# GoWeb

## Request解析

r \*http.Request

r.ParseForm() //解析

输入：

<http://localhost:9090/list?url_long=111&url_long=222&p1=1&p2=2&p3=3>

### r.Form

输出：

map[url\_long:[111 222] p1:[1] p2:[2] p3:[3]]

### r.URL.Path

/list

### r.URL.Scheme

### r.Method

GET

POST

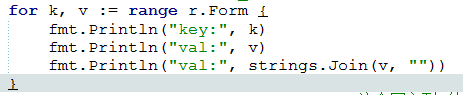
### 打印输入参数

fmt.Println(**r.Form["url\_long"]**)

fmt.Println(**r.Form["p1"]**)

r.**FormValue**("username") //这个不需要提前调用r.ParseForm()

### 遍历输入参数



输出：

key: url\_long

val: [111 222]

val: 111222

key: p1

val: [1]

val: 1

key: p2

val: [2]

val: 2

key: p3

val: [3]

val: 3

## 表单

### 下拉菜单

<select name="fruit">

<option value="apple">apple</option>

<option value="pear">pear</option>

<option value="banane">banane</option>

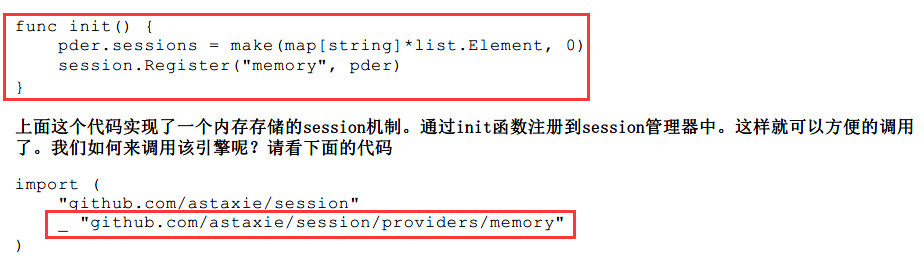
</select>

### 单选按钮

<input type="radio" name="gender" value="1">男

<input type="radio" name="gender" value="2">女

## session



### 预防session劫持



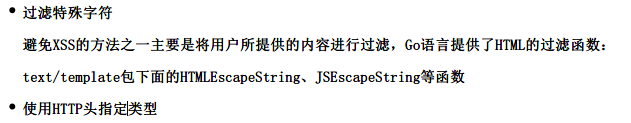
**上面两个手段的组合可以在实践中消除session劫持的风险，一方面，由于sessionID频繁改变，使攻击者难有机会获取有效的sessionID；另一方面，因为sessionID只能在cookie中传递，然后设置了httponly，所以基于URL攻击的可能性为零，同时被XSS获取sessionID也不可能。最后，由于我们还设置了MaxAge=0，这样就相当于session cookie不会留在浏览器的历史记录里面。**

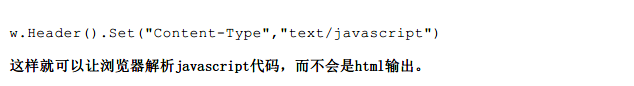
## 安全与加密

### 预防CSRF攻击 (跨站请求伪造)

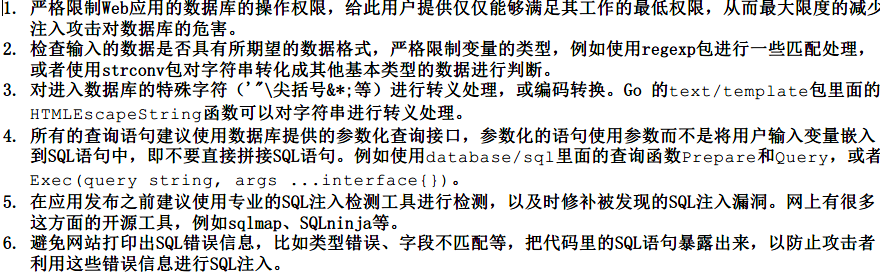


### 避免XSS攻击 (跨站脚本攻击)





### 避免SQL注入



## json

b := []byte(`{"Name":"Wednesday","Age":6,"Parents":["Gomez","Morticia"]}`)

