

# apn配置

## apn参数

```
typedef struct
{
    std::string apn;                apn名字
    std::string user;              用户名
    std::string passwd;            密码
    int authpref;                  身份认证类型
    NET_IP_VER_E ip_ver;           IPV4 or IPV6 or IPV46
    std::string ip_address(可选);   ip 地址
    std::string dns_address(可选);  dns
    std::string gateways(可选);     网关
    int priority(可选)              优先级
} RIL_RMNET_PARAMETERS_ST;
```

## ip\_ver 类型

```
typedef enum {
    NET_IP_VER_V4 0x1,
    NET_IP_VER_V6 0x2,
    NET_IP_VER_V4V6 0x3,
}NET_IP_VER_E;
```

## 认证类型

```
typedef enum {
NET_AUTH_PREF_PAP_CHAP_NOT_ALLOWED = 0,
NET_AUTH_PREF_PAP_ONLY_ALLOWED = 1,
NET_AUTH_PREF_CHAP_ONLY_ALLOWED = 2,
NET_AUTH_PREF_PAP_CHAP_BOTH_ALLOWED = 3,
}NET_AUTH_PREF_E;
```

## 方案设计

### Json

可能的apn写在 json中

通过key指定取某一个具体apn

```
{
```

```
"version" : "1",
"apnConfig":"apn1",

"apn1":{
    "name": "...",      apn名字
    "user" : "...",    用户名
    "passwd" : "...",  密码
    "authpref" : "...", 身份认证类型
    "ip_ver" : "...",   IPV4 or IPV6 or IPV46
    "priority" : "...", 优先级(可选)
}
```

```
}
```

### Telephony

在拨号的时候获取apn配置(或者在初始化的时候获取一次)

通过rild接口将apn结构体传入rild

再通过rpc调用ag550进行拨号

# 完整流程

