**软件概要设计书（云平台）**

版本inner.0

|  |  |  |  |
| --- | --- | --- | --- |
| **文件编号：** | ICVP.QS.0005 | **生效日期：** | YYYY-MM-DD |
| **编 制：** |  | **会 签：** |  |
| **审 核：** |  | **批 准：** |  |

**修改记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **版次号** | 修改条款及内容 | **修改人** | **页码** | **修改日期** |
| 1 | V1.2 | 1. 修改3.4.2接口：   修改返回字段status(绑定状态)：   1. 未绑定 2. 已绑定 3. 已解绑 4. 绑定中 5. 解绑中 6. 新增3.2.11接口：查询用户证件信息 7. 新增3.2.13接口：提交实名认证 | 黄云 |  | 2020-10-27 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**目录**

[1 文档介绍 12](#_Toc54714189)

[1.1 文档目的 12](#_Toc54714190)

[1.2 适用范围 12](#_Toc54714191)

[1.3 术语与解释 12](#_Toc54714192)

[1.4 文档约定 12](#_Toc54714193)

[2 用户服务设计 13](#_Toc54714194)

[2.1 系统功能模块图 13](#_Toc54714195)

[2.2 系统功能模块介绍 13](#_Toc54714196)

[2.3 功能模块说明 13](#_Toc54714197)

[2.3.1 用户与终端设备 13](#_Toc54714198)

[2.3.2 用户与服务 14](#_Toc54714199)

[2.3.3 用户授权关系 14](#_Toc54714200)

[2.3.4 用户安全 15](#_Toc54714201)

[2.3.5 用户三方登录和绑定 15](#_Toc54714202)

[3 接口设计 16](#_Toc54714203)

[3.1 注册相关 16](#_Toc54714204)

[3.1.1 注册接口 16](#_Toc54714205)

[3.1.1.1 接口说明 16](#_Toc54714206)

[3.1.1.2 接口url 17](#_Toc54714207)

[3.1.1.3 接口入参 17](#_Toc54714208)

[3.1.1.4 请求示例 17](#_Toc54714209)

[3.1.1.5 接口出参 17](#_Toc54714210)

[3.1.1.6 响应示例 18](#_Toc54714211)

[3.2 用户基本信息维护 18](#_Toc54714212)

[3.2.1 添加用户信息（登录态） 18](#_Toc54714213)

[3.2.1.1 接口说明 18](#_Toc54714214)

[3.2.1.2 接口url 18](#_Toc54714215)

[3.2.1.3 接口入参 18](#_Toc54714216)

[3.2.1.4 请求示例 18](#_Toc54714217)

[3.2.1.5 接口出参 19](#_Toc54714218)

[3.2.1.6 返回示例 19](#_Toc54714219)

[3.2.2 查询用户信息接口（登录态） 19](#_Toc54714220)

[3.2.2.1 接口说明 19](#_Toc54714221)

[3.2.2.2 接口url 19](#_Toc54714222)

[3.2.2.3 接口入参 19](#_Toc54714223)

[3.2.2.4 请求示例 19](#_Toc54714224)

[3.2.2.5 接口出参 20](#_Toc54714225)

[3.2.2.6 返回示例 20](#_Toc54714226)

[3.2.3 修改用户信息接口（登录态） 20](#_Toc54714227)

[3.2.3.1 接口说明 20](#_Toc54714228)

[3.2.3.2 接口url 20](#_Toc54714229)

[3.2.3.3 接口入参 20](#_Toc54714230)

[3.2.3.4 请求示例 21](#_Toc54714231)

[3.2.3.5 接口出参 21](#_Toc54714232)

[3.2.3.6 返回示例 21](#_Toc54714233)

[3.2.4 校验用户名接口 21](#_Toc54714234)

[3.2.4.1 接口说明 21](#_Toc54714235)

[3.2.4.2 接口url 22](#_Toc54714236)

[3.2.4.3 接口入参 22](#_Toc54714237)

[3.2.4.4 请求示例 22](#_Toc54714238)

[3.2.4.5 接口出参 22](#_Toc54714239)

[3.2.4.6 返回示例 22](#_Toc54714240)

[3.2.5 修改密码接口（登录态） 22](#_Toc54714241)

[3.2.5.1 接口说明 22](#_Toc54714242)

[3.2.5.2 接口url 23](#_Toc54714243)

[3.2.5.3 接口入参 23](#_Toc54714244)

[3.2.5.4 请求示例 23](#_Toc54714245)

[3.2.5.5 接口出参 23](#_Toc54714246)

[3.2.5.6 返回示例 23](#_Toc54714247)

[3.2.6 找回密码接口 23](#_Toc54714248)

[3.2.6.1 接口说明 23](#_Toc54714249)

[3.2.6.2 接口url 24](#_Toc54714250)

[3.2.6.3 接口入参 24](#_Toc54714251)

[3.2.6.4 请求示例 24](#_Toc54714252)

[3.2.6.5 接口出参 24](#_Toc54714253)

[3.2.6.6 返回示例 24](#_Toc54714254)

[3.2.7重绑手机（登录态） 24](#_Toc54714255)

[3.2.7.1 接口说明 25](#_Toc54714256)

[3.2.7.2 接口url 25](#_Toc54714257)

[3.2.7.3 接口入参 25](#_Toc54714258)

[3.2.7.4 请求示例 25](#_Toc54714259)

[3.2.7.5 接口出参 25](#_Toc54714260)

[3.2.7.6 返回示例 25](#_Toc54714261)

[3.2.8校验用户是否设置密码（登录态） 26](#_Toc54714262)

[3.2.8.1 接口说明 26](#_Toc54714263)

[3.2.8.2 接口url 26](#_Toc54714264)

[3.2.8.3 接口入参 26](#_Toc54714265)

[3.2.8.4 请求示例 26](#_Toc54714266)

[3.2.8.5 接口出参 26](#_Toc54714267)

[3.2.8.6 返回示例 26](#_Toc54714268)

[3.2.9 用户属性添加与修改(登录态) 27](#_Toc54714269)

[3.2.9.1 接口说明 27](#_Toc54714270)

[3.2.9.2 接口url 27](#_Toc54714271)

[3.2.9.3 接口入参 27](#_Toc54714272)

[3.2.9.4 请求示例 28](#_Toc54714273)

[3.2.9.5 响应参数 28](#_Toc54714274)

[3.2.9.6 返回示例 28](#_Toc54714275)

[3.2.10 用户属性信息查看（登录态） 28](#_Toc54714276)

[3.2.10.1 接口说明 28](#_Toc54714277)

[3.2.10.2 接口url 28](#_Toc54714278)

[3.2.10.2 接口入参 28](#_Toc54714279)

[3.2.10.4 请求示例 29](#_Toc54714280)

[3.2.10.5 接口出参 29](#_Toc54714281)

[3.2.10.6 返回示例 29](#_Toc54714282)

[3.2.11 查询用户证件信息（登录态） 30](#_Toc54714283)

[3.2.11.1 接口说明 30](#_Toc54714284)

[3.2.11.2 接口url 30](#_Toc54714285)

[3.2.11.2 接口入参 30](#_Toc54714286)

[3.2.11.4 请求示例 30](#_Toc54714287)

[3.2.11.5 接口出参 30](#_Toc54714288)

[3.2.11.6 返回示例 31](#_Toc54714289)

[3.2.12 用户列表获取 31](#_Toc54714290)

[3.2.12.1 接口说明 31](#_Toc54714291)

[3.2.12.2 接口url 31](#_Toc54714292)

[3.2.12.3 接口入参 31](#_Toc54714293)

[3.2.12.4 请求示例 31](#_Toc54714294)

[3.2.12.5 接口出参 31](#_Toc54714295)

[3.2.12.6 返回示例 32](#_Toc54714296)

[3.3 用户设备 32](#_Toc54714297)

[3.3.1 用户设备信息上报接口（登录态） 32](#_Toc54714298)

[3.3.1.1 接口说明 32](#_Toc54714299)

[3.3.1.2 接口url 32](#_Toc54714300)

[3.3.1.3 接口入参 32](#_Toc54714301)

[3.3.1.4 请求示例 33](#_Toc54714302)

[3.3.1.5 接口出参 33](#_Toc54714303)