如果在我们不知道他的结构的情况下，我们把他解析到interface{}里面

var f interface{}

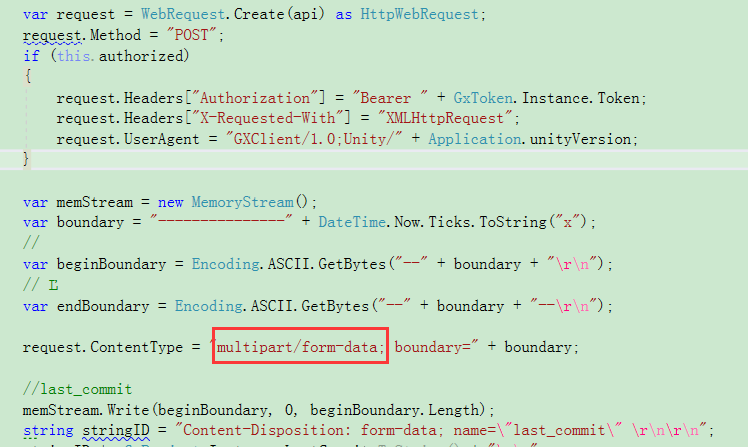
err := json.Unmarshal(b, &f)

## beego

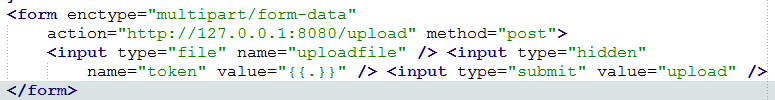
<https://github.com/astaxie/beego>

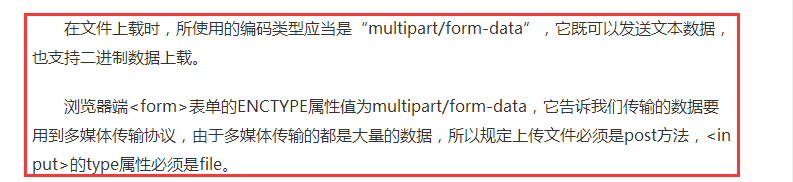
## 表单提交 （POST）

### GXPack 也用的multipart/form-data



### Multipart/form-data

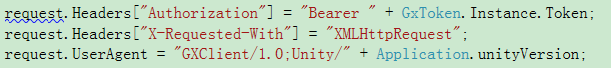




### application/x-www-form-urlencoded (form.enctype的缺省值？)

## Get

### GXPack



### Iris

## Beego

<https://beego.me/quickstart>

https://github.com/astaxie/beego

## Iris

API:

app.RegisterView(iris.HTML("./views", ".html").Reload(true))

app.OnErrorCode(iris.StatusInternalServerError, func(ctx iris.Context) {

// .Values are used to communicate between handlers, middleware.

errMessage := ctx.Values().GetString("error")

if errMessage != "" {

ctx.Writef("Internal server error: %s", errMessage)

return

}

ctx.Writef("(Unexpected) internal server error")

})

### app.Post("/upload", func(ctx iris.Context) {

app.Post("/decode", func(ctx iris.Context) {

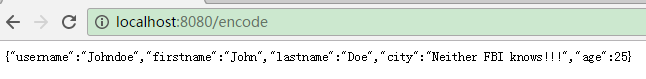
var user User

ctx.ReadJSON(&user)

ctx.Writef("%s %s is %d years old and comes from %s", user.Firstname, user.Lastname, user.Age, user.City)

})

### Json输出



app.Get("/encode", **func**(ctx iris.Context) {

doe := User{

Username: "Johndoe",

Firstname: "John",

Lastname: "Doe",

City: "Neither FBI knows!!!",

Age: 25,

}

ctx.JSON(doe)

})

# Go

## Go语法

make(map[string]Provide)

make([]byte, 32)

### defer

defer定义的语句会在最后执行。

使用场合：

及时释放资源

e.g.

pder.lock.Lock()

defer pder.lock.Unlock()

### map

//interface{} key

//interface{} value

v := make(map[interface{}]interface{}, 0)

