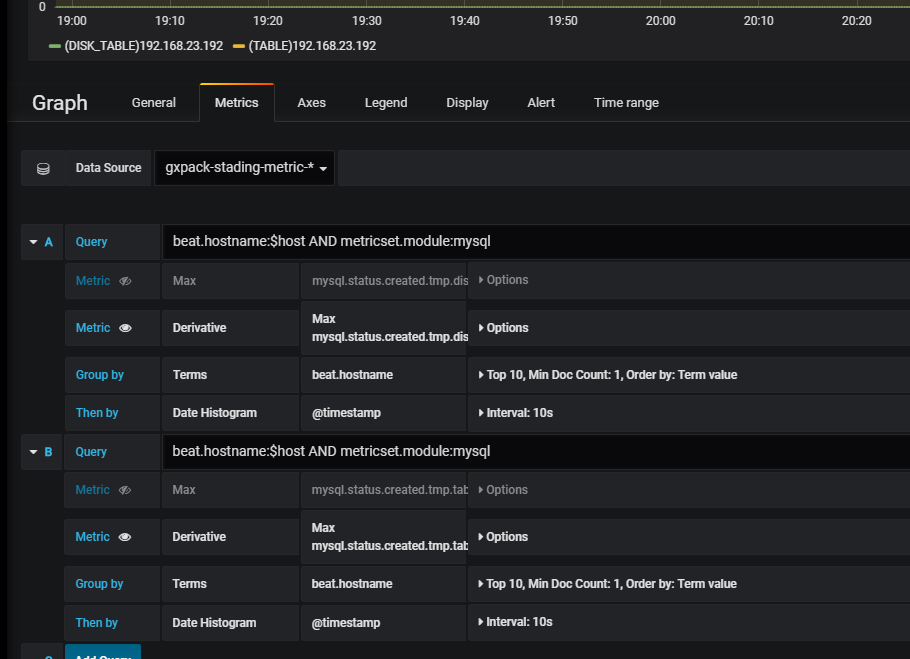
Aborted Clients



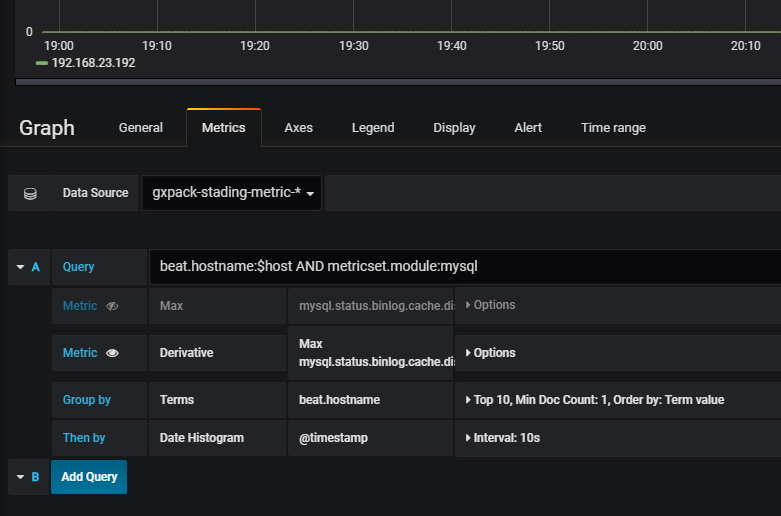
MySQL Net IO



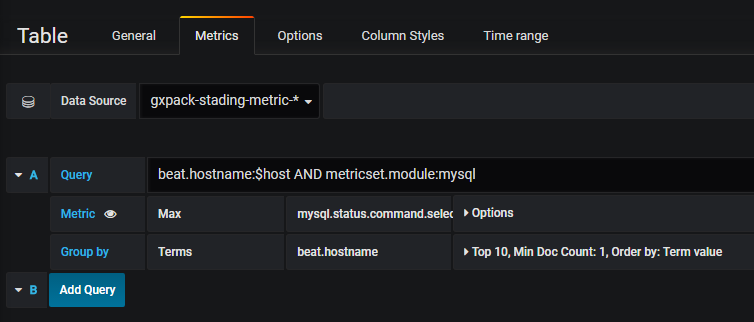
TMP DiskTables



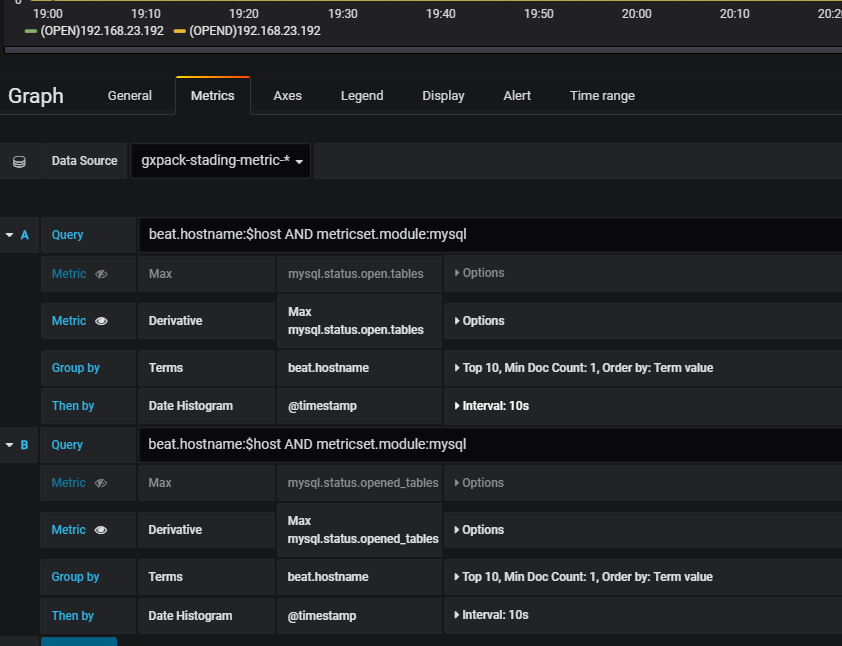
Binlog Disk Use



Select Count

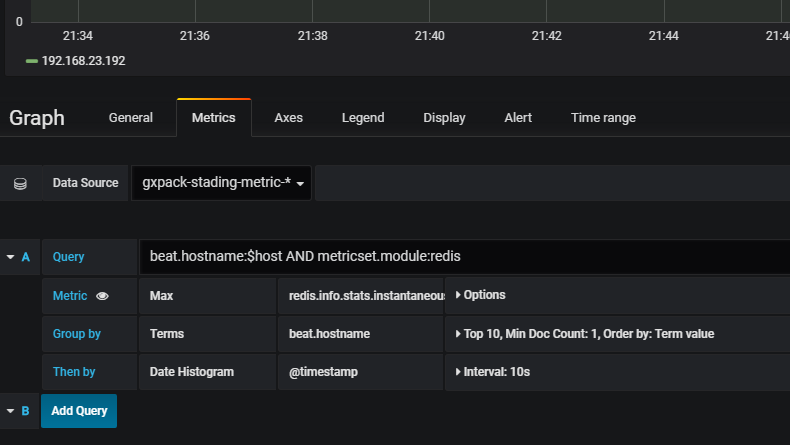


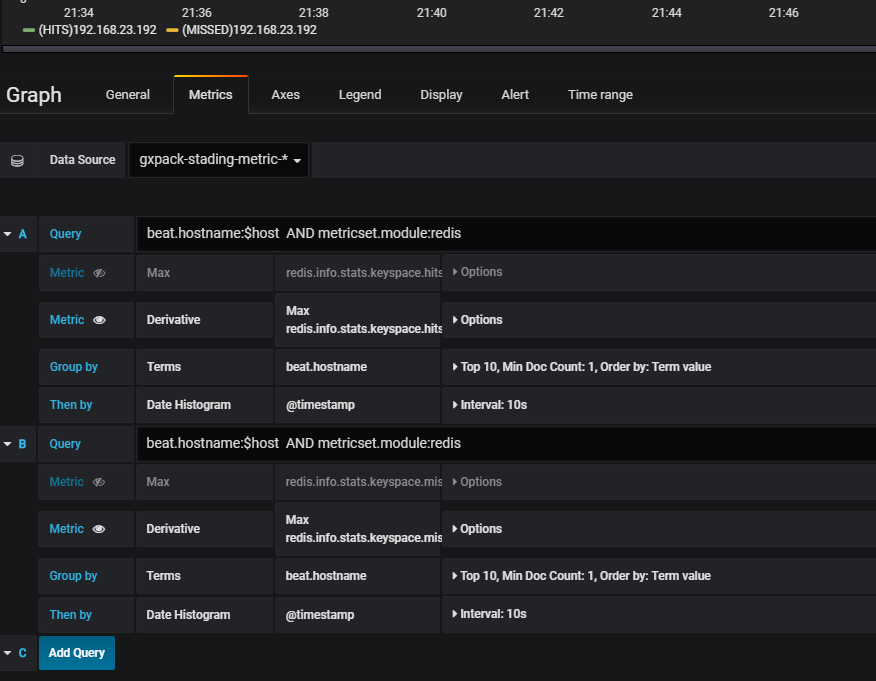
Opened Tables

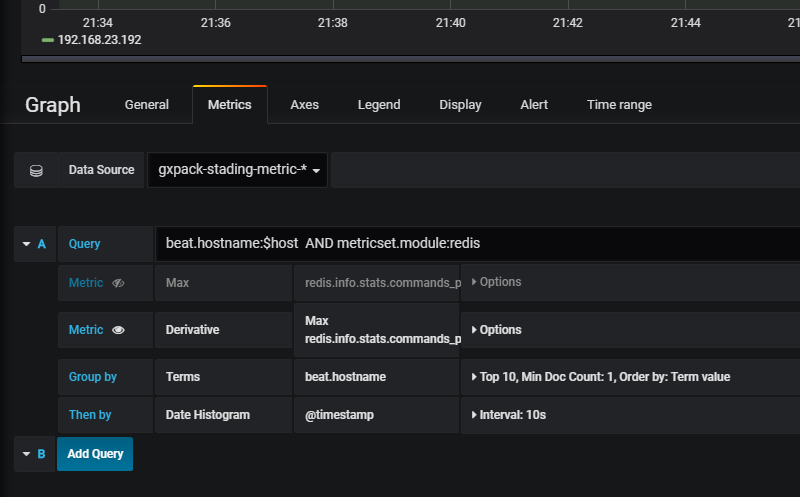


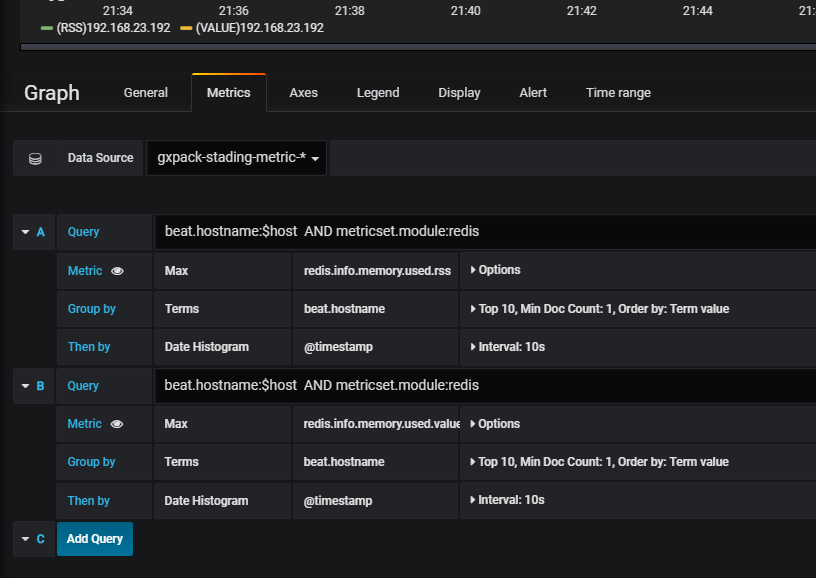
===============================================================================redis