[3.3.1.6 返回示例 33](#_Toc54714304)

[3.3.2 获取用户推送rid接口（非登录态） 33](#_Toc54714305)

[3.3.2.1 接口说明 33](#_Toc54714306)

[3.3.2.2 接口url 33](#_Toc54714307)

[3.3.2.3 接口入参 34](#_Toc54714308)

[3.3.2.4 请求示例 34](#_Toc54714309)

[3.3.2.5 接口出参 34](#_Toc54714310)

[3.3.2.6 返回示例 34](#_Toc54714311)

[3.3.3 获取用户最近使用的设备信息（非登录态） 34](#_Toc54714312)

[3.3.3.1 接口说明 34](#_Toc54714313)

[3.3.3.2 接口url 35](#_Toc54714314)

[3.3.3.3 接口入参 35](#_Toc54714315)

[3.3.3.4 请求示例 35](#_Toc54714316)

[3.3.3.5 接口出参 35](#_Toc54714317)

[3.3.3.6 返回示例 35](#_Toc54714318)

[3.4 用户车辆 36](#_Toc54714319)

[3.4.1 根据车辆查询绑定用户 36](#_Toc54714320)

[3.4.1.1 接口说明 36](#_Toc54714321)

[3.4.1.2 接口url 36](#_Toc54714322)

[3.4.1.3 接口入参 36](#_Toc54714323)

[3.4.1.4 请求示例 36](#_Toc54714324)

[3.4.1.5 接口出参 36](#_Toc54714325)

[3.4.1.6 返回示例 37](#_Toc54714326)

[3.4.2 获取用户绑定车辆接口（登录态） 37](#_Toc54714327)

[3.4.2.1 接口说明 37](#_Toc54714328)

[3.4.2.2 接口url 37](#_Toc54714329)

[3.4.2.3 接口入参 37](#_Toc54714330)

[3.4.2.4 接口请求示例 37](#_Toc54714331)

[3.4.2.5 接口出参 37](#_Toc54714332)

[3.4.2.6 响应示例 38](#_Toc54714333)

[3.4.3 绑定用户车辆 38](#_Toc54714334)

[3.4.3.1 接口说明 39](#_Toc54714335)

[3.4.3.2 接口url 39](#_Toc54714336)

[3.4.3.3 接口入参 39](#_Toc54714337)

[3.4.3.4 请求示例 39](#_Toc54714338)

[3.4.3.5 接口出参 39](#_Toc54714339)

[3.4.3.6 返回示例 39](#_Toc54714340)

[3.4.4解绑用户车辆 39](#_Toc54714341)

[3.4.4.1 接口说明 39](#_Toc54714342)

[3.4.4.2 接口url 40](#_Toc54714343)

[3.4.4.3 接口入参 40](#_Toc54714344)

[3.4.4.4 请求示例 40](#_Toc54714345)

[3.4.4.5 接口出参 40](#_Toc54714346)

[3.4.4.6 返回示例 40](#_Toc54714347)

[3.4.5查询历史车主 41](#_Toc54714348)

[3.4.5.1 接口说明 41](#_Toc54714349)

[3.4.5.2 接口url 41](#_Toc54714350)

[3.4.5.3 接口入参 41](#_Toc54714351)

[3.4.5.4 请求示例 41](#_Toc54714352)

[3.4.5.5 接口出参 41](#_Toc54714353)

[3.4.5.6 返回示例 41](#_Toc54714354)

[3.5 用户服务 42](#_Toc54714355)

[3.5.1 服务绑定 42](#_Toc54714356)

[3.5.1.1 接口说明 42](#_Toc54714357)

[3.5.1.2 接口url 42](#_Toc54714358)

[3.5.1.3 接口入参 43](#_Toc54714359)

[3.5.1.4 请求示例 43](#_Toc54714360)

[3.5.1.5 接口出参 43](#_Toc54714361)

[3.5.1.6 返回示例 43](#_Toc54714362)

[3.5.2 修改用户服务清单 44](#_Toc54714363)

[3.5.2.1 接口说明 44](#_Toc54714364)

[3.5.2.2 接口url 44](#_Toc54714365)

[3.5.2.3 接口入参 44](#_Toc54714366)

[3.5.2.4 请求示例 44](#_Toc54714367)

[3.5.2.5 接口出参 44](#_Toc54714368)

[3.5.2.6 返回示例 45](#_Toc54714369)

[3.5.3获取用户服务清单 45](#_Toc54714370)

[3.5.3 接口说明 45](#_Toc54714371)

[3.5.3.2 接口url 45](#_Toc54714372)

[3.5.3.3 接口入参 45](#_Toc54714373)

[3.5.3.4 请求示例 45](#_Toc54714374)

[3.5.3.5 接口出参 45](#_Toc54714375)

[3.5.3.6 返回示例 46](#_Toc54714376)

[3.5.4用户自定义角色 46](#_Toc54714377)

[3.5.4.1 接口说明 46](#_Toc54714378)

[3.5.4.2 接口url 46](#_Toc54714379)

[3.5.4.3 接口入参 47](#_Toc54714380)

[3.5.4.4 请求示例 47](#_Toc54714381)

[3.5.4.5 接口出参 47](#_Toc54714382)

[3.5.4.6 返回示例 47](#_Toc54714383)

[3.5.5获取角色清单 47](#_Toc54714384)

[3.5.5.1 接口说明 47](#_Toc54714385)

[3.5.5.2 接口url 47](#_Toc54714386)

[3.5.5.3 接口入参 48](#_Toc54714387)

[3.5.5.4 请求示例 48](#_Toc54714388)

[3.5.5.5 接口出参 48](#_Toc54714389)

[3.5.5.6 返回示例 48](#_Toc54714390)

[3.5.6 修改角色 49](#_Toc54714391)

[3.5.6.1 接口说明 49](#_Toc54714392)

[3.5.6.2 接口url 49](#_Toc54714393)

[3.5.6.3 接口入参 49](#_Toc54714394)

[3.5.6.4 请求示例 49](#_Toc54714395)

[3.5.6.5 接口出参 49](#_Toc54714396)

[3.5.6.6 返回示例 49](#_Toc54714397)

[3.5.7 用户添加服务到角色 50](#_Toc54714398)

[3.5.7.1 接口说明 50](#_Toc54714399)

[3.5.7.2 接口url 50](#_Toc54714400)

[3.5.7.3 接口入参 50](#_Toc54714401)

[3.5.7.4 请求示例 50](#_Toc54714402)

[3.5.7.5 接口出参 50](#_Toc54714403)

[3.5.7.6 返回示例 50](#_Toc54714404)

[3.5.8 用户选择从角色中删除服务 51](#_Toc54714405)

[3.5.8.1 接口说明 51](#_Toc54714406)

[3.5.8.2 接口url 51](#_Toc54714407)

[3.5.8.3 接口入参 51](#_Toc54714408)

[3.5.8.4 请求示例 51](#_Toc54714409)

[3.5.8.5 接口出参 51](#_Toc54714410)

[3.5.8.6 返回示例 51](#_Toc54714411)

[3.5.9获取用户角色对应的服务清单 52](#_Toc54714412)

[3.5.9.1 接口说明 52](#_Toc54714413)

[3.5.9.2 接口url 52](#_Toc54714414)

[3.5.9.3 接口入参 52](#_Toc54714415)

[3.5.9.4 请求示例 52](#_Toc54714416)

[3.5.9.5 接口出参 52](#_Toc54714417)

[3.5.9.6 返回示例 52](#_Toc54714418)

[3.6 车辆绑定与授权 53](#_Toc54714419)

[3.6.1 用户车辆授权 53](#_Toc54714420)

[3.6.1.1 接口说明 53](#_Toc54714421)

[3.6.1.2 接口url 53](#_Toc54714422)

[3.6.1.3 接口入参 53](#_Toc54714423)

[3.6.1.4 请求示例 54](#_Toc54714424)

[3.6.1.5 接口出参 54](#_Toc54714425)

[3.6.1.6 返回示例 54](#_Toc54714426)

[3.6.2 获取用户对车辆授权 54](#_Toc54714427)

[3.6.2.1 接口说明 54](#_Toc54714428)

[3.6.2.2 接口url 54](#_Toc54714429)

