

# 附件-腾讯里约开放平台-tifapi部署-1.0

## 1. Tifapi应用基本安装配置

Tifapi用来实现RIO与微信、企业微信的通信，在部署微信小程序支持环境时需要用到。

### 1.1. 版本获tifapi取

Tifapi由数广产品团队维护，项目实施时需联系运维组获取最新可用的版本。

### 1.2. 解压安装

解压tifapi应用到/data/solution文件夹：

```
tar xvfz tifapi20190818.tar.gz
#对相关文件进行授权
chmod +x bin/*
```

### 1.3. 导入下列redis数据

```
HSET "tif:paas:secret" "gwymp" "wx*****;9*****c;1"
```

**说明：** gwymp为调用api网关服务的paasid， wx为微信开放平台提供的appid,9\*\*\*\*\*c为微信开放平台提供的appsecret,0为使用微信的认证服务，1为使用openapi的认证服务。

### 1.4. 修改etc/config.json

配置文件如下： （注意重点配置）

```
{
  #监听的ip和端口
  "host": "0.0.0.0:18083",
  "log_dir": "./log",
  #tifapi调用api的paasid，如在apigate中非此id，需修改为了系统中设定的paasid。一般
  #情况下需配置为超级paasid，不然要做大量的服务订阅。
  "tif_paasid": "tif",
  #上述paasid对应的paastoken
  "tif_token": "22G5deiQMq0aVqB0IS08YR0hK4BpWx1W",
  "sign_time_limit": 600,
  "check_sign": true,
  "userext_expire_hours": 720,
  "credential": {
    "expire": 60,
    "sync_expire": 30,
    "code_expire": 300,
  }
}
```

```

        "idauth_client_id": "",
        "idauth_client_secret": "",
        "cityservice_appid": "",
        "cityservice_aes_secret": ""
    },
    "wxpay": {
        "mchid": "1498726202",
        "certserialno":
"67914E342D2B658FFDBD621660534254FB4FF749",
        "privatepem": "./etc/private.pem"
    },
    "db": {
        #redis的访问地址，跟智能网关的配置保持一致
        "redis": "111.13.91.6:6379",
        #redis的密码，跟智能网关的配置保持一致
        "redis_pwd": "ee06167b10a177f60766d35baa81955d",
        #mongodb的访问地址，跟智能网关的配置保持一致
        "mongo": "111.13.91.6:27017",
        #mongodb的用户，跟智能网关的配置保持一致
        "db_user": "rio",
        #rio用户的密码，跟智能网关的配置保持一致
        "db_pwd": "ee06167b10a177f60766d35baa81955d"
    },
    "ebus": {
        "hosts": [
            #访问apigate的地址
            "http://127.0.0.1:8080"
        ],
        "services": {
            #以下为openapi登录接口配置，此配置在使用openapi时才使用（即上节导入数据
            #最后一段为1），gwypm替换成实际的paasid
            "openapi_mplogin": "/ebus/gwypm/api/mp/login",
            "openapi_token":
"/ebus/gwypm/api/get_access_token",
            "openapi_authface":
"/ebus/gwypm/api/check_face_verify_result",
            "openapi_faceimg":
"/ebus/gwypm/api/get_face_image",
            "openapi_realname":
"/ebus/gwypm/api/mp/get_real_name",
            "openapi_mplogout": "/ebus/gwypm/api/logout",
            #openapi登录配置结束
            "ecard_residence":
"/ebus/huazi/test/app/service/electronic/card/query",
            "ecard_driver":
"/ebus/youlujia/sandbox/dgdAPIServer/getUserInfoToHome/getDriverInfoForCard",
            "ecard_vehicle":
"/ebus/youlujia/dgd/dgdAPIServer/getUserInfoToHome/getVehicleInfoForCard",
            "ecard_visa":
"/ebus/tianjian/gdstxcx/queryCertInfo/queryzjxx",
            "ecard_visa_detail":
"/ebus/tianjian/gdstxcx/queryCertInfo/queryqzsyys",
            "ecard_hpfund":