### struct

type Server struct {

ServerName string

ServerIP string

}

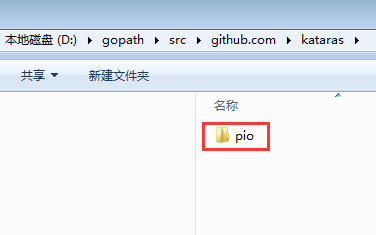
## 路径



## Go Get

go get github.com/kataras/pio

路径：



# 最好的6个Go语言Web框架

<https://blog.csdn.net/dev_csdn/article/details/78740990>

Beego: 一个Go语言下开源的，高性能Web框架   
\* <https://github.com/astaxie/beego>   
\* [https://beego.me](https://beego.me/)

Buffalo: 一个Go语言下快速Web开发框架   
\* <https://github.com/gobuffalo/buffalo>   
\* [https://gobuffalo.io](https://gobuffalo.io/)

Echo: 一个高性能，极简的Web框架   
\* <https://github.com/labstack/echo>   
\* [https://echo.labstack.com](https://echo.labstack.com/)

Gin: 一个Go语言写的HTTP Web框架。它提供了Martini风格的API并有更好的性能。   
\* <https://github.com/gin-gonic/gin>   
\* <https://gin-gonic.github.io/gin>

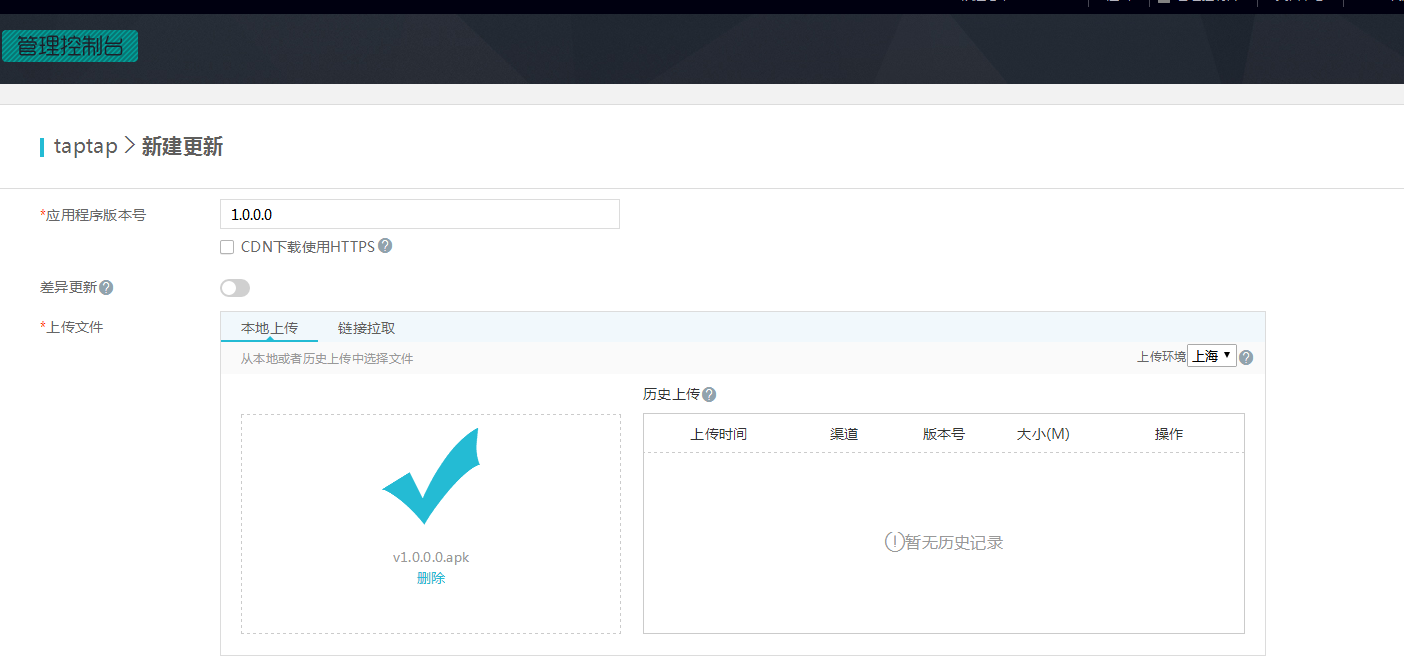
Iris: 目前发展最快的Go Web框架。提供完整的MVC功能并且面向未来。   
\* <https://github.com/kataras/iris>   
\* [https://iris-go.com](https://iris-go.com/)