[3.6.2.3 接口入参 55](#_Toc54714430)

[3.6.2.4 请求示例 55](#_Toc54714431)

[3.6.2.5 接口出参 55](#_Toc54714432)

[3.6.2.6 返回示例 55](#_Toc54714433)

[3.6.3 用户车辆授权角色 55](#_Toc54714434)

[3.6.3.1 接口说明 56](#_Toc54714435)

[3.6.3.2 接口url 56](#_Toc54714436)

[3.6.3.3 接口入参 56](#_Toc54714437)

[3.6.3.4 请求示例 56](#_Toc54714438)

[3.6.3.5 接口出参 56](#_Toc54714439)

[3.6.3.6 返回示例 56](#_Toc54714440)

[3.6.4 获取用户授权车辆授权的角色 56](#_Toc54714441)

[3.6.4.1 接口说明 57](#_Toc54714442)

[3.6.4.2 接口url 57](#_Toc54714443)

[3.6.4.3 接口入参 57](#_Toc54714444)

[3.6.4.4 请求示例 57](#_Toc54714445)

[3.6.4.5 接口出参 57](#_Toc54714446)

[3.6.4.6 返回示例 57](#_Toc54714447)

[3.7 用户安全相关 58](#_Toc54714448)

[3.7.1 用户绑定手机 58](#_Toc54714449)

[3.7.2 用户解绑手机 58](#_Toc54714450)

[3.7.3 用户换绑手机 58](#_Toc54714451)

[3.7.4 用户绑定邮箱 58](#_Toc54714452)

[3.7.5 用户解绑邮箱 58](#_Toc54714453)

[3.7.6 用户换绑邮箱 58](#_Toc54714454)

[3.7.7 用户绑定pin 58](#_Toc54714455)

[3.7.8 用户解绑pin 58](#_Toc54714456)

[3.7.9 用户安全项绑定查询 58](#_Toc54714457)

[3.7.9.1 接口说明 58](#_Toc54714458)

[3.7.9.2 接口url 58](#_Toc54714459)

[3.7.9.3 接口入参 58](#_Toc54714460)

[3.7.9.4 请求示例 59](#_Toc54714461)

[3.7.9.5 接口出参 59](#_Toc54714462)

[3.7.9.6 返回示例 59](#_Toc54714463)

[3.7.10 获取系统安全问题 59](#_Toc54714464)

[3.7.10.1 接口说明 59](#_Toc54714465)

[3.7.10.2 接口url 59](#_Toc54714466)

[3.7.10.3 接口入参 60](#_Toc54714467)

[3.7.10.4 请求示例 60](#_Toc54714468)

[3.7.10.5 接口出参 60](#_Toc54714469)

[3.7.10.6 返回示例 60](#_Toc54714470)

[3.7.11 用户添加安全问题和答案 60](#_Toc54714471)

[3.7.11.1 接口说明 60](#_Toc54714472)

[3.7.11.2 接口url 61](#_Toc54714473)

[3.7.11.3 接口入参 61](#_Toc54714474)

[3.7.11.4 请求示例 61](#_Toc54714475)

[3.7.11.5 接口出参 61](#_Toc54714476)

[3.7.11.6 返回示例 61](#_Toc54714477)

[3.8 用户证件信息维护 61](#_Toc54714478)

[3.8.1 用户行驶证信息添加或修改 61](#_Toc54714479)

[3.8.1.1 接口说明 61](#_Toc54714480)

[3.8.1.2 接口url 62](#_Toc54714481)

[3.8.1.3 接口入参 62](#_Toc54714482)

[3.8.1.4 请求示例 62](#_Toc54714483)

[3.8.1.5 接口出参 62](#_Toc54714484)

[3.8.1.6 返回示例 62](#_Toc54714485)

[3.8.2 用户车辆行驶证信息查询 63](#_Toc54714486)

[3.8.2.1 接口说明 63](#_Toc54714487)

[3.8.2.2 接口url 63](#_Toc54714488)

[3.8.2.3 接口入参 63](#_Toc54714489)

[3.8.2.4 请求示例 63](#_Toc54714490)

[3.8.2.5 接口出参 63](#_Toc54714491)

[3.8.2.6 返回示例 63](#_Toc54714492)

[3.9 生物识别 64](#_Toc54714493)

[3.9.1 用户生物识别信息添加 64](#_Toc54714494)

[3.9.1.1 接口说明 64](#_Toc54714495)

[3.9.1.2 接口url 64](#_Toc54714496)

[3.9.1.3 接口入参 64](#_Toc54714497)

[3.9.1.4 请求示例 64](#_Toc54714498)

[3.9.1.5 接口出参 64](#_Toc54714499)

[3.9.1.6 返回示例 65](#_Toc54714500)

[3.9.2 用户生物识别信息校验 65](#_Toc54714501)

[3.9.2.1 接口说明 65](#_Toc54714502)

[3.9.2.2 接口url 65](#_Toc54714503)

[3.9.2.3 接口入参 65](#_Toc54714504)

[3.9.2.4 请求示例 65](#_Toc54714505)

[3.9.2.5 接口出参 65](#_Toc54714506)

[3.9.2.6 返回示例 65](#_Toc54714507)

[3.10 三方绑定相关 66](#_Toc54714508)

[3.10.1 三方授权登录接口（非登录态） 66](#_Toc54714509)

[3.10.1.1 接口说明 66](#_Toc54714510)

[3.10.1.2 接口url 66](#_Toc54714511)

[3.10.1.3 接口入参 66](#_Toc54714512)

[3.10.1.4 请求示例 66](#_Toc54714513)

[3.10.1.5 接口出参 66](#_Toc54714514)

[3.10.1.6 返回示例 67](#_Toc54714515)

[3.10.2 三方绑定登录接口(非登录态) 67](#_Toc54714516)

[3.10.2.1 接口说明 67](#_Toc54714517)

[3.10.2.2 接口url 68](#_Toc54714518)

[3.10.2.3 接口入参 68](#_Toc54714519)

[3.10.2.4 请求示例 68](#_Toc54714520)

[3.10.2.5 接口出参 68](#_Toc54714521)

[3.10.2.6 返回示例 68](#_Toc54714522)

[3.10.3 查询用户是绑定三方账号（登录态） 68](#_Toc54714523)

[3.10.3.1 接口说明 69](#_Toc54714524)

[3.10.3.2 接口url 69](#_Toc54714525)

[3.10.3.3 接口入参 69](#_Toc54714526)

[3.10.3.4 请求示例 69](#_Toc54714527)

[3.10.3.5 接口出参 69](#_Toc54714528)

[3.10.3.6 返回示例 69](#_Toc54714529)

[3.10.4 三方解绑（登录态） 69](#_Toc54714530)

[3.10.4.1 接口说明 69](#_Toc54714531)

[3.10.4.2 接口url 70](#_Toc54714532)

[3.10.4.3 接口入参 70](#_Toc54714533)

[3.10.4.4 请求示例 70](#_Toc54714534)

[3.10.4.5 接口出参 70](#_Toc54714535)

[3.10.4.6 返回示例 70](#_Toc54714536)

[3.10.5 三方绑定（登录态） 70](#_Toc54714537)

[3.10.5.1 接口说明 71](#_Toc54714538)

[3.10.5.2 接口url 71](#_Toc54714539)

[3.10.4.3 接口入参 71](#_Toc54714540)

[3.10.4.4 请求示例 71](#_Toc54714541)

[3.10.4.5 接口出参 71](#_Toc54714542)

[3.10.4.6 返回示例 71](#_Toc54714543)

[4 错误码 72](#_Toc54714544)

# 文档介绍

## 文档目的

本文档主要介绍用户微服务包含的主要功能模块，以及每个功能模块的技术设计。

## 适用范围

研发部设计、开发、测试和所有被授权人员

## 术语与解释

|  |  |
| --- | --- |
| **缩写、术语** | **解 释** |
| ms-tsp-user | 用户相关服务 |

## 文档约定

* 接口通用约定

后台提供GET和POST两种类型的请求;基于restful风格接口设计规范。后端返回采用统一的json数据结构，必定包含respCode,respMsg,respData三个字段.结构示例如下：