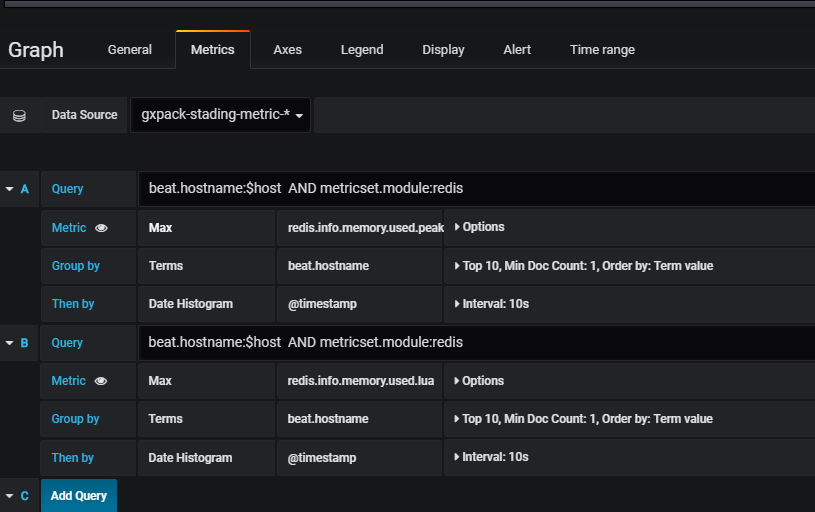
ops

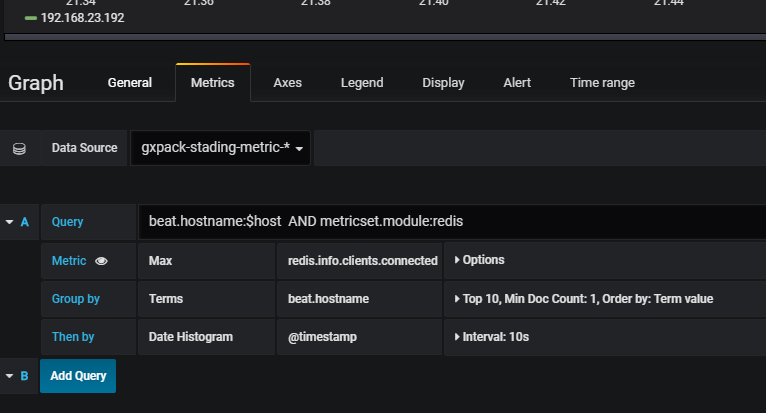


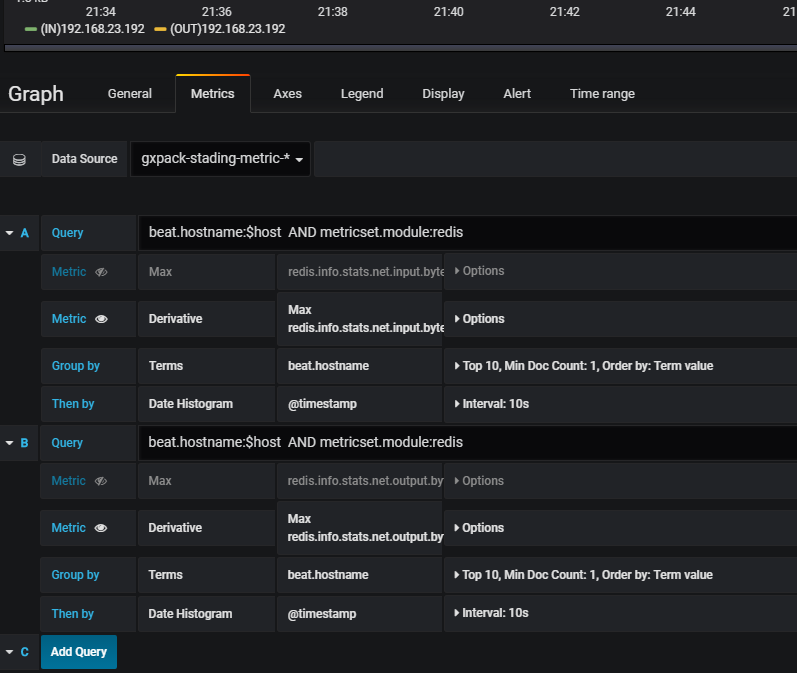




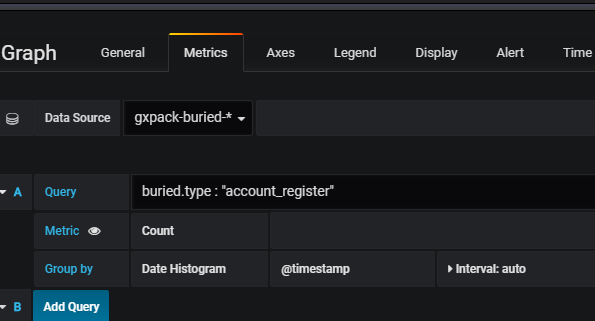


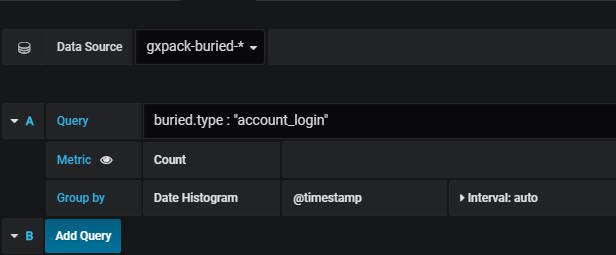


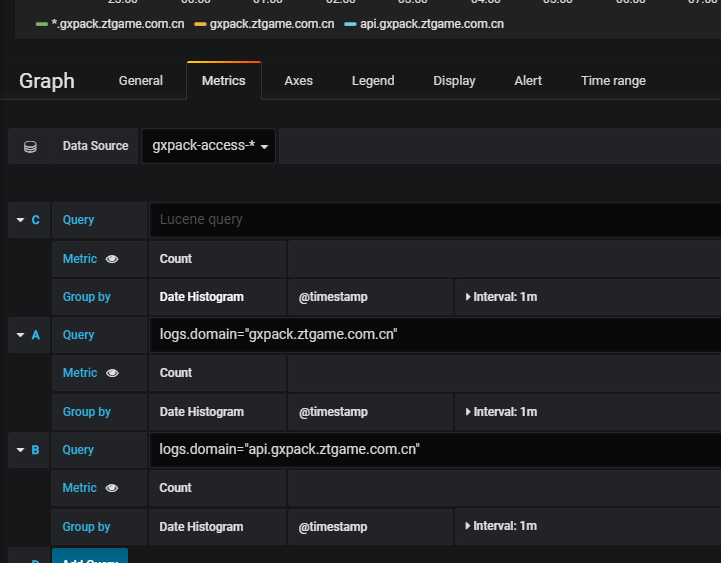


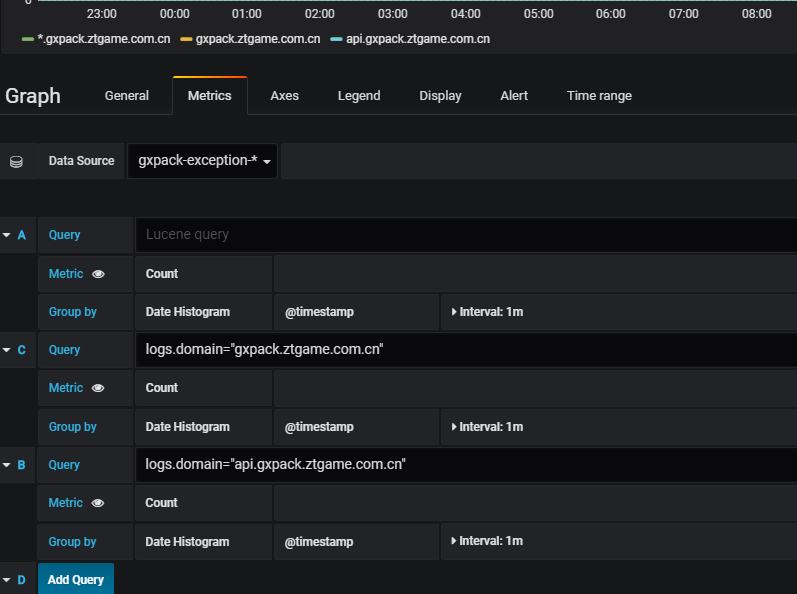


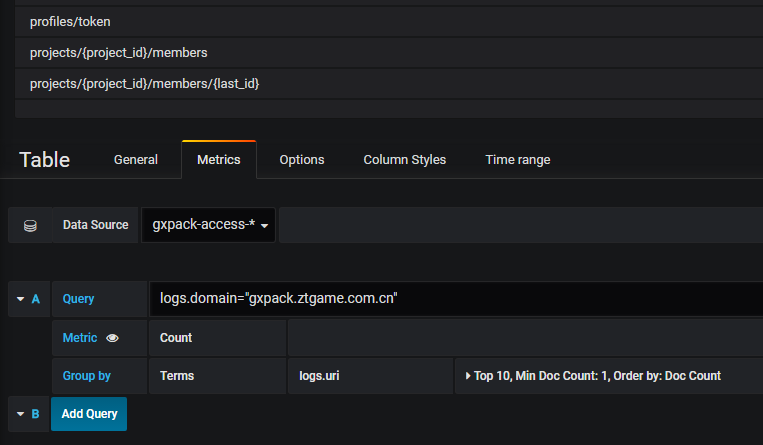
===============================================================================Web Monitor

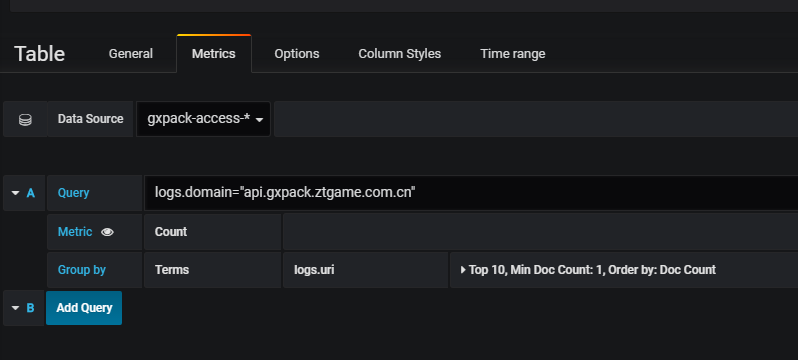


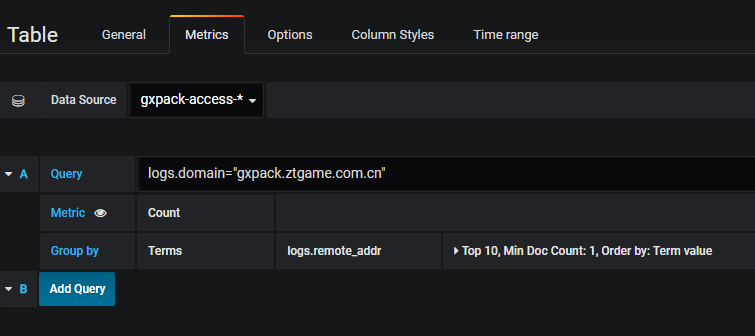


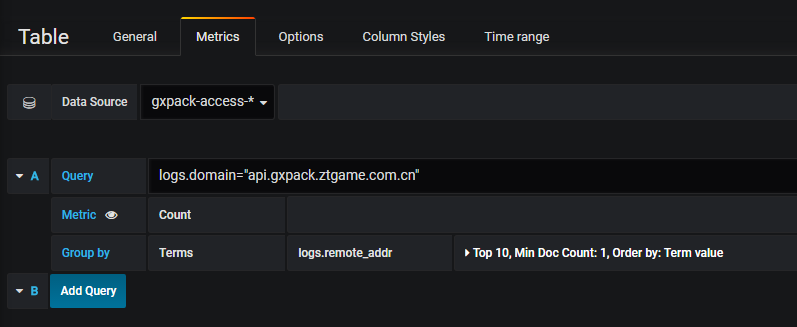






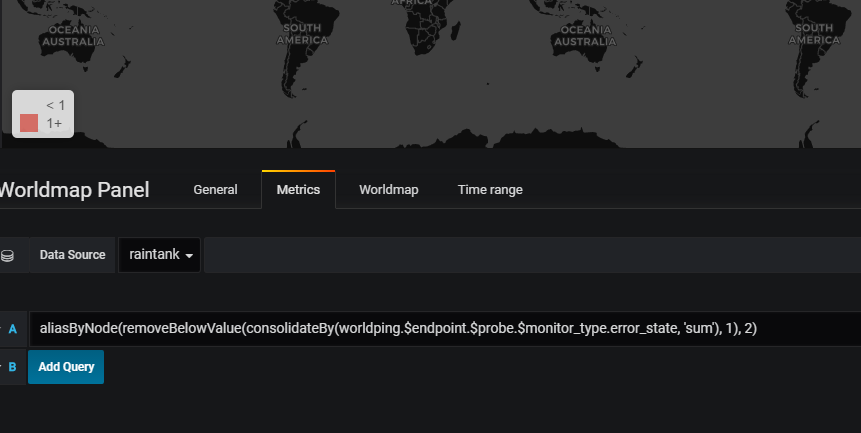


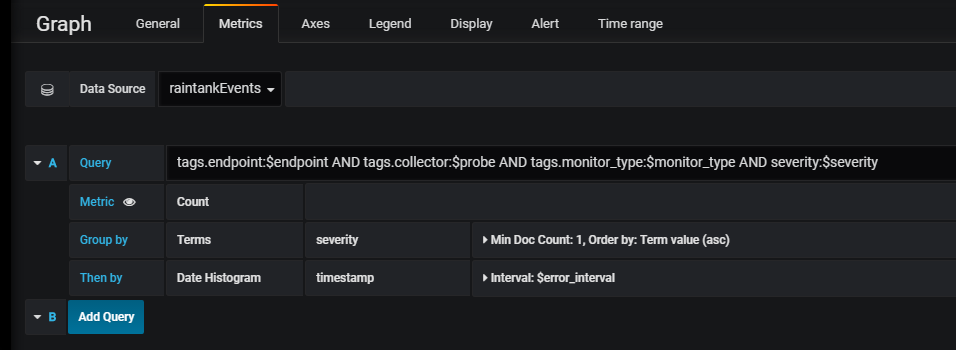


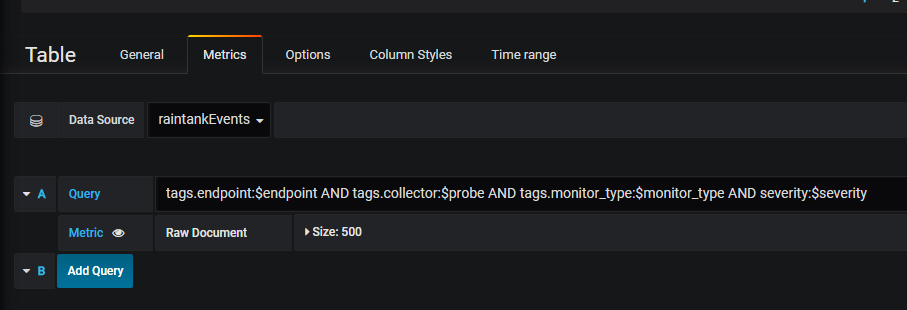


=====================================================================

World Pinghttp://stading.kibana.gxpack.io







## nginx

**Mac：**

配置文件：

/usr/local/etc/nginx/nginx.conf

重启：

brew services restart nginx

**Note:**

**要用80端口启动nginx必须用sudo起**

## php-fpm

Mac:

查看版本号

php-fpm –version

启动：

php-fpm

寻找配置文件路径：

php-fpm –t

配置文件：

/usr/local/etc/php/7.2/php-fpm.conf

之前碰到系统起的之前版本的情况，把新版的php-fpm加载.bash\_profile的path环境变量里再重启php-fpm就好了:

export PHP72PATH=/usr/local/opt/php@7.2/sbin

## phpMyAdmin

**Mac:**

/Library/WebServer/Documents/phpMyAdmin

## mysql

**Mac:**

启动：

brew services start mysql

brew services stop mysql

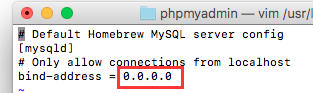
连接：

mysql -u root –p

之前出现mac机只有用localhost才能连

解决方案是：

修改配置文件/usr/local/etc/my.cnf如下：



**Windows:**

## apache

**Mac:**

配置：

/etc/apache2/httpd.conf

启动：

sudo apachectl restart

## Apache+PHP+MySQL

<https://blog.csdn.net/cgema/article/details/72457985>

**Apache:**

sudo apachectl start

sudo apachectl stop

sudo apachectl restart

**Php:**

**MySQL:**

**MAMP (集成安装工具 支持apachectl nginx)**

**修改apache配置：**

https://blog.csdn.net/tyyytcj/article/details/78371058?locationNum=10&fps=1

/etc/apache2/httpd.conf

## mac下的php php-fpm

To enable PHP in Apache add the following to httpd.conf and restart Apache:

LoadModule php7\_module /usr/local/opt/php/lib/httpd/modules/libphp7.so

<FilesMatch \.php$>

SetHandler application/x-httpd-php

</FilesMatch>

Finally, check DirectoryIndex includes index.php

DirectoryIndex index.php index.html

The php.ini and php-fpm.ini file can be found in:

/usr/local/etc/php/7.2/

To have launchd start php now and restart at login:

brew services start php

Or, if you don't want/need a background service you can just run:

php-fpm

==> Summary

/usr/local/Cellar/php/7.2.5: 515 files, 78.9MB

===============================================================================

[/usr/local/opt/php@7.2](mailto:/usr/local/opt/php@7.2)

[/usr/local/opt/php@7.2/sbin/**php-fpm**](mailto:/usr/local/opt/php@7.2/sbin/php-fpm)

php-fpm –t

/usr/local/etc/php/7.2/php-fpm.conf

## mac下的nginx

brew install nginx

nginx –t

/usr/local/etc/nginx/nginx.conf

**ps aux | grep nginx**

查找nginx进程

关闭：

**ps -ef|grep nginx**

www访问目录竟然在这里：

/usr/local/Cellar/nginx/1.13.12/html

<https://blog.csdn.net/beyond__devil/article/details/52711328>

<https://blog.csdn.net/beyond__devil/article/details/52711328>

EN.

<https://blog.frd.mn/install-nginx-php-fpm-mysql-and-phpmyadmin-on-os-x-mavericks-using-homebrew/>

http://blog.imqia.me/posts/osx-install-nmp/

brew install nginx

sudo nginx ( 运行)

brew install composer

brew install redis

brew install memcached

brew install autoconf

brew install phpmyadmin

To have launchd start redis now and restart at login:

brew services start redis

Or, if you don't want/need a background service you can just run:

redis-server /usr/local/etc/redis.conf

brew link --overwrite redis

## nginx下设置php-fpm使用socket文件的方法

<https://blog.csdn.net/u011630575/article/details/48287381>



/usr/local/etc/php/7.2/php-fpm.conf

## 部署

Filebeat（文件监控）==>LogStash==>elasticsearch

MetriBeat（性能监控)==> elasticsearch

rabbitmq (请求队列，缓存耗时请求)

Packetbeat (允许您监视应用级协议（如HTTP和MySQL）以及DNS和其他服务的实时网络流量

)

ELK就是指ElasticSearch (OK)、Logstash和Kiabana (OK)

Mac下php nginx 的安装

## 安装路径

filebeat /usr/local/Cellar/filebeat/6.2.4

\*\*\*\*\*大部分工具都在这个路径：\*\*\*\*\*

/usr/local/Cellar/

## 第三方应用

### Metricbeat

## 调用顺序

Routes：api.php

ap

Api.php

==>PackageController.php

==>IPackages.php

==>Packages.php

==>Model\Packages:xxx

==>(static::cache static::select )

Controller调用底层查询完以后 再 以json格式返回结果



## 排序和分页?



## 安装

tar zxvf redis-4.0.8

cd redis-4.0.8

make

src/redis-server

————————————————————————

Install brew;

/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"

Brew install mysql

mysql.server start or

brew services start mysql



—————————————————————————

brew install rabbitmq

brew services start rabbitmq

—————————————————————————

brew install logstash

brew services start logstash

—————————————————————————

brew install elasticsearch

brew services start elasticsearch

————————————————————————-

brew install Kibana

brew services start kibana

————————————————————————

brew install Grafana

brew services start grafana

phpMyAdmin

xxfpm "php-cgi.exe -c D:/Software/php7/php.ini" -n 2 -i 127.0.0.1 -p 9000

## php无后缀路径问题

比如要访问 gxpack.io/package

在根目录创建package🡺在package下放个index.php

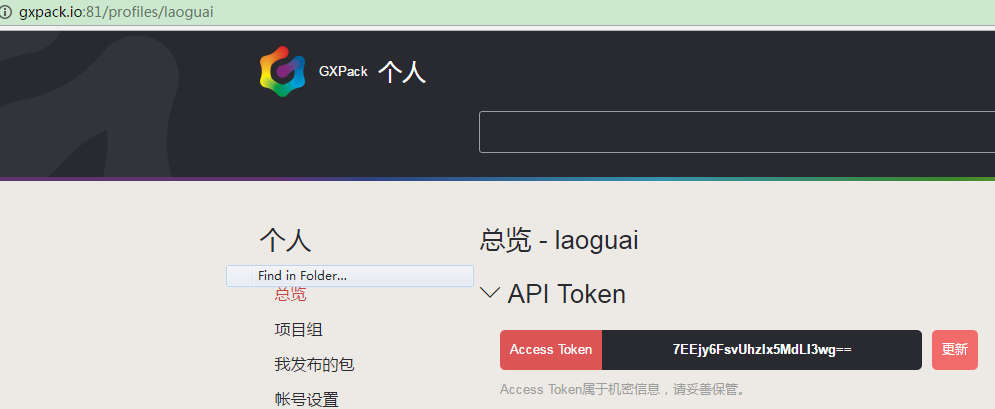
**提交新的远端项目配置**

## 第三方工具

Node.js

Composer Composer是PHP中用来管理依赖关系的工具

## Access Token



**7EEjy6FsvUhzIx5MdLI3wg==**

## URL

Token+**Project**

<http://api.gxpack.io:81/projects>

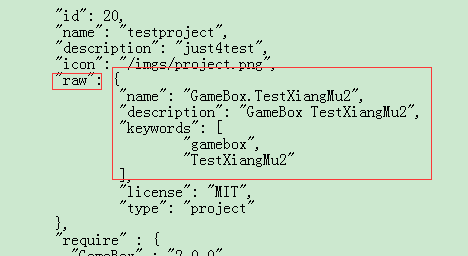
<http://api.gxpack.io:81/projects/20/>

<http://api.gxpack.io:81/projects/20/versions>

## 提交配置

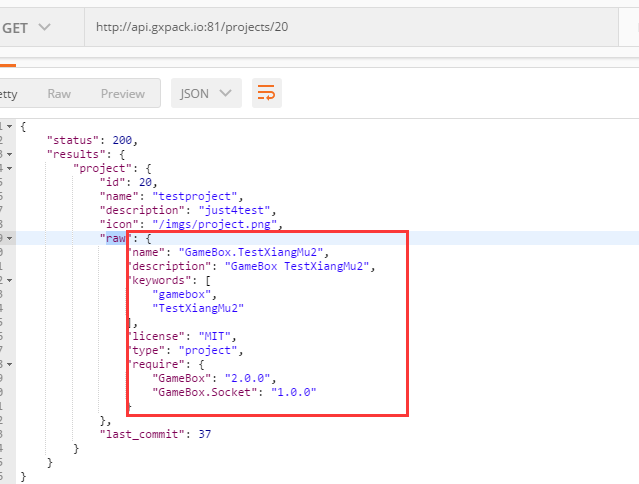
只能提交raw里面的





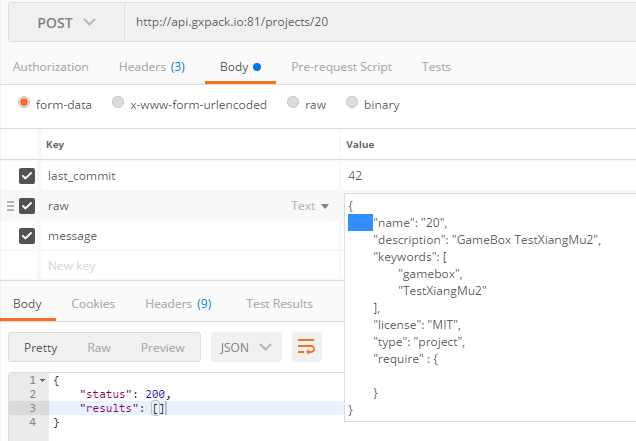
## 项目提交 (通过代码post提交)

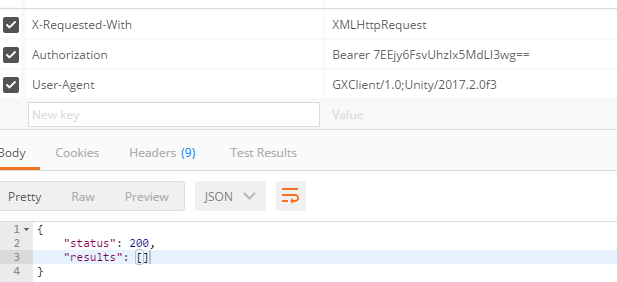
提交的是方框里的内容(raw内容)



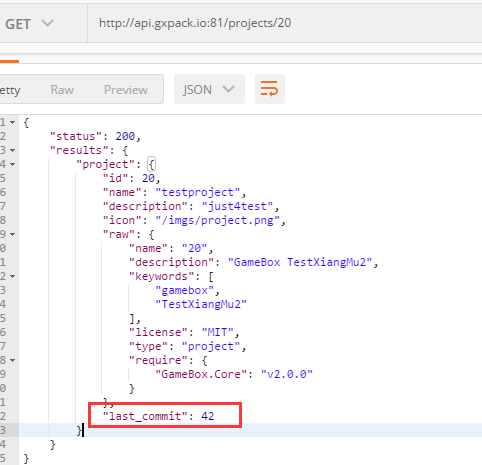
## Post请求

**https://blog.csdn.net/shifeilongsuper/article/details/50493943**





注意发送的 last\_commit必须用服务器上最新的ID：



## 抓包工具

**Fiddler**

**Charles (俞斌推荐)**

**正常的请求：**

----------------------------903847255762209719172741

Content-Disposition: form-data; name="last\_commit"

43

----------------------------903847255762209719172741

Content-Disposition: form-data; name="raw"

{

"name": "20",

"description": "GameBox TestXiangMu2",

"keywords": [

"gamebox",

"TestXiangMu2"

],

"license": "MIT",

"type": "project",

"require" : {

"GameBox.Core" : "v2.0.0",

"GameBox.Bucket" : "1.0.8"