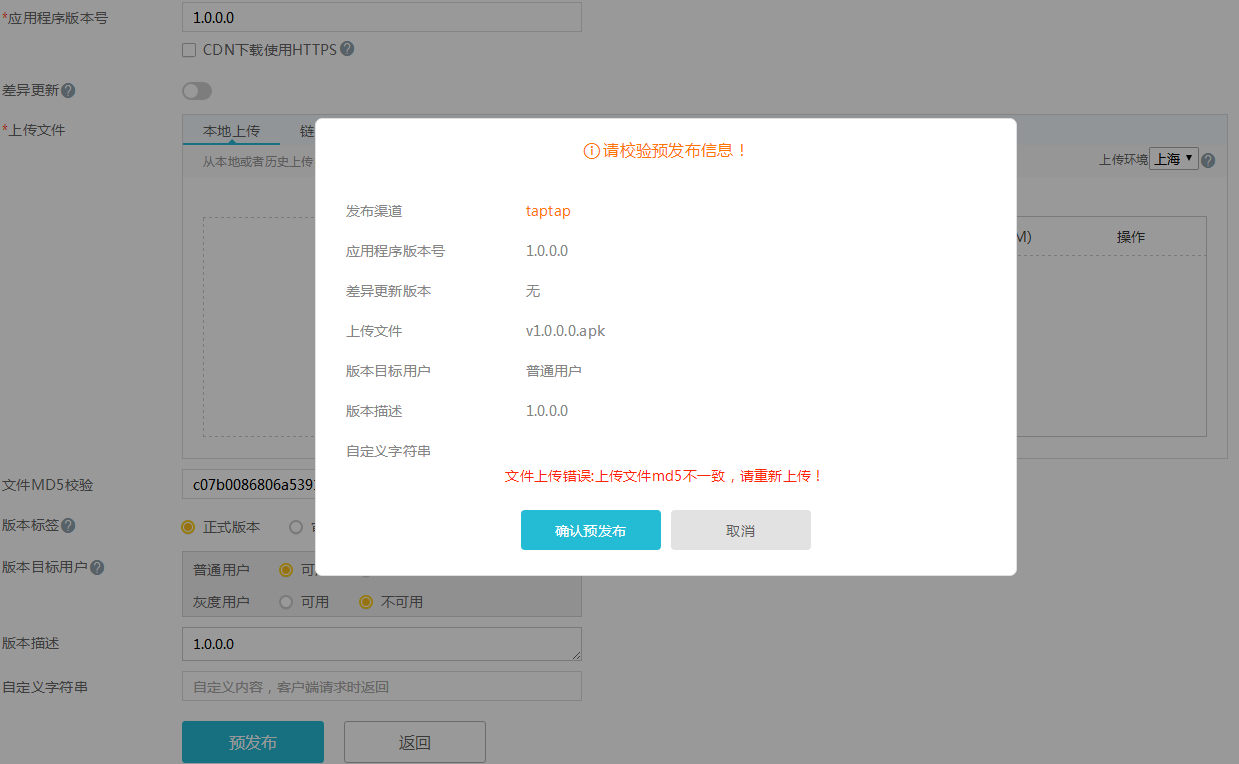
Revel: 一个高生产率，全栈Go语言的Web框架。   
\* <https://github.com/revel/revel>   
\* [https://revel.github.io](https://revel.github.io/)