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:null

}

* 错误码约定

错误码为字符串类型，共5位，分成两个部分：错误产生来源+四位数字编号。 说明：错误产生来源分为A/B/C，A表示错误来源于用户，比如参数错误，用户安装版本过低，用户注册错误等问题；B表示错误来源于当前系统，往往是业务逻辑出错，或程序健壮性等问题；C表示错误来源于第三方服务，比如极光推送服务出错，消息投递超时等问题；四位数字编号从0001到9999，大类之间的步长间距预留100。

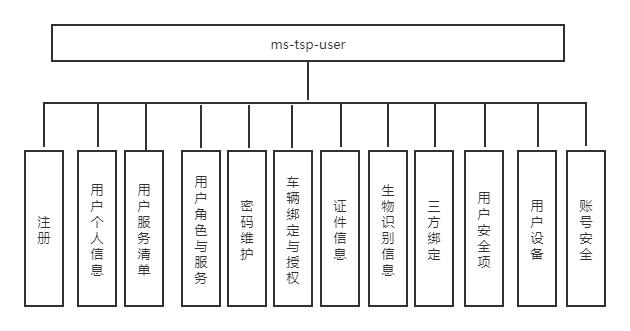
* 客户端请求约定

现将tsp用户提供的所有的服务接口分为登录态和非登录态；

登录态需要在访问后端接口时，客户端将需要在http header中的Authorization添加登录返回的资源访问令牌accessToken后访问后端接口。非登录态不需要资源访问令牌。

# 用户服务设计

## 系统功能模块图



**图1 系统功能模块图**

## 系统功能模块介绍

用户微服务主要为to c实现用户个人信息，用户安全，用户与服务的管理，以及打通用户用户与车之间的绑定关系。

本服务主要包括多方式注册，注册信息维护，设备信息维护；用户个人信息查询和维护；用户服务清单查询和维护；用户角色和服务关系维护；用户忘记密码，修改密码等密码维护；用户与车之间关系查询和维护，以及用户对车进行授权；用户个人证件信息维护，用户生物识别信息维护；三方账号登录绑定；用户安全项管理；用户设备维护；用户账号安全等功能模块。

## 功能模块说明

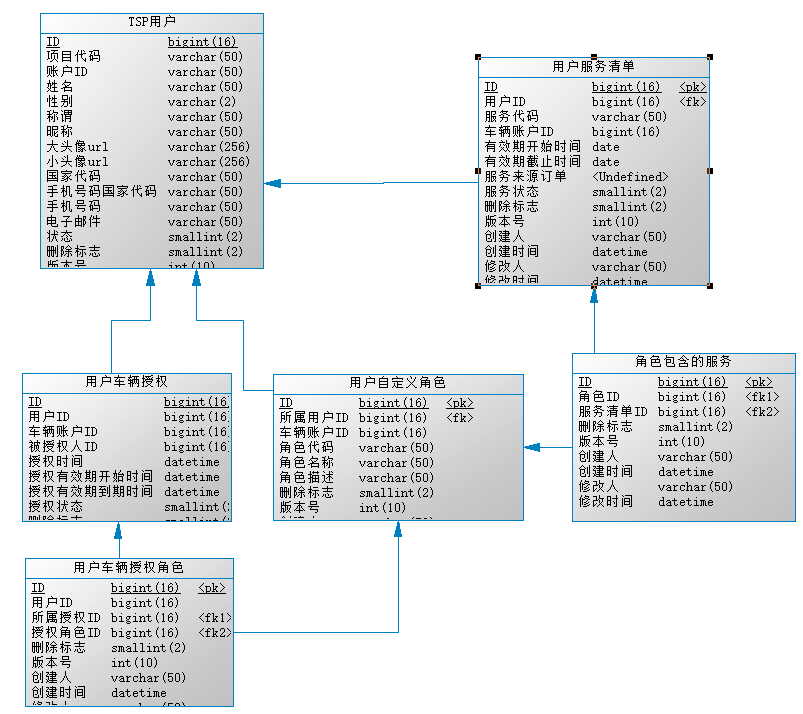
### 用户与终端设备

用户通过终端设备登录成功后，需要调用/v1/userDevice/save接口保存进行设备信息的保存，保存包括设备型号，系统类型，用户在推送平台注册的rid等信息，后端需要对用户设备相关信息进行存储。同时后端需要提供一个微服务接口用户提供车辆控制等微服务进行查询用户最近登录app时对应的推送rid，用于进行消息推送。

### 2.3.2 用户与服务

用户可以购买服务套餐，进而为用户生成服务清单；在登录时，需要查询用户服务清单，并将服务清单缓存至redis缓存中，当用户访问系统时，由鉴权模块根据当前用户访问资源来进行服务资源权限判断，从而为不同用户提供不同服务。

* + 1. 用户,角色与服务



**图4 用户角色与服务**

为了便于授权管理，用户可以根据自己的喜好创建自己的角色，并给角色添加对应的服务清单；当用于给车授权给其他人时，可以直接给对应的人授权指定的角色，这样被授权人就拥有了用户对应的服务访问权限。

### 2.3.3 用户授权关系

用户通过对自己拥有车辆进行授权给其他人的操作，并指定被授权人的角色等；被授权人就拥有了被授予的相应的服务权限。

### 2.3.4 用户安全

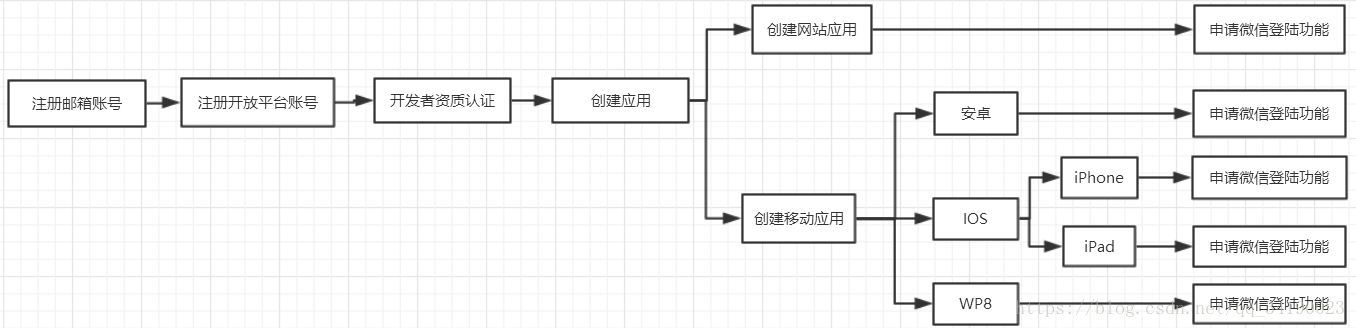
为了保障用户的账号安全和在平台内部的安全操作；安全项功能需要提供用户设置自己的常用安全项，在进行一些高风险操作时，根据用户设置的安全项系统将流程引导到用户进行二次验证。为防止账号被盗，密码遗忘或者泄露的风险，提供用户根据安全问题找回密码等功能。

### 2.3.5 用户三方登录和绑定

三方登录主要实现打通平台内用户和其他实现oauth2.0协议平台的用户，系统接入三方登录需要建立本系统用户和第三方系统用户的绑定关系。第三方登录是基于用户在第三方平台上已有的账号和密码来快速完成己方 应用的登录或者注册的功能。而这里的第三方平台，一般是已经拥有大量用户的平台，国外的比如Facebook，Twitter等，国内的比如微博、微信、QQ等。 第三方登录的目的是使用用户在其他平台上频繁使用的账号，来快速登录己方产品，也可以实现不注册就能登录，好处就是登录比较快容捷，不用注册。