}

}

----------------------------903847255762209719172741

Content-Disposition: form-data; name="message"

push\_by\_postman

----------------------------903847255762209719172741--

## 接口

### 包 <http://api.gxpack.io:81/packages/GameBox.Bucket>

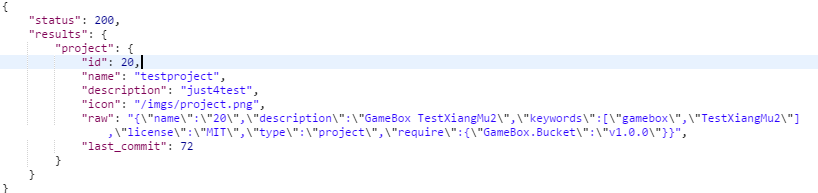


### 包2 <http://api.gxpack.io:81/packages/GameBox.Bucket/v1.0.2>



### Project <http://api.gxpack.io:81/projects/20>

GxProjectResult🡺GxProject



### Recommend <http://api.gxpack.io:81/recommend>

GxPacks🡺GxPackSummary



# ToDo

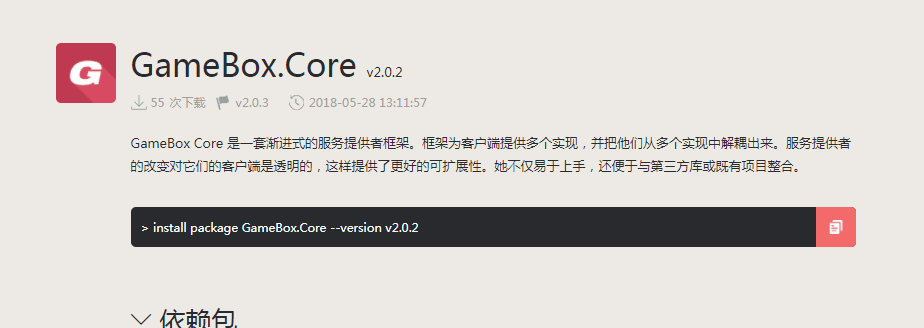
copy /Y "$(TargetDir)$(TargetName).dll" F:\@@@@@@@@@@@\GameBox\TestNewBucket\Assets\GameBox\Vendor\GameBox\TestBucket\net35\Editor

copy /Y "$(TargetDir)$(TargetName).pdb" F:\@@@@@@@@@@@\GameBox\TestNewBucket\Assets\GameBox\Vendor\\GameBox\TestBucket\net35\Editor

copy /Y "$(TargetDir)$(TargetName).dll" F:\@@@@@@@@@@@\GameBox\TestNewBucket\Assets\GameBox\Vendor\\GameBox\TestBucket\net35\Editor

copy /Y "$(TargetDir)$(TargetName).pdb" F:\@@@@@@@@@@@\GameBox\TestNewBucket\Assets\GameBox\Vendor\\GameBox\TestBucket\net35\Editor

## 客户端可以通过指令安装包



## ~2.0.0 ~1.0.0

处理

## Download gitlab处理

## Bucket合成一个动态链接库后自更新如何替换

Simplify & [self-confident](http://www.baidu.com/link?url=J6ISRBRPSPxNJk5BJlniyC7n-aIzcyXTJPpulmKKBiU_f3TiLMbUawRv_Cnq9bg_aSEPyvJRMyxq0-NtpMIiVrzfUHA7DIdsjkuvdKnwoi6WMqTx5uYnqHwlnoXVHzNP)

## 版本号

 null?

SemVer >=2.5.0

var r1 = new Range("1.0.1");

var r2 = new Range("2.0.0");

<0.0.0

-------------------------------------------------

var r1 = new Range("1.0.1");

var r2 = new Range("1.0.2");

<0.0.0

-------------------------------------------------

var r1 = new Range(">1.0.1");

var r2 = new Range(">1.0.2");

>1.0.2

-------------------------------------------------

var r1 = new Range(">v1.0.1");

var r2 = new Range(">v1.0.2");

>1.0.2

-------------------------------------------------

var r1 = new Range(">vv1.0.1");

var r2 = new Range(">vv1.0.2");

>1.0.2

## 启动包和项目依赖包下载

## 整体下载流程

# task

## 20180528-20180601

1解决依赖包下载问题

2权限处理

3 连续安装删除问题

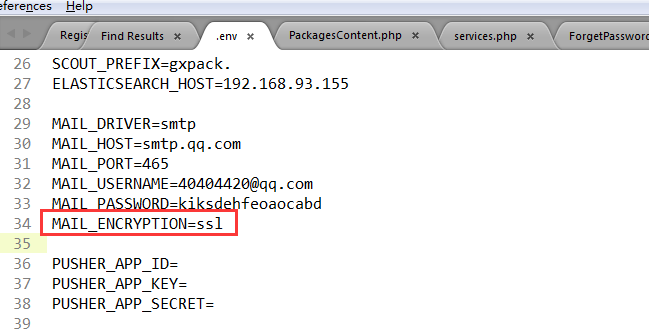
4 服务器无法邀请问题

5 服务器返回profile字段不一致问题

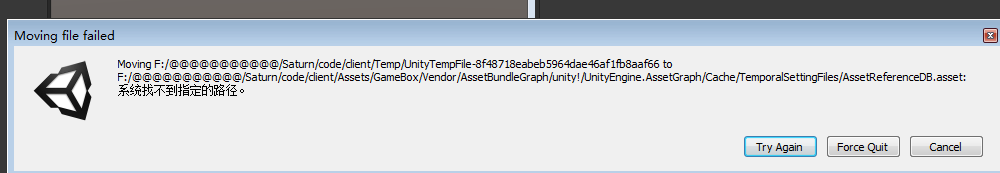
6 GXPACK交接

# Q&A

## [laravel Expected response code 250 but got code “530”](https://stackoverflow.com/questions/42574375/laravel-expected-response-code-250-but-got-code-530)



 replace your MAIL\_ENCRYPTION = null to MAIL\_ENCRYPTION = ssl



删除

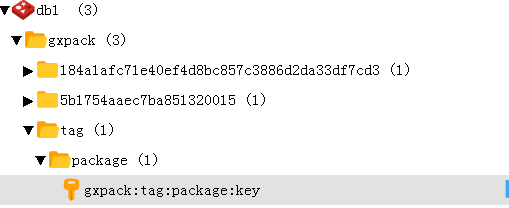
Softdelete

## Cache:tags

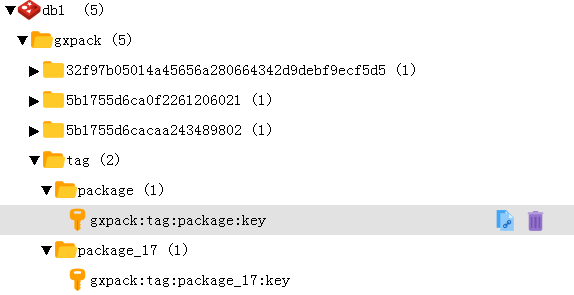
public static function getPackageContent($packageId, $version = null)

return Cache::tags(['package\_content'])->remember("content:$packageId:".(is\_null($version) ? 'last' : $version), $expiresAt









[a,b] a b是并列关系，就是把一个组同时打上a标签和b标签

Cache::tags(['package', 'package\_17'])->put('id:17', 'haha', $expiresAt);

Cache::tags(['package', 'package\_18'])->put('id:18', 'haha', $expiresAt);

flush()可以只带一个标签

Cache::tags('package\_18')->flush();

Has或者其它操作必须带全部标签：Cache::tags(['package', 'package\_18'])

(a,b,c)

Cache::tags('package', 'package\_17')->put('id:17', 'haha', $expiresAt);

Cache::tags('package', 'package\_18')->put('id:18', 'haha', $expiresAt);

用()而不是[]flush操作很诡异，比如Cache::tags('package', 'package\_17')->flush

会把Cache::tags('package', 'package\_18')也删了， 就是flush必须带根删除，也就是全部删除，不像[]那样可以部分删除，判断操作比如has也只判断第一个参数package,后面加多少参数都不影响判断。。。, \*\*\*\*\*最终发现，只有第一个参数’package’有意义，所以这种方式没用应用的必要\*\*\*\*\* 压根就是错误的写法

测试代码：

$expiresAt = Carbon::now()->addMinutes(60 \* 24 \* 7);

Cache::tags('package', 'package\_17')->put('id:17', 'haha', $expiresAt);

Cache::tags('package', 'package\_18')->put('id:18', 'haha', $expiresAt);

Cache::tags('package', 'package\_17')->forget('id:17');

if (Cache::tags('package', 'package\_18')->has('id:18'))

{

echo "has key has key has key111\n";

}else

{

echo "no key no key no key111\n";

}

if (Cache::tags('package', 'package\_17')->has('id:17'))

{

echo "has key has key has key222\n";

}else

{

echo "no key no key no key222\n";

}

// Cache::tags(['package', 'package\_17'])->put('id:17', 'haha', $expiresAt);

// Cache::tags(['package', 'package\_18'])->put('id:18', 'haha', $expiresAt);

// //Cache::tags(['package', 'package\_17'])->flush();

// if (Cache::tags(['package', 'package\_18'])->has('id:18'))

// {

// echo "has key has key has key111\n";

// }else

// {

// echo "no key no key no key111\n";

// }

// if (Cache::tags(['package', 'package\_17'])->has('id:17'))

// {

// echo "has key has key has key222\n";

// }else

// {

// echo "no key no key no key222\n";

// }

## 注释掉代码，执行删除tag的操作，等恢复代码后这个操作竟然自动执行了

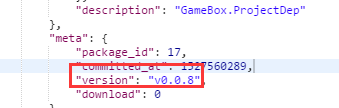


所以webhook要用post测试，操作要快，恢复日志代码立刻postman操作, 完了会被队列；里的命令执行操作？？。。。

* 添加删除这里都有问题：



* 删除这里有问题：



上面两个都是缓存同步的问题，数据库已经没有问题

## 无法加入项目组

ProjectsController.php.

ProjectTrait.php. joinInProjectWithInviteToken

## 发布新标签webhook不起作用

原因是Gitlab访问不了域名， 解决方案是webhook地址填ip,

然后在nginx配置里 配置代理proxy\_pass 又跳转成域名访问，（之前项目就是设置成只有指定域名才能访问）

## 数据库删除组件没效果

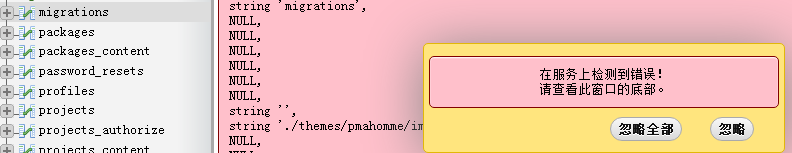
清理redis缓存

redis-cli

auth ztgame@123

flushall

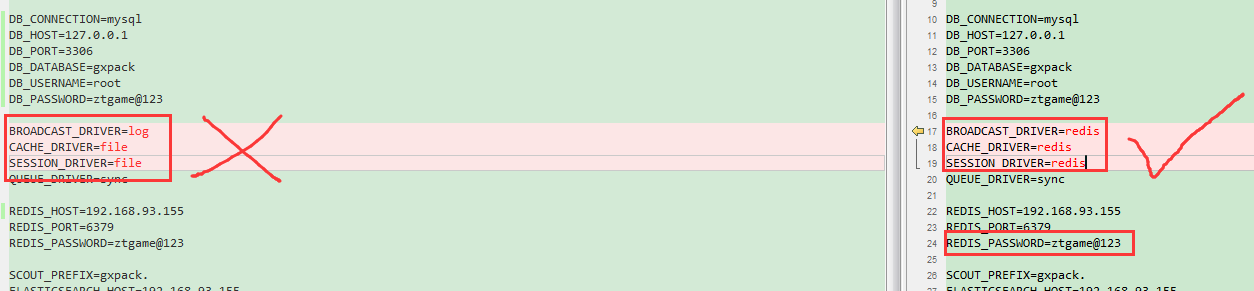
## 访问phpmyadmin老是弹警告界面



把php从7.2换成7.1就好了

## 添加包出错

\*\*\*\*\*最终的问题竟然出在这里：



\*\*\*\*\*最终的问题出在.env的redis配置上



\*\*\*\*\*最终的问题出在.env的redis配置上



\*\*\*\*\*最终的问题出在.env的redis配置上



Gxpack目录

npm install

compose install

cd /d e:

elasticsearch 还没加

之前存在的问题：

### redis

用brew 起的redis有问题，杀掉后用sudo redis-server /etc/redis.conf直接启

配置修改：(/etc/redis.conf)