```

```

"/ebus/gwypm/api/bg/hpfund/deposit_info",
    "ecard_shebao":
"/ebus/gwypm/api/bg/get_card_detail",
    #访问wxapi的地址, 如果调用wxapi的地址有变更, 此处也要同步修改, 下同。
    "wxapi_token": "/ebus/wxapi/cgi-bin/token",
    "wxapi_codesession":
"/ebus/wxapi/sns/jscode2session",
    "wxapi_face":
"/ebus/wxapi/cityservice/face/identify/getinfo",
    "wxapi_cityservice_codeinfo":
"/ebus/wxapi/cityservice/codeinfo",
    "wxapi_realname": "/ebus/wxapi/cgi-
bin/wxopen/getrealnameinfo",
    #访问wxapi配置结束
    "zwx_token": "/ebus/zwxapi/cgi-bin/gettoken",
    "qywx_token": "/ebus/qywxapi/cgi-bin/gettoken",
    "idauth_token": "/ebus/dabai/api/accesstoken",
    "idauth_req": "/ebus/dabai/api/idauthreq",
    "gss_authface":
"/ebus/gss/api/r/authentication/Identification",
    "gwy_authface":
"/ebus/gwypm/api/r/usercenter/login"
    }
},
"paas": {
#小工具访问地址
    "host": "http://127.0.0.1:8888"
}
}

```

## 1.5. 启动tifapi应用:

```

./start.sh

tifapi: no process found

tail -f ../log/tifapi.INFO

Log file created at: 2018/09/20 17:37:03

Running on machine: localhost

Binary: Built with gc go1.9.4 for linux/amd64

Log line format: [IWEF]mmdd hh:mm:ss.uuuuuu threadid file:line] msg

```

## 2. 规则初始化

### 2.1. 在准入网关导入以下4条规则

- [routeRule\\_8\\_ebus.json](#)
- [routeRule\\_0\\_tifapi.json](#)
- [routeRule\\_0\\_mp.json](#)
- [routeRule\\_0\\_Internet\\_0.json](#) 能直接访问互联网
- [routeRule\\_0\\_Internet\\_1.json](#) 不能直接访问互联网，需要修改该文件里的{{接入网关IP}}为真实ip

使用json2redis工具：

```
/data/rio/accessgate/init/json2redis -password
ee06167b10a177f60766d35baa81955d -redis 127.0.0.1:6379 -name accessgate -
config routeRule_8_ebus.json
/data/rio/accessgate/init/json2redis -password
ee06167b10a177f60766d35baa81955d -redis 127.0.0.1:6379 -name accessgate -
config routeRule_0_mp.json
/data/rio/accessgate/init/json2redis -password
ee06167b10a177f60766d35baa81955d -redis 127.0.0.1:6379 -name accessgate -
config routeRule_0_tifapi.json

# 以下两个配置中只能导入一个，请按实际应用场景选择
/data/rio/accessgate/init/json2redis -password
ee06167b10a177f60766d35baa81955d -redis 127.0.0.1:6379 -name accessgate -
config routeRule_0_Internet_0.json
/data/rio/accessgate/init/json2redis -password
ee06167b10a177f60766d35baa81955d -redis 127.0.0.1:6379 -name accessgate -
config routeRule_0_Internet_1.json
```

## 2.2. 在API网关导入如下规则

- [routeRule\\_\\$top.json](#) 本规则中包含了**超级paasid**的规则，注意根据需要进行修改。使用json2redis工具：

```
/data/rio/accessgate/init/json2redis -password
ee06167b10a177f60766d35baa81955d -redis 127.0.0.1:6379 -name apigate -
config routeRule\_stop.json
```

### 注意：

- json2redis为私有工具，在执行时会删除redis中已原有的键值
- 导入数据后，需要在智网关服务器上执行如下脚本进行数据升级，否则导入的数据无法显示。仅支持0.2.5以上版本。完成后在RIO控制台检查。

## 3. 创建相关服务

### 3.1. 创建微信API服务

- 创建wxapi应用
- 在wxapi应用上发布wxapi服务。
  - 如果服务器能访问外网, targetHost为: api.weixin.qq.com,path为/wxapi/,targetPath为/
  - 如果不能访问外网, targetHost为准入网关或者接入网关的ip:port,targetPath为/Internet/https/api.weixin.qq.com/
- 审批服务

### 3.2. 创建Tipapi相关服务

- 创建tif应用
- 在tif应用上发布tif服务。
  - 如果服务器能访问外网, targetHost所有tifapi服务器的ip:port,path为/tif/,targetPath为/

## 4. 创建索引

在mongodb中创建如下索引

```
db.person.createIndex({cid: 1}, {unique:true})
db.person.createIndex({createtime: 1})
db.person.createIndex({phone: 1})

db.account.createIndex({uid: 1}, {unique:true})
db.account.createIndex({openid: 1, appid: 1})
db.account.createIndex({cid: 1})
db.account.createIndex({unionid: 1})
db.account.createIndex({appid: 1, updatetime: 1})
db.account.createIndex({updatetime: 1})
```