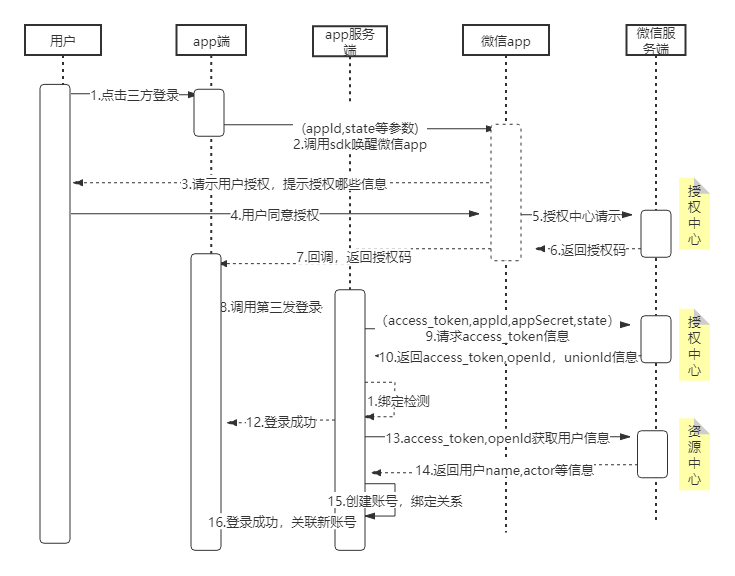
接入流程：

1）开发者资质申请，申请开发者资质后获取三方平台的appId,appSecret等开发者账号，配置三方回调地址。



**图5 开发者资质申请流程**

2）技术实现流程



**图6 三方登录绑定流程**

三方绑定通常设计及一个三方绑定关系表，包括用户id,第三方平台openId的绑定关系。这样在用户在三方平台完成授权后，在本系统检测用户是否完成三方绑定，如果没有绑定需要将流程导向绑定页，提示用户进行绑定关系的建立。否则本系统获取三方平台返回的openId后，通过绑定关系表，查询到用户的基本信息，从而完成整个登录的过程。

# 接口设计

## 注册相关

### 3.1.1 注册接口

#### 3.1.1.1 接口说明

用户注册接口。

#### 3.1.1.2 接口url

POST /v1/user/register

#### 3.1.1.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| account | string | Y | 用户名-手机号 |
| password | String | N | 密码，前端需要进行md5后传给后端,当registerType=1时必填,前端需要md5后传给后端 |
| authcode | String | N | 短信验证码，当registerType=2时必填 |
| registerType | int | Y | 注册类型：1:手机密码注册，2：手机短信注册 |
| deviceType | int | Y | 设备类型  1-andorid  2-ios |
| appType | int | N | 应用类型；1-app |

#### 3.1.1.4 请求示例

{

“account”:”18027888111”,

“password”:”ae27a4b4821b13cad2a17a75d219853e”,

“registerType”:1,

“deviceType”:1

}

#### 3.1.1.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| accessToken | string | Y | 鉴权token,客户端登录拿到此token后需要缓存，每次调用后端接口需要在header中增加此token |
| refreshToken | String | Y | 当后端返回A0302错误码时，用此数据调用后端提供的tokenRefresh接口进行token重刷 |

#### 3.1.1.6 响应示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:{

“accessToken”:”jwtToken”,

“refreshToken”:”jwtToken”

}

}

## 用户基本信息维护

### 3.2.1 添加用户信息（登录态）

#### 3.2.1.1 接口说明

添加用户信息接口,用于用户首次通过验证码注册时完善个人信息。

#### 3.2.1.2 接口url

POST /v1/user/save

#### 3.2.1.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| password | String | Y | 密码，前端需要进行md5后传给后端 |
| headImg | String | N | 用户图像 |
| nickname | string | Y | 用户昵称 |
| gender | int | N | 性别：1-男2-女 |
| ~~region~~ | ~~string~~ | ~~Y~~ | ~~地区~~ |
| province | String | N | 省 |
| city | String | N | 市 |
| area | String | N | 区 |

#### 3.2.1.4 请求示例

{

“password”:”xxx”,

“headImg”:”url”,

“nickname”:”test”,

“gender”:1,

“regionId”:12

}

#### 3.2.1.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.2.1.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”,

“respData”:null

}

### 3.2.2 查询用户信息接口（登录态）

#### 3.2.2.1 接口说明

查询用户信息接口。

#### 3.2.2.2 接口url

GET /v1/user/get

#### 3.2.2.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.2.2.4 请求示例

http://10.24.152.26:8888/tspUser/v1/user/get

#### 3.2.2.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| uid | string | Y | 用户accountId |
| nickname | string | N | 用户昵称 |
| gender | int | N | 1-男;2-女 |
| phone | string | Y | 手机号码 |
| headImg | string | N | 图像 |
| headImgUrl | String | N | 图像URL |
| ~~region~~ | ~~string~~ | ~~N~~ | ~~地区id~~ |
| province | string | N | 省 |
| city | string | N | 市 |
| area | string | N | 区 |

#### 3.2.2.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”,

“respData”:{

“uid”:”uuid str”,

“phone”:”18310411954”

}

}

### 3.2.3 修改用户信息接口（登录态）

#### 3.2.3.1 接口说明

修改用户信息接口，包括头像、昵称、手机号变更

#### 3.2.3.2 接口url

POST /tspUser/v1/user/update

#### 3.2.3.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| headImg | String | N | 更改后头像---经过前端压缩处理后传入，如果不传不做处理 |
| gender | int | N | 1-男，2女 |
| nickname | String | N | 更改后昵称 |
| ~~region~~ | ~~String~~ | ~~N~~ | ~~地区编码,如果不传不做处理~~ |
| province | String | N | 省 编码 |
| city | String | N | 市 编码 |
| area | String | N | 区 编码 |

#### 3.2.3.4 请求示例

{

“headImg”:”xxx”,

“gender”:1,

“nickname”,”昵称”,

“region”:”10000”

}

#### 3.2.3.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.2.3.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”,

“respData”:null

}

### 3.2.4 校验用户名接口

#### 3.2.4.1 接口说明

校验用户名接口，用户注册前检查

#### 3.2.4.2 接口url

GET /v1/user/checkAccount

#### 3.2.4.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| account | string | Y | 用户名-手机号 |

#### 3.2.4.4 请求示例

http://10.24.152.26:8888/tspUser/v1/user/checkAccount?account=18310411954

#### 3.2.4.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| isExist | boolean | Y | 如果存在返回true,否则返回false |

#### 3.2.4.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:{

“isExist”:true

}

}

### 3.2.5 修改密码接口（登录态）

#### 3.2.5.1 接口说明

在登录状态下修改用户密码

#### 3.2.5.2 接口url

POST /v1/user/changePassword

#### 3.2.5.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| oldPassword | String | N | 原密码,初始化密码不传，更改密码必传 |
| newPassword | String | Y | 新密码 |

#### 3.2.5.4 请求示例

{

“oldPassword”:”ae27a4b4821b13cad2a17a75d219853e”,

“newPassword”:”ce27a4b4821b13cad2a17a75d219853e”

}

#### 3.2.5.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.2.5.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”,

“respData”:null

}

### 3.2.6 找回密码接口

#### 3.2.6.1 接口说明

用户通过手机发送验证码(调用3.4发送验证码)找回密码

#### 3.2.6.2 接口url

POST /v1/user/resetPassword

#### 3.2.6.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| newPassword | String | Y | 新密码 |
| userName | String | Y | 用户名-手机号 |
| authToken | string | N | 验证码验证通过后获取的令牌 |
| authcode | string | N | 验证码，和authToken不能同时为空 |

#### 3.2.6.4 请求示例

{

“userName”:”187666263”,

“newPassword”:”ce27a4b4821b13cad2a17a75d219853e”,

“authcode”:”xxx”

}

#### 3.2.6.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.2.6.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”,

“respData”:null

}

### 3.2.7重绑手机（登录态）

#### 3.2.7.1 接口说明

重新绑定手机号码

接入说明：

1. 登录态下调用发送验证码短信接口，验证码类型为：2-更改手机号验证码
2. 调用校验验证码接口进行校验，返回验证令牌
3. 调用发送验证码短信接口，验证码类型为：4-重绑手机号验证码
4. 调用此接口进行新手机号的绑定

#### 3.2.7.2 接口url

POST /v1/user/rebindPhone

#### 3.2.7.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| authToken | string | Y | 验证码校验令牌 |
| phone | string | Y | 新手机号码 |
| authcode | String | Y | 新手机验证码 |

#### 3.2.7.4 请求示例

{

“authToken”:”xxxxxxxxxx”,

“phone”:”18012341234”,

“authcode”:”123456”

}

#### 3.2.7.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.2.7.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:null