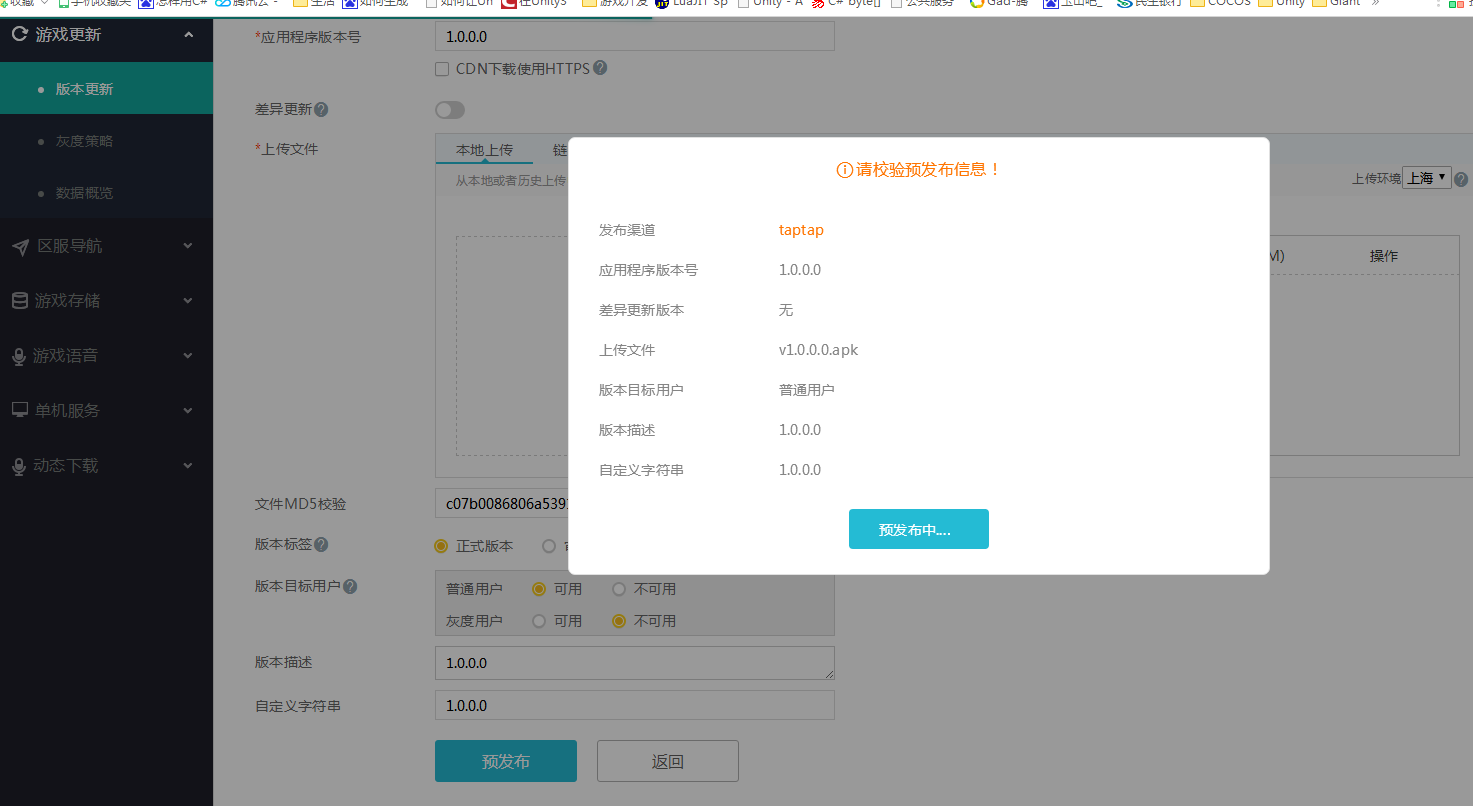
# dolphin

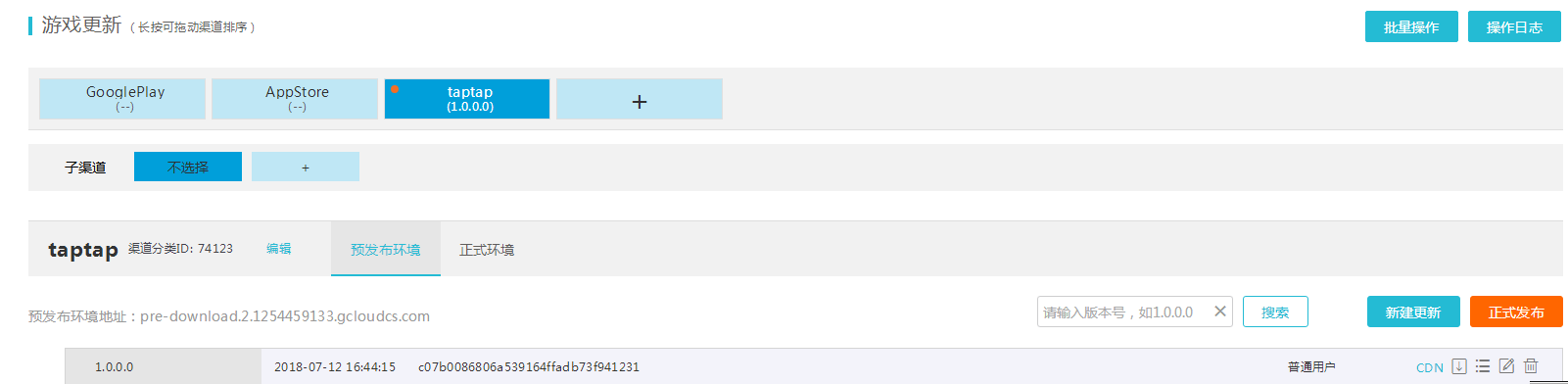
## 上传

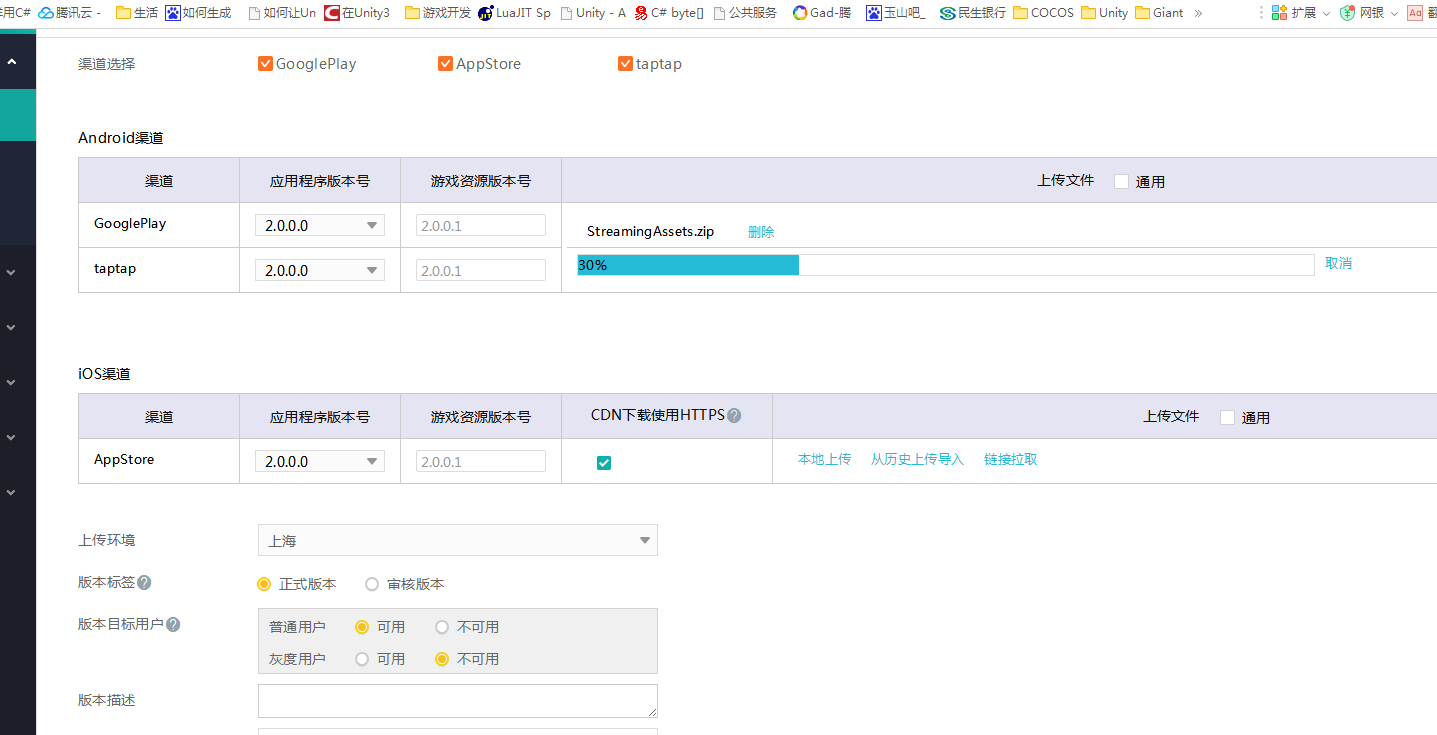












## 资源更新

每次打包的资源必须是游戏对应的全量资源，在更新的过程中，dolphin的sdk会自动和玩家本地的资源对比，下载差异资源； 以打包到zip文件中的单个文件为单位做差异，所以单个资源文件不应过大，避免大文件少量数据区修改导致大文件下载； 资源更新只做文件替换和增加，不做文件删除，游戏应避免和删除相关的游戏逻辑；

## 各种界面：

### cdn选择



### 下载地址



### 资源更新一定是基于指定大版本的



### 资源更新必须是zip格式文件



### 预发布资源



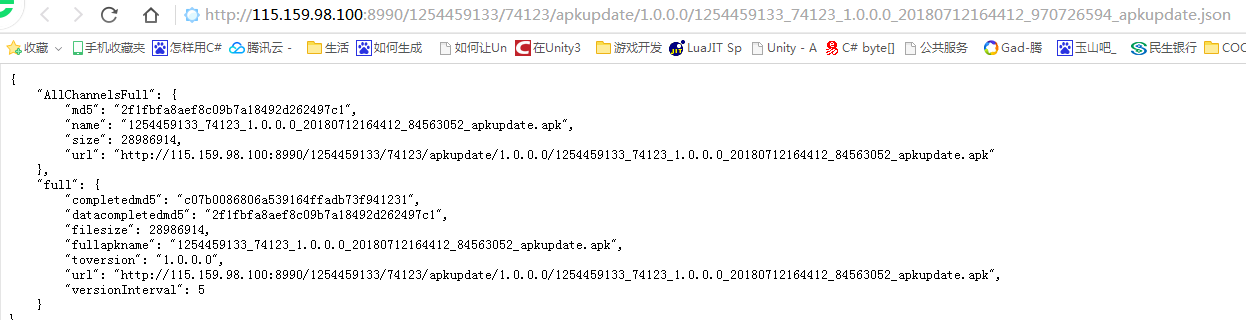
### 操作日志



## 正式发布



## 游戏版本信息



{

## 地址

预发布