密码：



运行非本机访问：



### rabbitmq

d:\Program Files\RabbitMQ Server\rabbitmq\_server-3.7.5\sbin>rabbitmq-plugins ena

ble rabbitmq\_management

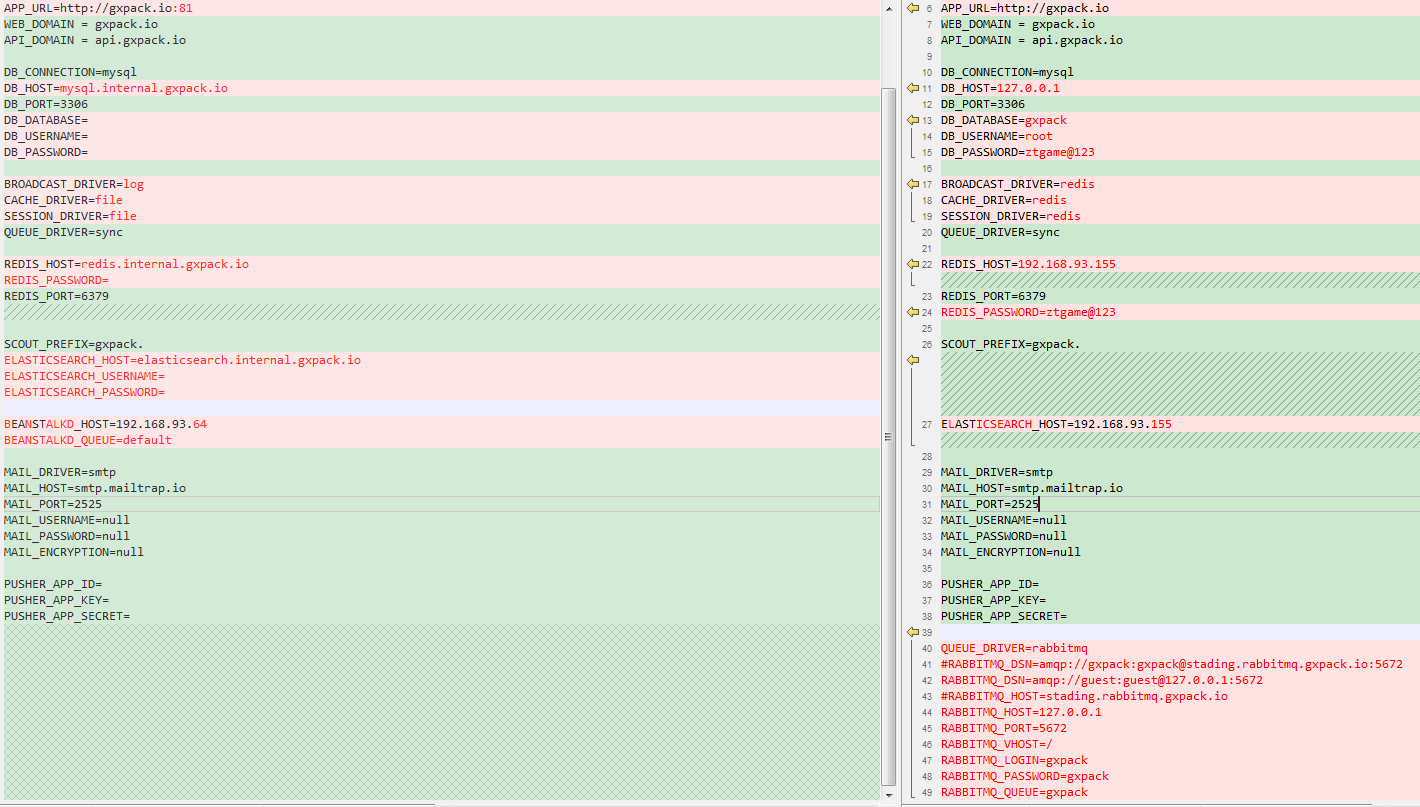
<http://127.0.0.1:15672/#/users>

gxpack:gxpack

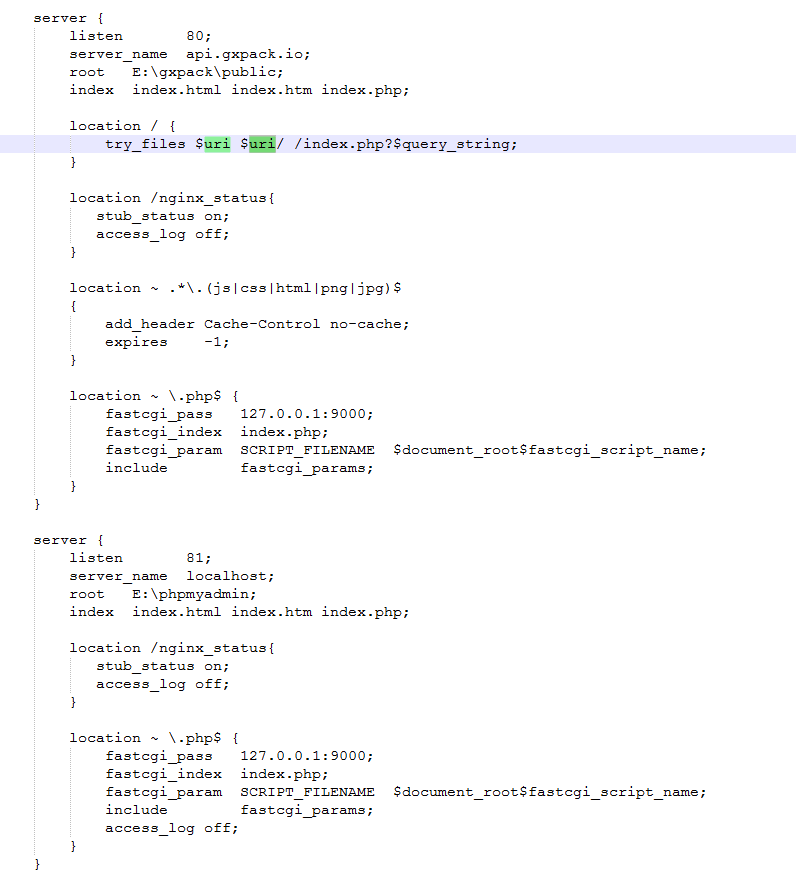
### phpmyadmin

<http://127.0.0.1:81/index.php>?

### .env



### nginx.conf



## 发布包出错

php artisan queue:listen

## 通过82端口访问phpmyadmin会造成ningx挂掉？

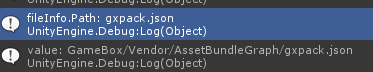
通过127.0.0.1:82可以正常访问

Nginx配置：

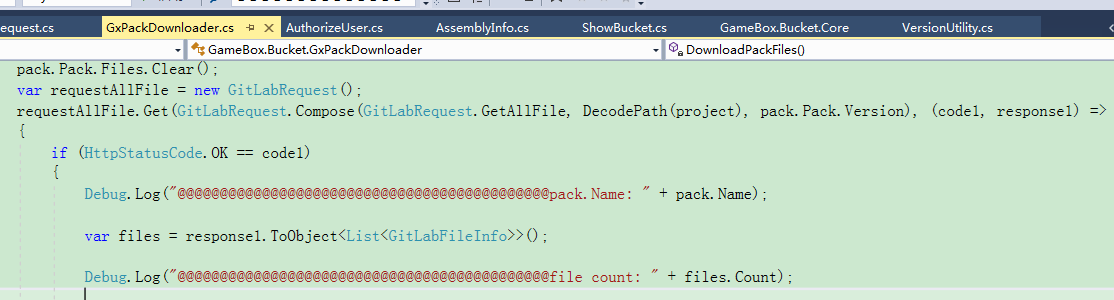


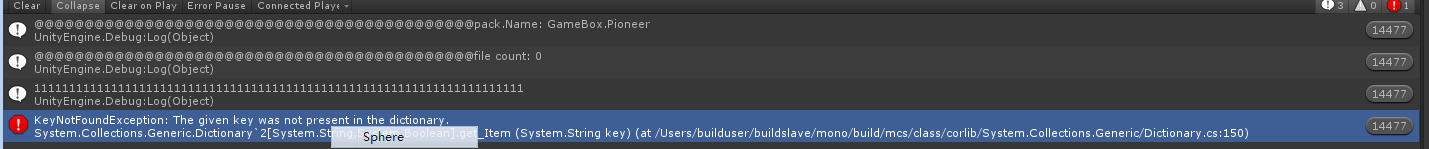
## Tommorow

http://gitlab.ztgame.com/api/v4/projects/tech%2FClient%2FGameBox.ThirdLib%2FReleases%2FAssetBundleGraph/repository/tree?&ref=v1.4.1&path=unity!/UnityEngine.AssetGraph/Editor/GUI/GraphicResources



## Gitlab包空文件无限请求







**上面函数里面报错竟然导致 整个代码段无限执行。。。**

**原因是出错的时候没调用：**



## Gitlab文件显示不全

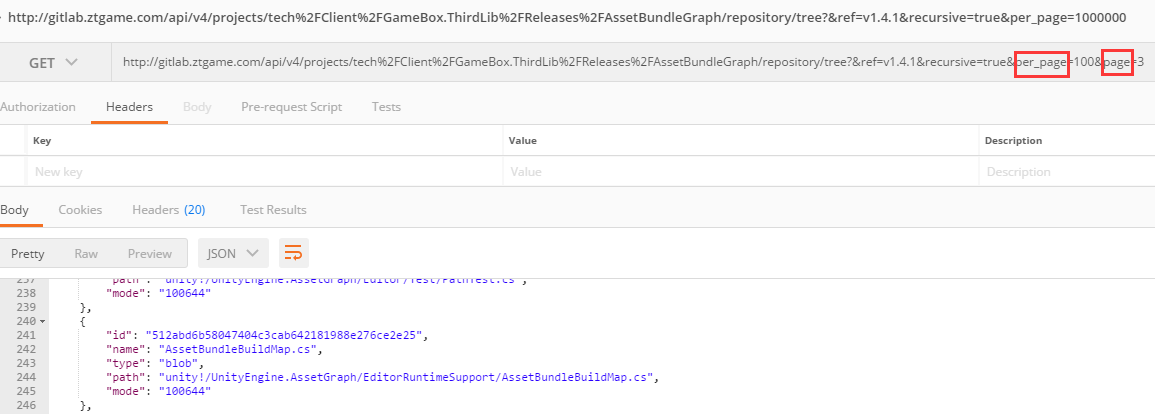
<http://gitlab.ztgame.com/api/v4/projects/tech%2FClient%2FGameBox.ThirdLib%2FReleases%2FAssetBundleGraph/repository/tree?&ref=v1.4.1&recursive=true&per_page=1000000>

<https://docs.gitlab.com/ce/api/#pagination>

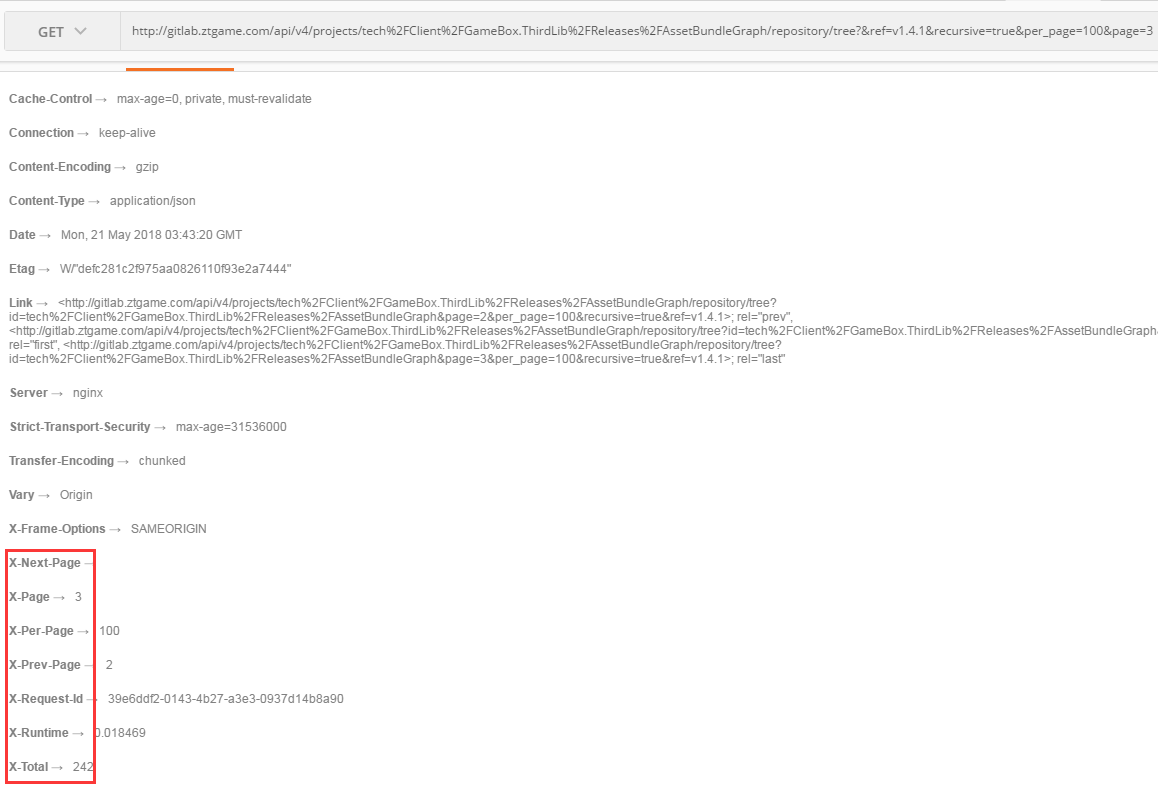
参数：

per\_page

page

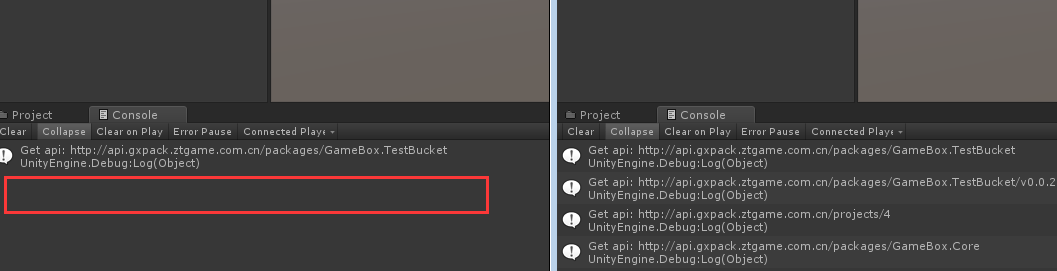


记录信息：



## 发布用的git仓库必须是public (不能private…)

## 一个客户端请求组件没返回。。。



NewtonJson库必须放在同一个目录。。。。晕

## Mac下elasticsearch.yml配置文件在哪

brew info elasticsearch:

Data: /usr/local/var/lib/elasticsearch/elasticsearch\_jenkins/

Logs: /usr/local/var/log/elasticsearch/elasticsearch\_jenkins.log

Plugins: /usr/local/var/elasticsearch/plugins/

Config: /usr/local/etc/elasticsearch/

To have launchd start elasticsearch now and restart at login:

brew services start elasticsearch

Or, if you don't want/need a background service you can just run:

elasticsearch

## Mac下运行phpmyadmin（nginx）登录失败



改了phpmyadmin的权限

还有改了nginx端口以及root路径就可以了

（后来又把nginx端口以及root改回之前有问题的时候，竟然还是正常访问了，真神奇…）

## Mac下运行phpmyadmin（apache）登录失败

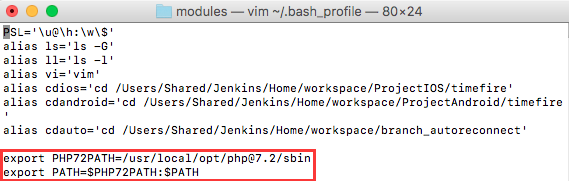


原本是起的php-fpm版本过低 (或者压根没启用)

解决：

新版php-fpm路径在这[/usr/local/opt/php@7.2/sbin/**php-fpm**](mailto:/usr/local/opt/php@7.2/sbin/php-fpm)

把这路径加入环境变量：



后启动php-fpm

php-fpm 或者

brew services start php

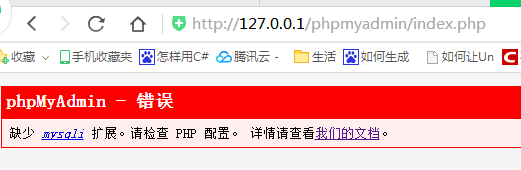
## Windows php运行

E:\downoad\gameboxrelated\php-7.2.4-Win32-VC15-x64>xxfpm "php-cgi.exe -c E:/downoad/gameboxrelated/php-7.2.4-Win32-VC15-

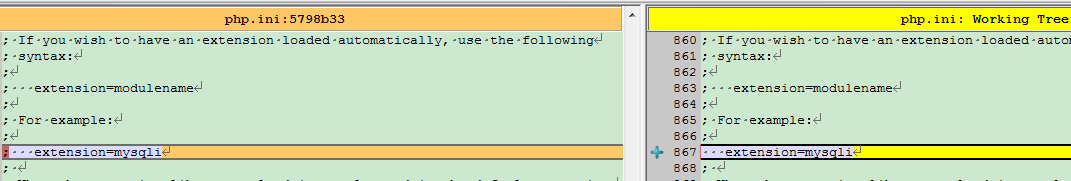
x64/php.ini" -n 2 -i 127.0.0.1 -p 9000



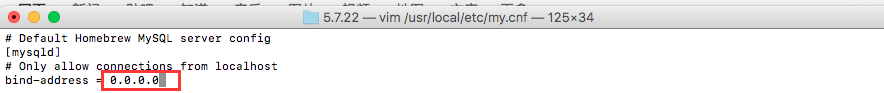
## Windows下访问phpmyadmin 提示缺少mysqli如下图：



按如下改配置OK：



## Mac下只能通过127.0.0.1访问mysql



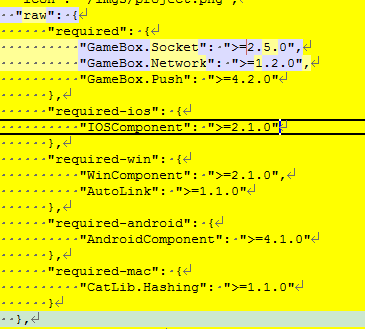
Bind-address改成0.0.0.0 OK

 show variables like '%data%'

## Gitlab返回的版本号排序有问题

比如 1.2.2会排在1.12.2的前面

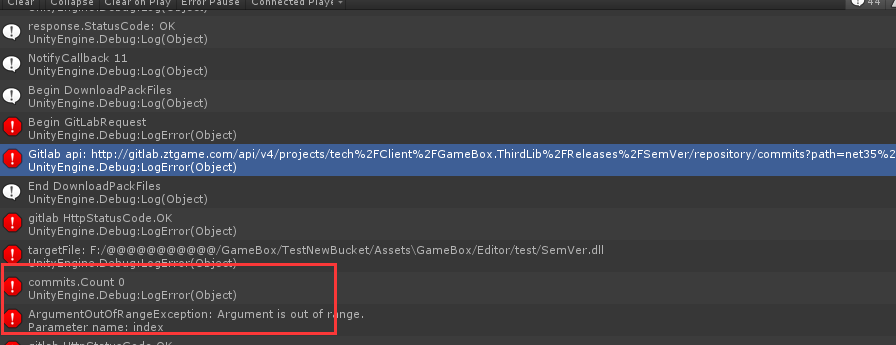
## 不同平台不同版本的依赖包处理



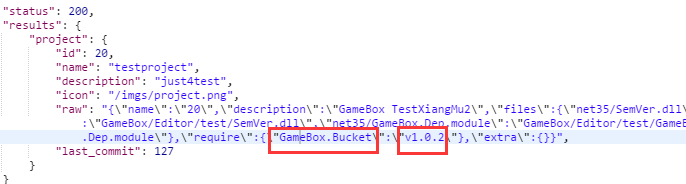
## Mac下mysql不能ip访问，只能通过localhost访问

## Phpmyadmin运行提示mysql报错

## 下载报错如下图：



原因是依赖的GameBox.Bucket：需要v1.0.2, 但组件包本身的文件只有v1.0.0的 （没有v1.0.2这个tag）



copy /Y "$(TargetPath)" "$(TargetDir)$(TargetName).dll"

copy /Y "$(TargetDir)$(TargetName).dll" F:\@@@@@@@@@@@\GameBox\TestNewBucket\Assets\Editor

copy /Y "$(TargetDir)$(TargetName).dll" F:\@@@@@@@@@@@\GameBox\TestNewBucket\Assets\GameBox\Editor\GameBox.Bucket

## 安装卸载信息更新不及时

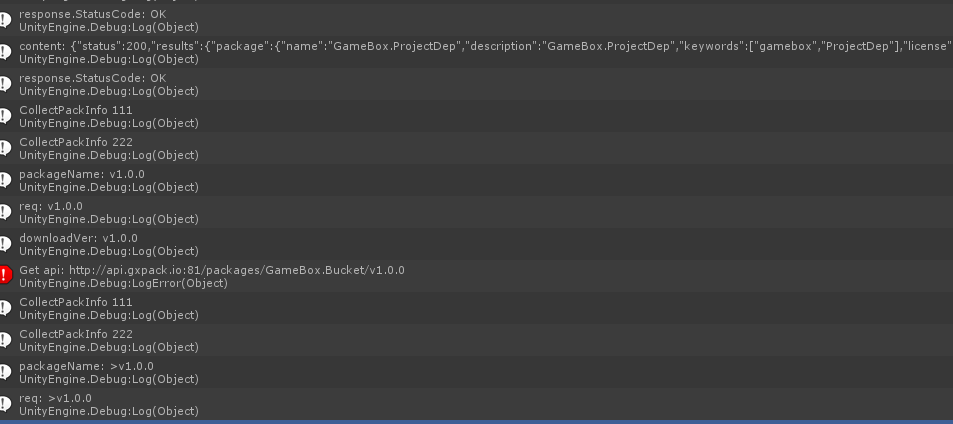
两个工程合成一个就可以了

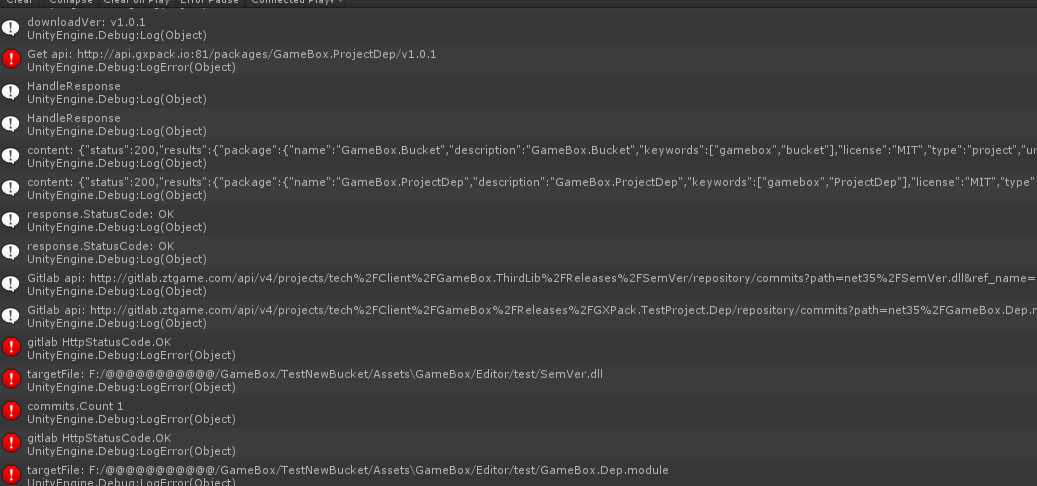
## 没下载完

## 界面不显示了

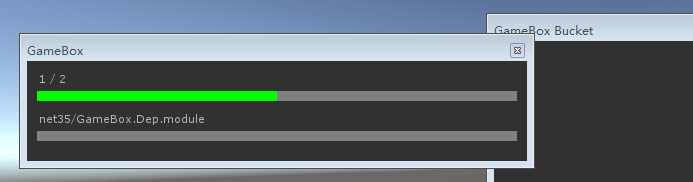
OnUpdate 执行到中途

OnGUI 压根不执行了

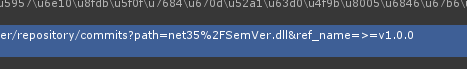




## 真正下载文件可能进行到一半卡主？

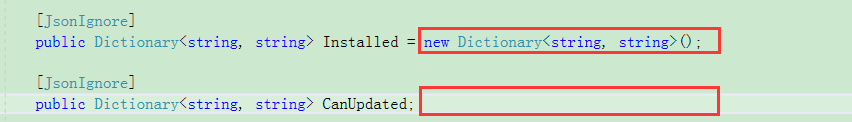


## TryUpdateMetaFile(commits[0], targetFile + ".json", out metafile 报错



ref\_name有问题

## 字典操作各种报错



初始化。。。

## Dic traverse

<https://www.cnblogs.com/wangchao928/archive/2012/05/17/2505671.html>



## 实际下载的地址





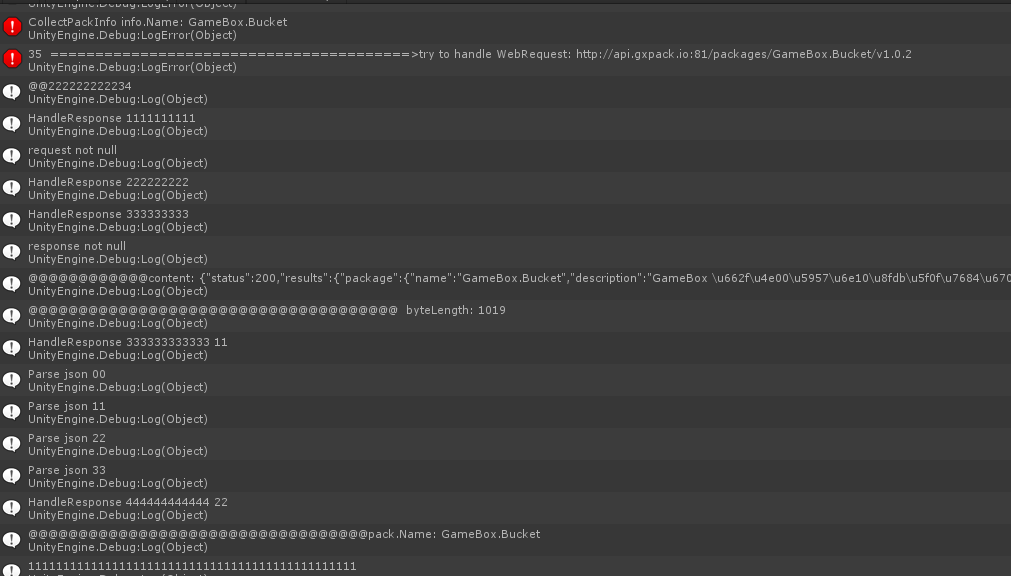
红框就是tag(版本号)