}

### 3.2.8校验用户是否设置密码（登录态）

#### 3.2.8.1 接口说明

登录态校验用户是否设置密码

#### 3.2.8.2 接口url

GET /v1/user/checkPassword

#### 3.2.8.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.2.8.4 请求示例

http://10.24.152.26:8888/tspUser/v1/checkPwd

#### 3.2.8.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| isExist | boolean | Y | 如果已经设置返回true,否则返回false |

#### 3.2.8.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:{

“isExist”:true

}

}

### 3.2.9 用户属性添加与修改(登录态)

#### 3.2.9.1 接口说明

提供用户完善个人信息，包括证件信息，联系地址，家庭住址等信息的保存和维护接口

#### 3.2.9.2 接口url

POST /v1/userProfile/edit

#### 3.2.9.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| id | int | N | id,更新时必填 |
| nationCode | string | N | 国家编码 |
| dateOfBirth | date | N | 出生日期 |
| idType | int | N | 证件类别0 - 身份证1 - 港澳通行证  2 - 护照 |
| idCardNum | string | N | 证件号码 |
| idCardValidFrom | date | N | 证件有效期开始时间 |
| idCardValidTo | date | N | 证件有效期结束时间 |
| idCardImage1Url | string | N | 证件照正面图像 |
| idCardImage2Url | string | N | 证件照反面图像 |
| contactProvince | string | N | 联系地址省 |
| contactCity | string | N | 联系地址市 |
| contactArea | string | N | 联系地址区 |
| contactAddress | string | N | 详细地址 |
| familyProvince | string | N | 家庭住址省 |
| familyCity | string | N | 家庭住址市 |
| familyArea | string | N | 家庭住址区 |
| familyAdress | string | N | 家庭详细住址 |
| urgentContactPerson | string | N | 紧急联系人 |
| urgentContactPhone | string | N | 紧急联系电话 |

#### 3.2.9.4 请求示例

{

“contactProvince”:”广东省”,

“contactCity”:”深圳市”,

“contactArea”:”龙华区”,

“contactAdress”:”宝能科技园7栋B座”

}

#### 3.2.9.5 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.2.9.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:null

}

### 3.2.10 用户属性信息查看（登录态）

#### 3.2.10.1 接口说明

获取用户基础属性信息，包括用户的证件信息，联系地址信息和用户家庭住址，紧急联系人等信息

#### 3.2.10.2 接口url

GET /v1/userProfile/get

#### 3.2.10.2 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.2.10.4 请求示例

http://10.210.100.18:8888/tspUser/v1/userProfile /get

#### 3.2.10.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| id | long | N | 记录id |
| nationCode | string | N | 国家编码 |
| dateOfBirth | date | N | 出生日期 |
| idType | int | N | 证件类别0 - 身份证1 - 港澳通行证  2 - 护照 |
| idCardNum | string | N | 证件号码 |
| idCardValidFrom | date | N | 证件有效期开始时间 |
| idCardValidTo | date | N | 证件有效期结束时间 |
| idCardImage1Url | string | N | 证件照正面图像 |
| idCardImage2Url | string | N | 证件照反面图像 |
| contactProvince | string | N | 联系地址省 |
| contactCity | string | N | 联系地址市 |
| contactArea | string | N | 联系地址区 |
| contactAddress | string | N | 详细地址 |
| familyProvince | string | N | 家庭住址省 |
| familyCity | string | N | 家庭住址市 |
| familyArea | string | N | 家庭住址区 |
| familyAdress | string | N | 家庭详细住址 |
| urgentContactPerson | string | N | 紧急联系人 |
| urgentContactPhone | string | N | 紧急联系电话 |

#### 3.2.10.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:{

“id”:1000001,

“accountId”:”string”,

“contactProvince”:”广东省”,

“contactCity”:”深圳市”,

“contactArea”:”龙华区”,

“contactAdress”:”宝能科技园7栋B座”

}

}

### 3.2.11 查询用户证件信息（登录态）

#### 3.2.11.1 接口说明

获取用户证件类型、证件号码信息

#### 3.2.11.2 接口url

GET /v1/user/identification/getCertificateList

#### 3.2.11.2 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.2.11.4 请求示例

http://10.210.100.18:8888/tspUser/v1/user/identification/getCertificateList

#### 3.2.11.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| id | long | N | 记录id |
| idType | int | N | 证件类别0 - 身份证1 - 港澳通行证  2 - 护照 |
| idCardNum | string | N | 证件号码 |
| idCardImage1Url | string | N | 证件照片正面URL |
| idCardImage2Url | string | N | 证件照片反面URL |

#### 3.2.11.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:[{

“id”:1000001,

“idType”:1,

“idCardNum”:”440221199909195354”,

“idCardImage1Url”:”http://xxxx.jpg”,

“idCardImage2Url”:” http://xxxx.jpgss”

},{

“id”:1000002,

“idType”:2,

“idCardNum”:”s”,

“idCardImage1Url”:”http://xxxx.jpg”,

“idCardImage2Url”:” http://xxxx.jpgss”

}]

}

### 3.2.12 用户列表获取

#### 3.2.12.1 接口说明

获取用户列表信息

#### 3.2.12.2 接口url

GET /v1/user/list

#### 3.2.12.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.2.12.4 请求示例

http://10.210.100.18:8888/tspUser/v1/user /list

#### 3.2.12.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.2.12.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:null

}

### 3.2.13 提交实名认证（登录态）

#### 接口说明

首先查询云端是否存储了用户实名证件照信息(3.2.10接口)，如果不存在，需要根据绑定的时候选择的证件类型上传相关证件照片，并完成活体视频录制后提交进行实名认证

#### 接口url

POST /v1/vehicleBind/submitBind

#### 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| frontImgFile | File | N | 证件正面照片 |
| backImgFile | File | N | 证件反面照片 |
| invoiceFile | File | Y | 活体视频 |

#### 请求示例

http://10.210.100.18:8888/tspUser/v1/user/submitUserAuth

#### 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 返回示例

{

"respCode": "00000",

"respMsg": "成功",

"respTime": 1600996035

}

其它respCode说明：

|  |  |
| --- | --- |
| respCode | 说明 |
| A1108 | 用户证件照识别不通过 |
| A1109 | 活体检测验证不通过 |

## 用户设备

### 3.3.1 用户设备信息上报接口（登录态）

#### 3.3.1.1 接口说明

登录完成后客户端需要调用此接口完成设备信息的上报

接入说明：

1. 登录
2. 客户端获取设备的相关信息，包括三方推送的rid进行设备信息的上报
3. 后端进行设备信息的存储

#### 3.3.1.2 接口url

POST /v1/deviceInfo/save

#### 3.3.1.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| deviceName | string | N | 设备名 |
| deviceBrand | string | N | 设备品牌 |
| deviceModel | string | N | 设备型号 |
| osType | int | N | 1-android;2-ios |
| osVersion | string | N | 操作系统版本号 |
| deviceId | string | N | 设备id |
| pushRid | string | N | 推送rid |

#### 3.3.1.4 请求示例

{

“deviceName”:”xxxxxxxxxx”,

“deviceBrand”:”xxxxxxxxxx”,

“deviceModel”:“iphone 6s”,

“osType”:2,

“osVersion”:”ios8.4”,

“deviceId”:”aabbccc”,

“pushRid”:”xxxxxxxxxx”

}

#### 3.3.1.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.3.1.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:null

}

### 3.3.2 获取用户推送rid接口（非登录态）

#### 3.3.2.1 接口说明

提供内部服务获取用户推送rid，用于进行第三方推送

#### 3.3.2.2 接口url

GET /v1/deviceInfo/get

#### 3.3.2.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| accountId | string | Y | 用户账户id |

#### 3.3.2.4 请求示例

http://10.210.100.18:8888/tspUser/v1/deviceInfo /get

#### 3.3.2.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| phone | string | Y | 用户手机号码 |
| deviceName | string | N | 设备名 |
| deviceBrand | string | N | 设备品牌 |
| deviceModel | string | N | 设备型号 |
| osType | int | N | 1-android;2-ios |
| osVersion | string | N | 操作系统版本号 |
| deviceId | string | N | 设备id |
| pushRid | string | N | 推送rid |