<http://115.159.98.100:8990/1254459133/74123/apkupdate/1.0.0.0/1254459133_74123_1.0.0.0_20180712164412_84563052_apkupdate.apk>

<http://115.159.98.100:8990/1254459133/74123/apkupdate/1.0.0.0/1254459133_74123_1.0.0.0_20180712164412_84563052_apkupdate.apk>

http://115.159.98.100:8990/1254459133/74123/apkupdate/3.0.0.0/1254459133\_**74123**\_3.0.0.0\_20180712234258\_1234620793\_apkupdate.apk

[http://115.159.98.100:8990/1254459133/74121/**apkupdate/**3.0.0.0/1254459133\_**74121**\_3.0.0.0\_20180712234259\_36897268\_apkupdate.apk](http://115.159.98.100:8990/1254459133/74121/apkupdate/3.0.0.0/1254459133_74121_3.0.0.0_20180712234259_36897268_apkupdate.apk)

http://115.159.98.100:8990/1254459133/74121/**cures**/2.0.0.1/1254459133\_74121\_2.0.0.1\_20180712233402\_733684003\_cures.zip

http://115.159.98.100:8990/1254459133/74123/apkupdate/1.0.0.0/1254459133\_74123\_1.0.0.0\_20180712164412\_84563052\_apkupdate.apk

<http://115.159.98.100:8990/1254459133/74123/cures/1.0.0.1/1254459133_74123_1.0.0.1_20180712230621_245860808_cures.zip>

