

wetoken

公众号token服务，统一管理token，jsticket，过期后自动从官网获取，保存到数据库、本地缓存，并对外提供获取、刷新接口。

- ✓ 支持独立部署
- ✓ 支持代码集成
- ✓ 代码、接口添加公众号
- ✓ 代码、接口获取token，jsticket
- ✓ 获取时自动检查过期
- ✓ 定时从官网获取
- ✓ 基于 hydra 构建

一、独立服务

多个系统共用一个公众号，可使用wtserver，集中维护token，jsticket。根据向导即可安装数据表结构，启动后即可对外提供服务。

编译安装

```
~/work/bin$ go install github.com/micro-plat/wetoken/wtserver # mysql
```

或

```
~/work/bin$ go install -tags "oci" github.com/micro-plat/wetoken/wtserver # oracle
```

安装数据表

```
wtserver install -r zk://192.168.106.18 -c t
```

启动服务

```
wtserver start -r zk://192.168.106.18 -c t
```

二、代码集成

只有一个系统使用公众号，为减少子系统维护，可直接通过代码集成到现有系统中。

1、创建任务表:

1. 编译 wetoken

```
~/work/bin$ go install github.com/micro-plat/wetoken #mysql
```

或

```
~/work/bin$ go install -tags "oci" github.com/micro-plat/wetoken # oracle
```

2. 运行命令

wetoken [注册中心地址] [平台名称] 即可将 wetoken 需要的表创建到 /平台/var/db/db 配置对应的数据库中。

```
~/work/bin$ wetoken zk://192.168.0.109 mall #根据/mall/var/db/db创建数据库
```

或

```
~/work/bin$ wetoken zk://192.168.0.109 mall mdb #根据/mall/var/db/mdb创建数据库
```

2、编码

绑定服务

```
app.Initializing(func(c component.IContainer) error {  
    wetoken.Bind(app, true, "wx9e02ddcc34513fd4") //绑定公众号查询，刷新接口和自动刷新任务  
})
```

或

```
app.Initializing(func(c component.IContainer) error {  
    wetoken.Cron(app, true, "wx9e02ddcc34513fd4") //只绑定自动刷新任务  
})
```

当 app.IsDebug==true 时会自动绑定公众号添加服务，可通过接口添加公众号

获取token

```
tk:=token.NewToken(container, appid)
token, err:=tk.Get()//不存在或已过期，自动从官网获取
```

获取sticket

```
tk:=token.NewTicket(container, appid)
ticket, err:=tk.Get()//不存在或已过期，自动从官网获取
```

三、微信 sdk 中使用 token

```
package main

import "github.com/micro-plat/wechat/mp"
import "github.com/micro-plat/wechat/mp/menu"

func main(){
    tk := mp.NewDefaultAccessTokenByURL(
        "wx9e02ddcc34513fd4", "6acb2b999177524beba3d97d54df2de5",
        "http://localhost:9999/wx9e02ddcc34513fd4/wechat/token/get")
    ctx := mp.NewContext(tk)
    mu := &menu.Menu{
        Buttons: []menu.Button{
            menu.Button{Type: menu.ButtonTypeView, Name: "搜索1", URL: "http://www.baidu.com"},
            menu.Button{Type: menu.ButtonTypeView, Name: "搜索2", URL: "http://www.baidu.com"},
            menu.Button{Type: menu.ButtonTypeView, Name: "搜索3", URL: "http://www.baidu.com"}
        },
    }
    if err := menu.Create(ctx, mu);err!=nil{
        fmt.Println(err)
    }
}
```

四、其它

服务列表

服务名	类型	wetoken.Bind	wetoken.Cron	说明
/wechat/app/create	api	可选	可选	添加公众号信息

服务名	类型	wetoken.Bind	wetoken.Cron	说明
/[appid]/wechat/token/get	api	√	×	获取最新token
/[appid]/wechat/ticket/get	api	√	×	获取最新ticket
/[appid]/wechat/token/refresh	cron	√	√	定时刷新token任务
/[appid]/wechat/ticket/refresh	cron	√	√	定时刷新ticket任务

添加公众号

```
~/work/bin$ curl "http://192.168.5.71:9999/wechat/app/create?appid=wx9e02ddcc34513fd4&secret=9_iZx3iItbJssKKzJTe9CeoJqY7678P0nYkHRnI"
{"errcode":0, "errmsg":"success"}
```

获取 access token

```
~/work/bin$ curl http://192.168.5.71:9999/wx9e02ddcc34513fd4/wechat/token/get
{"errcode":0, "errmsg":"success", "access_token":"9_iZx3iItbJssKKzJTe9CeoJqY7678P0nYkHRnI"}
```

获取 jsapi ticket

```
~/work/bin$ curl http://192.168.5.71:9999/wx9e02ddcc34513fd4/wechat/ticket/get
{"errcode":0, "errmsg":"success", "ticket":"HoagFKDcsGMVCiY2vOjf9ki38m7AHQZG34U1VtA70B09j"}
```