#### 3.3.2.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:{

“pushRid”:”xxx”,

“osType”:1

}

}

### 3.3.3 获取用户最近使用的设备信息（非登录态）

#### 3.3.3.1 接口说明

提供内部服务调用，根据用户手机号获取最近使用app设备信息

#### 3.3.3.2 接口url

GET /mgt/user/app/device

#### 3.3.3.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| phone | string | Y | 用户手机号码 |

#### 3.3.3.4 请求示例

http://10.210.100.18:8888/tspUser/mgt/user/app/device?phone=135xxxx

#### 3.3.3.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| deviceName | Sttring | N | 设备名 |
| deviceBrand | string | N | 设备品牌 |
| deviceModel | string | N | 设备型号 |
| osType | int | N | 1-android;2-ios |
| osVersion | string | N | 操作系统版本号 |
| deviceId | string | N | 设备id |
| pushRid | string | N | 推送rid |

#### 3.3.3.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:{

“pushRid”:”xxx”,

“osType”:1

}

}

## 用户车辆

### 3.4.1 根据车辆查询绑定用户

#### 3.4.1.1 接口说明

根据vin获取当前绑定的用户

#### 3.4.1.2 接口url

GET/inner/userVeh/getUserInfo

#### 3.4.1.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| vin | string | Y | 车辆唯一标识 |
| desensitizedFlag | boolean | N | 是否脱敏 默认false |

#### 3.4.1.4 请求示例

http://10.210.100.18:8888/tspUser[/inner/userVeh/getUserInfo](http://localhost:8889/swagger-ui.html#/operations/%E6%9E%81%E5%85%89%E6%8E%A8%E9%80%81%E6%8E%A5%E5%8F%A3/bindRegistrationIdUsingPOST)

#### 3.4.1.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| uid | string | Y | 用户唯一标识 |
| phone | string | Y | 手机号码 |
| userName | string | N | 用户真实姓名 |
| bindTime | string | N | 绑定时间(yyyy/MM/dd) |
| authFlag | string | N | 是否已实名认证 |

#### 3.4.1.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:{

“uid”:”xxxx”,

“phone”:”18310411921”,

“userName”:”张三”,

"bindTime": "2020/05/13",

"authFlag": "已认证"

}

}

### 3.4.2 获取用户绑定车辆接口（登录态）

#### 3.4.2.1 接口说明

获取用户绑定车辆列表

#### 3.4.2.2 接口url

GET /v1/userVeh/list

#### 3.4.2.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.4.2.4 接口请求示例

http://10.210.100.18:8888/tspUser[/v1/userVeh/list](http://localhost:8889/swagger-ui.html#/operations/%E6%9E%81%E5%85%89%E6%8E%A8%E9%80%81%E6%8E%A5%E5%8F%A3/bindRegistrationIdUsingPOST)

#### 3.4.2.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| vehicles | List<Vehicle> | N | 车辆信息列表 |

Vehicle车辆系信息

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| veh\_img | string | Y | 车辆图片 |
| vehLicense | String | Y | 车辆牌照 |
| vehModel | string | Y | 车辆型号 |
| vin | string | Y | 车辆唯一标识 |
| id | Long | Y | 用户车辆绑定关系id |
| status | int | Y | 绑定状态   1. 未绑定 2. 已绑定 3. 已解绑 4. 绑定中 5. 解绑中 |

#### 3.4.2.6 响应示例

**实时返回实体：**

{

"respCode": "00000",

"respMsg": "成功",

"respData":[ {

" veh\_img ": “https://image.baidu.com/search/detail?ct=503316480”,

" vehLicense ": “粤B NB888”,

" vehModel ": “xxxx”,

" vin ": “xxxx”,

“id”:1,

"status": 1

},

{

" veh\_img ": “https://image.baidu.com/search/detail?ct=503316481”,

" vehLicense ": “粤B NB888”,

" vehModel ": “xxxx”,

" vin ": “xxxx”,

“id”:1,

"status": 1

}

]

}

### 3.4.3 绑定用户车辆

#### 3.4.3.1 接口说明

给用户绑定车辆，创建用户车辆账户,header中需要传递用户uid

#### 3.4.3.2 接口url

POST /v1/userVeh/bind

#### 3.4.3.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| vin | string | Y | 车辆唯一标识 |

#### 3.4.3.4 请求示例

{

“vin”:”xxx”

}

#### 3.4.3.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.4.3.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:null

}

### 3.4.4解绑用户车辆

#### 3.4.4.1 接口说明

解除用户和车辆的绑定关系,http header中需要传递用户uid

#### 3.4.4.2 接口url

POST /v1/userVeh/unbind

#### 3.4.4.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| vin | sting | Y | 车辆唯一标识 |
| unbindType | int | Y | 解绑类型  0 - 车辆过户  1 - 车辆退回  3 - 车辆报废  4 - 车辆丢失  5 - 其他 |
| unbindReason | stirng | N | 解绑原因 |
| unbindChannel | int | Y | 解绑渠道  1-APP  2-小程序  3-服务店  4-企业账号 |

#### 3.4.4.4 请求示例

{

“vin”:”xxx”,

“unbindType”:1,

“unbindReason”:”车辆退回”,

“unbindChannel”:1

}

#### 3.4.4.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.4.4.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:null

}

### 3.4.5查询历史车主

#### 3.4.5.1 接口说明

根据vin号查询历史车主列表

#### 3.4.5.2 接口url

GET /inner/userVeh/historyBindings

#### 3.4.5.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| vin | sting | Y | 车辆唯一标识 |
| desensitizedFlag | boolean | N | 是否脱敏标识 默认false |

#### 3.4.5.4 请求示例

/inner/userVeh/historyBindings?vin=200910314110001

#### 3.4.5.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.4.5.6 返回示例

{

"respCode": "00000",

"respMsg": "成功",

"respData": [

{

"uid": "10086",

"phone": "13649869091",

"userName": "张三",

"idCardNum": "22",

"updateBy": "王小三",

"updateTime": "2020-06-04T19:12:03",

"reason": null,

"status": 1,

"bindTime": "2020/05/13",

"unbindTime": "2020/06/08"

},

{

"uid": "10086",

"phone": "13649869091",

"userName": "张三",

"idCardNum": "22",

"updateBy": "赵小七",

"updateTime": "2020-06-10T12:23:43",

"reason": "没有原因",

"status": 1,

"bindTime": "2020/06/10",

"unbindTime": "2020/06/10"

}

],

"respTime": 1597043251

}

## 用户服务

### 3.5.1 服务绑定

#### 3.5.1.1 接口说明

批量添加用户服务清单，即车主，车与服务之间建立关系

#### 3.5.1.2 接口url

POST /v1/user/saveService

#### 3.5.1.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| vehicleAccId | int | Y | 车辆账户id |
| services | list<Service> | Y | 服务集合 |

Service对象参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| serviceCode | string | Y | 服务编码 |
| validFrom | date | Y | 有效期开始时间 |
| validTo | date | Y | 有效期结束时间 |
| serviceOrderId | string | N | 订单来源 |

#### 3.5.1.4 请求示例

{

“vehicleAccId”:1,

services:[

{

“serviceCode”:”A00001”,

“validFrom”:” 2020-06-12 10:53:13”,

“validTo”:” 2020-07-12 10:53:13”

}

]

}

#### 3.5.1.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.5.1.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respTime”:” 2020-06-12 10:53:13”

}

### 3.5.2 修改用户服务清单

#### 3.5.2.1 接口说明

批量修改用户服务清单，主要包括启停服务状态

#### 3.5.2.2 接口url

POST /v1/user/updateService

#### 3.5.2.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| services | list<Service> | Y | 服务集合 |

Service对象参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| id | long | Y | 用户服务id |
| serviceStatus | int | Y | 启用停用0-停用；1-正常 |

#### 3.5.2.4 请求示例

{

services:[

{

“id”:1,

“status”: 1

}

]

}

#### 3.5.2.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.5.2.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respTime”:” 2020-06-12 10:53:13”

}

### 3.5.3获取用户服务清单

#### 3.5.3 接口说明

获取用户车辆账户服务清单

#### 3.5.3.2 接口url

GET /v1/user/getServices

#### 3.5.3.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| vehicleAccId | long | Y | 车辆账户id |
| serviceStatus | int | N | 0-停用；1-正常;默认返回全部类型 |