<http://115.159.98.100:8990/1254459133/74123/cures/1.0.0.1/1254459133_74123_1.0.0.1_20180712230621_245860808_cures.zip>

http://115.159.98.100:8990/1254459133/74121/cures/2.0.0.1/1254459133\_74121\_2.0.0.1\_20180712233402\_733684003\_cures.zip

正式发布

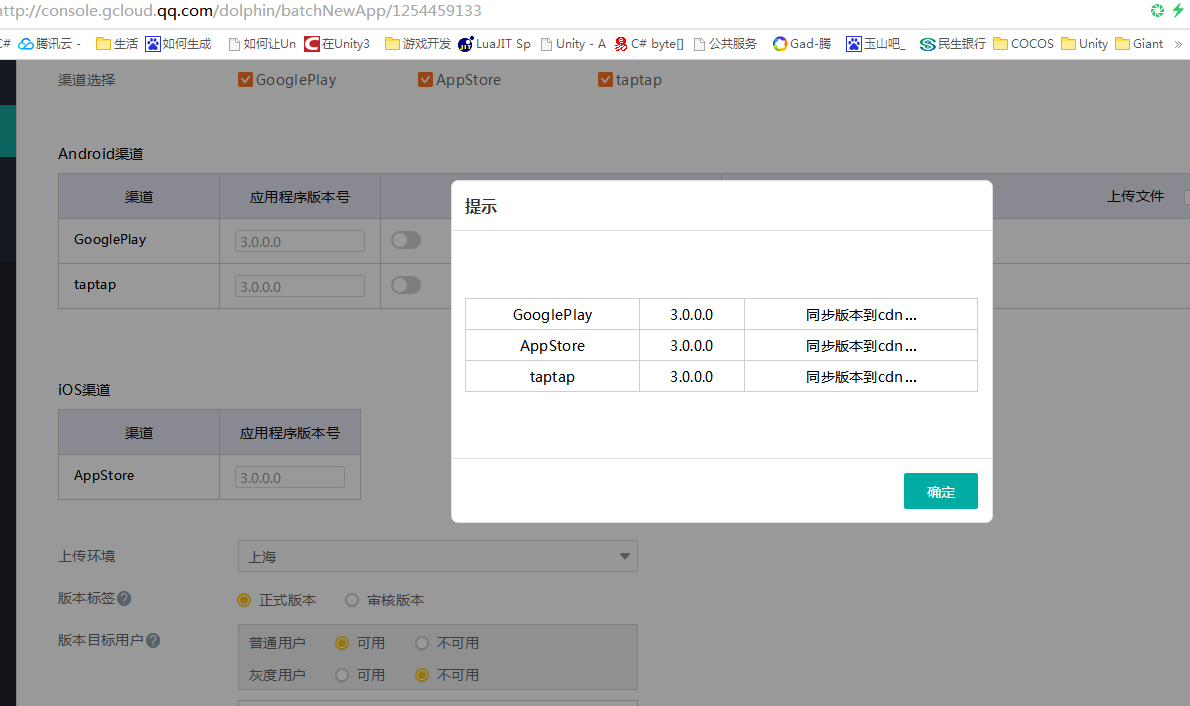
<http://115.159.98.100:8990/1254459133/74123/apkupdate/1.0.0.0/1254459133_74123_1.0.0.0_20180712164412_84563052_apkupdate.apk>

<http://115.159.98.100:8990/1254459133/74123/apkupdate/1.0.0.0/1254459133_74123_1.0.0.0_20180712164412_84563052_apkupdate.apk>

<http://115.159.98.100:8990/1254459133/74121/cures/2.0.0.1/1254459133_74121_2.0.0.1_20180712233402_733684003_cures.zip>

http://115.159.98.100:8990/1254459133/74121/cures/2.0.0.1/1254459133\_74121\_2.0.0.1\_20180712233402\_733684003\_cures.zip

## 同步到CDN：--》同步到预发布环境



# AssetExpress

## 上传、下载

Add a function for large files upload?

https://github.com/kataras/iris/issues/690

iris官方：

<https://github.com/kataras/iris/blob/master/_examples/http_request/upload-files/main.go#L38>

<https://stackoverflow.com/questions/36254522/golang-which-function-for-uploading-large-files>

<https://github.com/kataras/iris/issues/619>

# IRIS