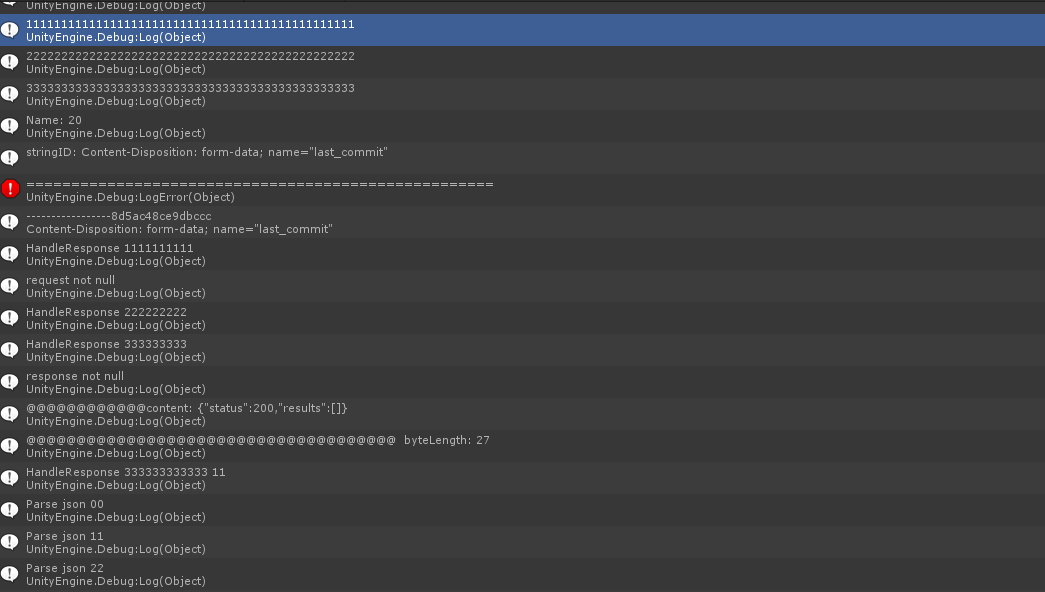
## 在资源管理器删除下载的DLL，

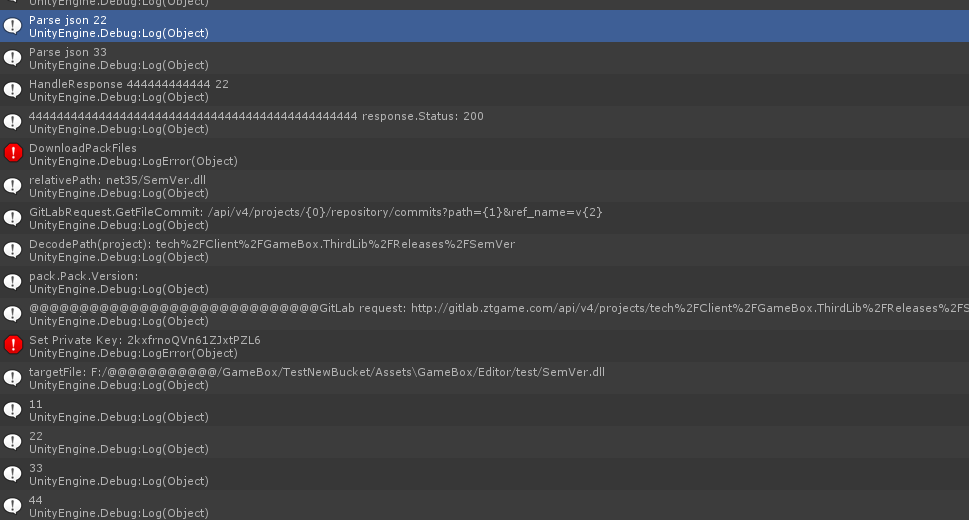
## 卸载再安装不成功，重新打开安装成龙

安装完直接卸载也不成功

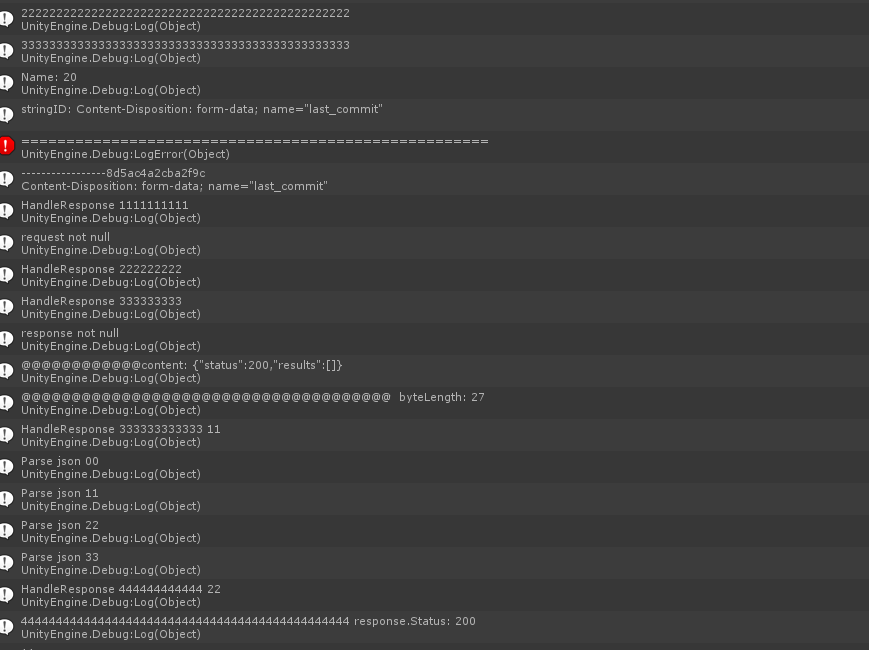
\*\*\*\*\*原因是last\_commit没有更新



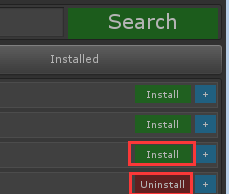




## 正常卸载



## Install、Uninstall区分





推荐的如果是这个项目需要的则显示 Uninstall ()

不是需要的则显示 Install

1什么时候安装呢？

2点Install Uninstall又做如何处理

## Projects只能用ID来访问吗？

## Json读取问题

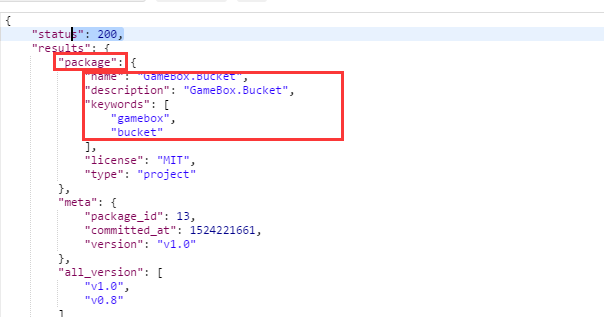
info.Pack = response.ToObject<GxPackResult>("results").Package **不能用**

**可用：**

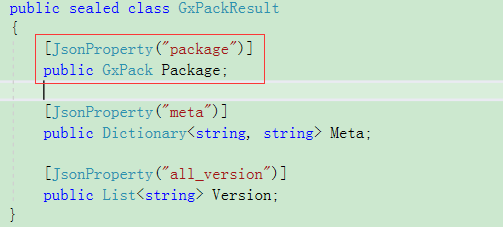
var packResult = response.ToObject<GxPackResult>("results");

info.Pack = packResult.Package;

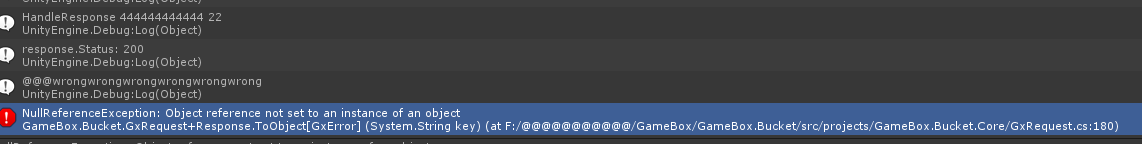
## Json多重嵌套如何读取



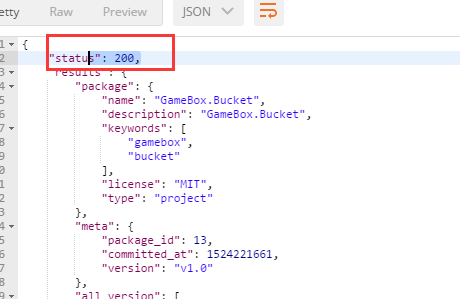
子字段加类：



## 用正式的GXPack出错

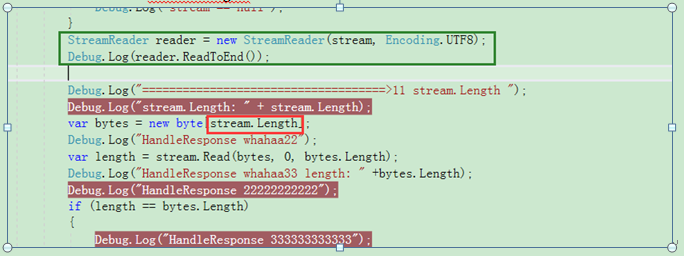


原因是配置文件用的status:200 按出错来处理:



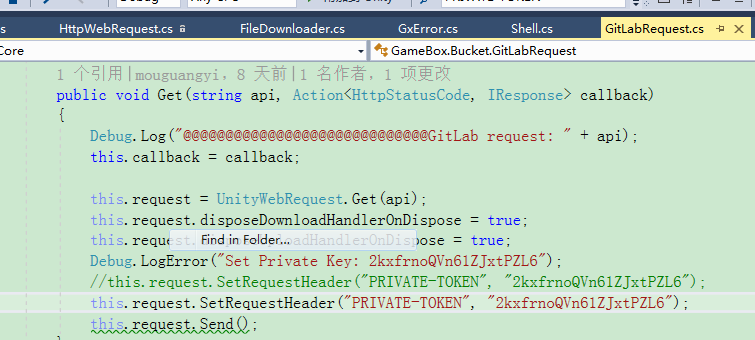
## Httpwebrequest访问返回没有数据流

实际上有流，只是Stream.length不能用



## 下载gitlab上的文件总是返回首页的html页

GitLabRequest 有个设置PrivateKey



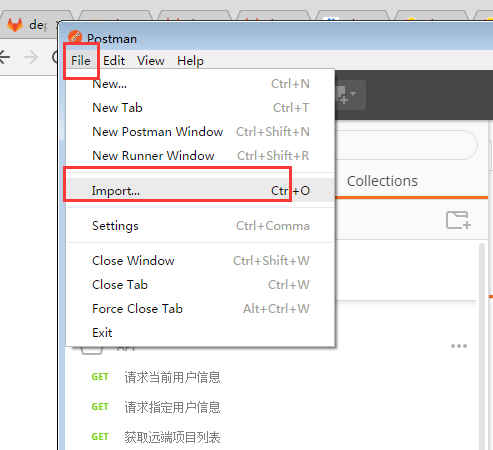
在下载类里也要加访问token:

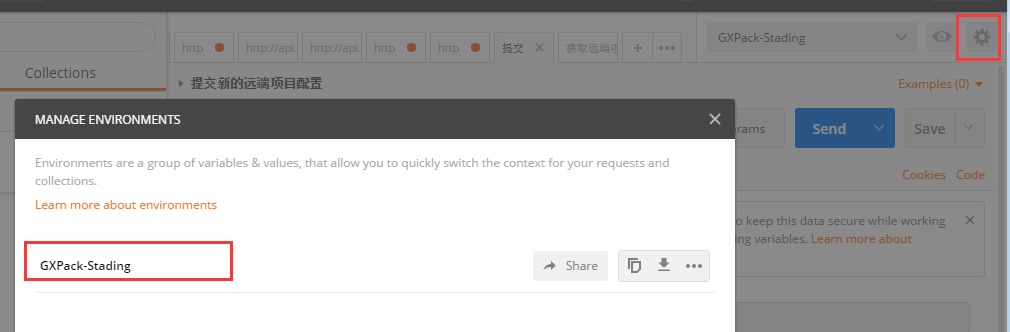


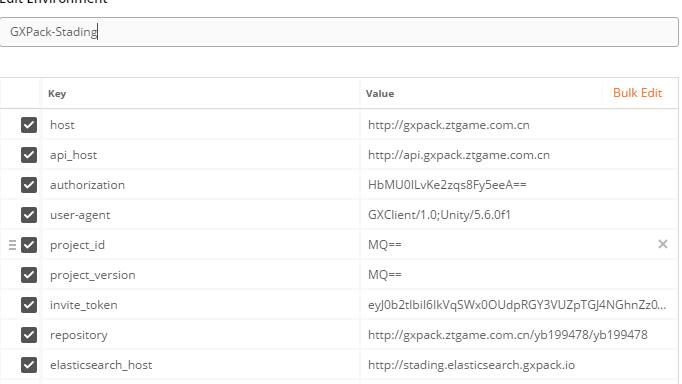
## mock环境不能正常弹出界面

下载的json解析出错了

# 俞斌







**确保对外暴露没有泛型**

## 走前交接

1 192上的Rabbitmq是没有守护进程的，必须手动重启

2 必须启动192 193上的supervisor (sudo supervisor start), supervisor会自动启动其它服务

3 192.168.23.193（弱鸡） 192.168.23.192 用户名 user 密码ztgame@123

4 macmini(现在ip 192.168.93.64, 如ip有变需要在自己host上做更改)上有三种服务断电后要启动(nginx(手动) php-fpm(手动) walle(不用管))

5 192 193所有服务都安装在/home/vdisk上面

**要关注的指标**：

1 \*\*\*\*\*\* failedjob < 1 (异常会打埋点，1分钟没掉下去会堆mysql)

2 Exception 不能有^

3 Response Code不能有502 (502指ftp不够了)

4 QPM < 1万

**代码：**

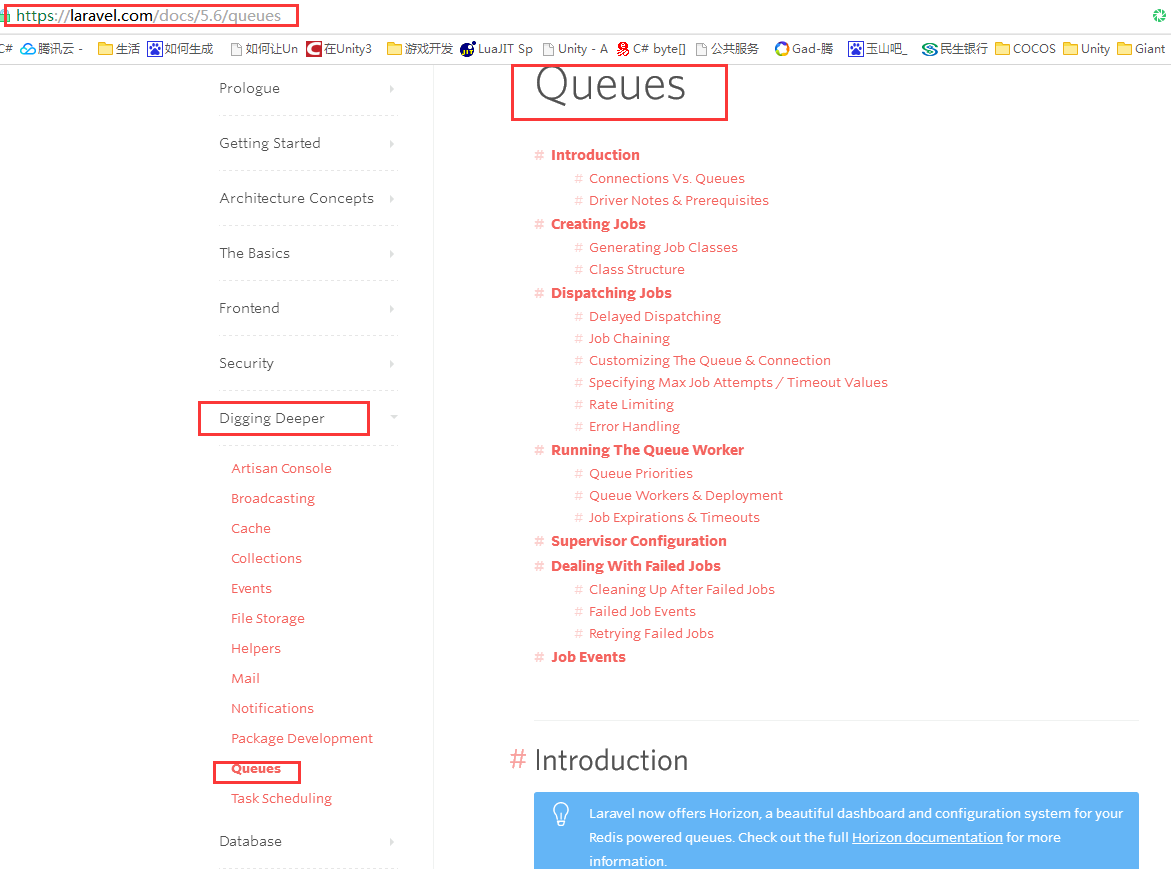
Mode目录下的代码基本上都有缓存(具体要看代码) ,要针对具体问题清理缓存(redis),（全部清理有可能击穿数据库）,一般缓存7天才失效

同步模式测完要异步模式测试，请求访问者的信息这些信息在异步都是没有的

.env

QUEUE\_DRIVER=sync 🡺

QUEUE\_DRIVER=redis 或者rabbitmq



queue启动：

php artisan queue:listen --queue=gxpack-import,gxpack-download,gxpack-logs-exception,gxpack-logs-access,gxpack-buried,gxpack

开启上面的指令才会消费队列

6月中旬以后可以提问

## 剩下的问题

192.168.93.64 walle.gxpack.io

1无法加入项目

2 其它用户登录图片资源不显示

## json编码解析

http://www.bejson.com/

## Apache ab压力测试

**YiBKOjpDMISIWWf2TG7ylA==**

**YiBKOjpDMISIWWf2TG7ylA==**

## 资源修复？

## 下载的带点号的包名不能按其名字建一个文件，必须按点号隔开建立多个文件夹

比如包名gamebox.bucket 建立gamebox/bucket的层级

## gitlab版本号排的有问题

1.2.2 会排在1.12.3前面, 要注意这个坑

## download rule gitlab文件夹下载规则

net35 net46 (FrameWork层) netstandard20 unity53 unity56 unity2017.2 （应用层）

应用层优先、选取最接近自己的版本（选取的版本不能跨大版本）,如何没有合适的版本就从net35 net46里选



## 包依赖包的位置

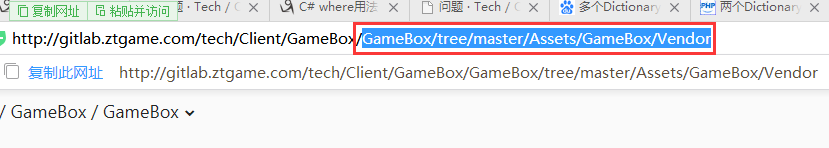
Package :

{

Required:{}

}

## 下载输出规则



## 请求日志

## Phpcomposer

http://docs.phpcomposer.com/01-basic-usage.html#Package-Versions

## symfony

## phpMyAdmin

## Grafana

## kibana

Kibina garabana grabana grafana

## form-data

## WebHook接口调试

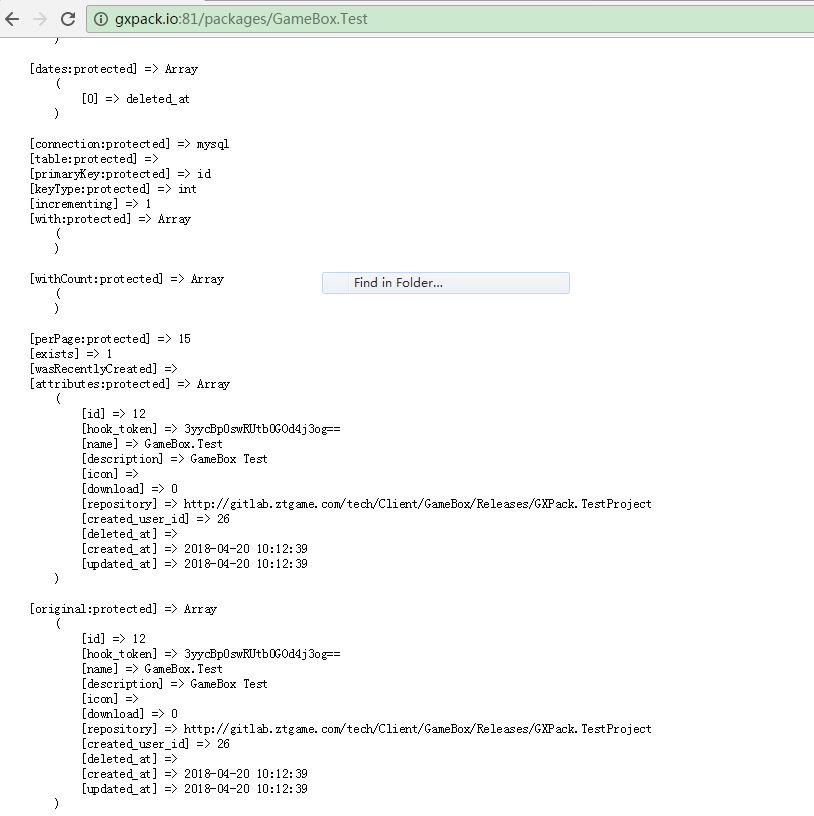
用Postman+ 三个参数 （其中一个是[hook\_token] => OW2C6uTx4KpfEMjSguw==

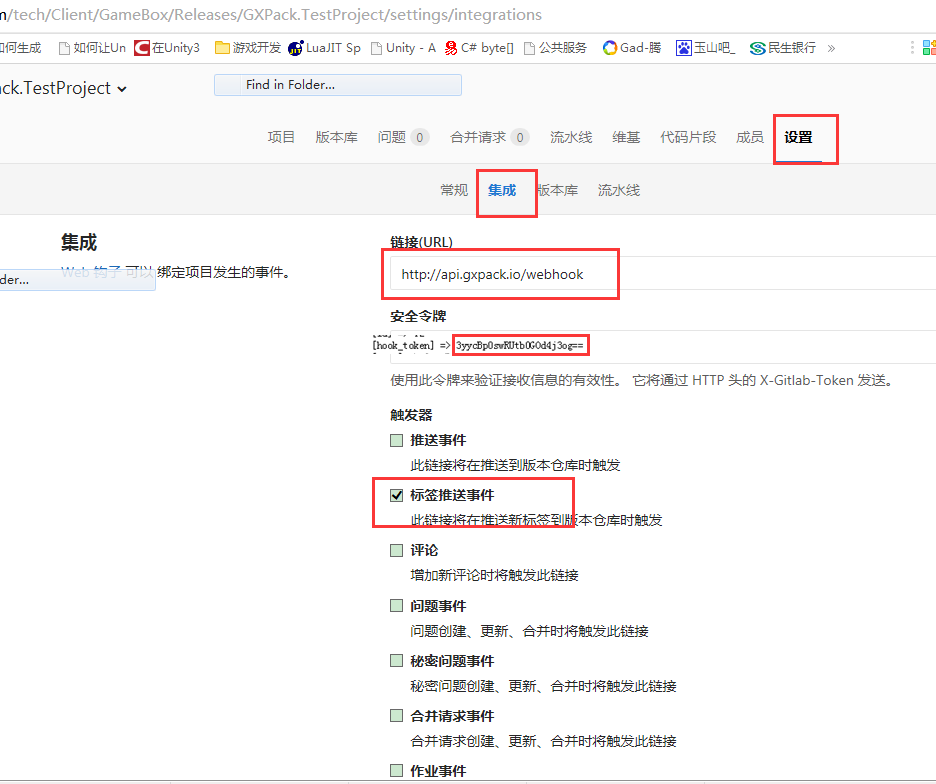
）

## Status Http状态码

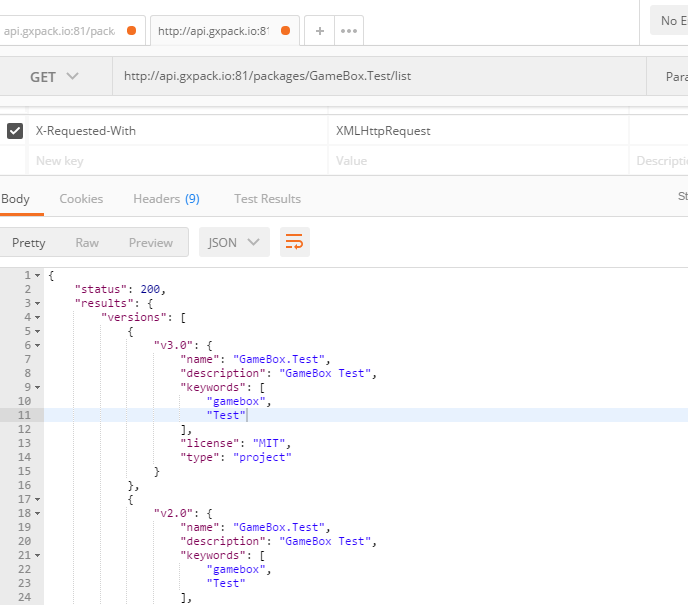
## Postman

## 设置token webhook





## 测试



内容相似度

# 王辉