#### 3.5.3.4 请求示例

http://10.210.100.18:8888/tspUser/v1/user/getServices?vehicleAccId=10001

#### 3.5.3.5 接口出参

respData为服务集合对象，对象实体信息如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| id | long | Y | 用户服务id |
| serviceCode | string | Y | 服务代码 |
| serviceName | string | Y | 服务名称 |
| validFrom | date | Y | 有效开始时间 |
| validTo | date | Y | 有效结束时间 |
| serviceStatus | int | Y | 服务状态0-停用；1-正常 |
| serviceOrderId | string | N | 订单来源 |

#### 3.5.3.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:[

{

“id”:1,

“serviceCode”:”A00000”,

“serviceName”:”车控服务”,

“validFrom”:” 2020-06-12 10:53:13”,

“validTo”:” 2020-06-12 10:53:13”,

“serviceStatus”:1

},

{

“id”:2,

“serviceCode”:”A00001”,

“serviceName”:”车控服务”,

“validFrom”:” 2020-06-12 10:53:13”,

“validTo”:” 2020-06-12 10:53:13”,

“serviceStatus”:1

}

],

“respTime”:” 2020-06-12 10:53:13”

}

### 3.5.4用户自定义角色

#### 3.5.4.1 接口说明

用户自行添加角色接口，用于维护用户，车辆账户和角色的关系

#### 3.5.4.2 接口url

GET /v1/role/save

#### 3.5.4.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| vehAccountId | long | Y | 车辆账户id |
| roleName | string | Y | 角色名称 |
| roleDesp | string | N | 角色描叙 |

#### 3.5.4.4 请求示例

{

“vehAccountId”:100001,

“roleName”:”朋友”,

“roleDesp”:”朋友”

}

#### 3.5.4.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.5.4.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respTime”:” 2020-06-12 10:53:13”

}

### 3.5.5获取角色清单

#### 3.5.5.1 接口说明

获取用户自定义车辆账户角色清单

#### 3.5.5.2 接口url

GET /v1/role/list

#### 3.5.5.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| vehAccId | long | Y | 车辆账户id |

#### 3.5.5.4 请求示例

http://10.210.100.18:8888/tspUser/v1/user /list?vehAccId=1

#### 3.5.5.5 接口出参

角色对象集合，角色对象结构如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| roleName | string | Y | 角色名称 |
| roleDesp | string | N | 角色描叙 |
| roleCode | string | Y | 角色代码 |
| roleId | long | Y | 角色id |

#### 3.5.5.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:[

{

“roleName”:”朋友”,

“roleDesp”:”测试角色”,

“roleCode”:”A0001”

“roleId”:1

}，

{

“roleName”:”家人”,

“roleDesp”:”测试角色”,

“roleCode”:”A0002”

“roleId”:2

}，

]

}

### 3.5.6 修改角色

#### 3.5.6.1 接口说明

修改角色，主要包括角色名称，角色描叙

#### 3.5.6.2 接口url

GET /v1/role/update

#### 3.5.6.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| roleId | long | Y | 角色id |
| roleName | string | N | 角色名称 |
| roleDesp | string | N | 角色描叙 |

#### 3.5.6.4 请求示例

{

“roleId”:1,

“roleName”:”朋友”,

“roleDesp”:”测试角色”

}，

#### 3.5.6.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.5.6.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respTime”:” 2020-06-12 10:53:13”

}

### 3.5.7 用户添加服务到角色

#### 3.5.7.1 接口说明

用户选择添加服务到角色，便于分组管理服务

#### 3.5.7.2 接口url

POST /v1/role/addService

#### 3.5.7.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| serviceIds | list<Long> | Y | 服务清单id集合 |
| roleId | long | Y | 角色id |

#### 3.5.7.4 请求示例

{

“roleId”:1,

“serviceIds”:[1,2,3]

}

#### 3.5.7.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.5.7.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respTime”:” 2020-06-12 10:53:13”

}

### 3.5.8 用户选择从角色中删除服务

#### 3.5.8.1 接口说明

用户删除角色绑定的服务

#### 3.5.8.2 接口url

POST /v1/role/removeService

#### 3.5.8.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| serviceIds | list<Long> | Y | 服务清单id集合 |
| roleId | long | Y | 角色id |

#### 3.5.8.4 请求示例

{

“roleId”:1,

“serviceIds”:[1,2,3]

}

#### 3.5.8.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.5.8.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respTime”:” 2020-06-12 10:53:13”

}

### 3.5.9获取用户角色对应的服务清单

#### 3.5.9.1 接口说明

获取用户自定义角色对应的服务清单

#### 3.5.9.2 接口url

GET /v1/role/getServices

#### 3.5.9.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| roleId | long | Y | 角色Id |

#### 3.5.9.4 请求示例

<http://10.210.100.18:8888/tspUser/v1/role/listService?roleId=1>

#### 3.5.9.5 接口出参

respData为服务集合对象，对象实体信息如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| serviceId | long | Y | 用户服务id |
| serviceCode | string | Y | 服务代码 |
| serviceName | string | Y | 服务名称 |
| validFrom | date | Y | 有效开始时间 |
| validTo | date | Y | 有效结束时间 |
| serviceStatus | int | Y | 服务状态0-停用；1-正常 |
| serviceOrderId | string | N | 订单来源 |

#### 3.5.9.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:[

{

“serviceId”:1,

“serviceCode”:”A00000”,

“serviceName”:”车控服务”,

“validFrom”:” 2020-06-12 10:53:13”,

“validTo”:” 2020-06-12 10:53:13”,

“serviceStatus”:1

},

{

“serviceId”:2,

“serviceCode”:”A00001”,

“serviceName”:”车控服务”,

“validFrom”:” 2020-06-12 10:53:13”,

“validTo”:” 2020-06-12 10:53:13”,

“serviceStatus”:1

}

],

“respTime”:” 2020-06-12 10:53:13”

}

## 车辆绑定与授权

### 3.6.1 用户车辆授权

#### 3.6.1.1 接口说明

用户选择车辆进行授权

#### 3.6.1.2 接口url

POST /v1/userVeh/grant

#### 3.6.1.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| vehicleAccId | long | Y | 车辆账户id |
| toUserId | long | Y | 被授权人id |
| validFrom | date | Y | 授权开始时间 |
| validTo | date | Y | 授权结束时间 |
| authToken | String | Y | 二次验证通过令牌 |

#### 3.6.1.4 请求示例

{

“vehicleAddId”:1,

“toUserId”:10001,

“respTime”:” 2020-06-12 10:53:13”,

“respTime”:” 2020-06-12 10:53:13”

}

#### 3.6.1.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.6.1.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respTime”:” 2020-06-12 10:53:13”

}

### 3.6.2 获取用户对车辆授权

#### 3.6.2.1 接口说明

获取用户对车辆进行的授权

#### 3.6.2.2 接口url

GET /v1/userVeh/getGrant

#### 3.6.2.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| vehicleAccId | long | Y | 车辆账户id |

#### 3.6.2.4 请求示例

<http://10.210.100.18:8888/tspUser/v1/userVeh/getGrant?vehicleAccId=1>

#### 3.6.2.5 接口出参

respData为服务集合对象，对象实体信息如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| grantId | long | Y | 授权id |
| toUserId | string | Y | 服务代码 |
| validFrom | date | Y | 授权有效开始时间 |
| validTo | date | Y | 授权有效结束时间 |
| authTime | date | Y | 授权时间 |
| authStatus | int | Y | 授权状态 |

#### 3.6.2.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:

{

“grantId”:1,

“toUserId”:”A00000”,

“validFrom”:” 2020-06-12 10:53:13”,

“validTo”:” 2020-06-13 10:53:13”,

“authTime”:” 2020-06-12 10:53:13”,

“authStatus”:1

},

“respTime”:” 2020-06-12 10:53:13”

}

### 3.6.3 用户车辆授权角色

#### 3.6.3.1 接口说明

用户选择给授权车辆授权角色或回收角色

#### 3.6.3.2 接口url

POST /v1/userVeh/grantRole

#### 3.6.3.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| roleId | long | Y | 角色id |
| grantId | long | Y | 所属授权id |
| authToken | String | Y | 二次验证通过令牌 |

#### 3.6.3.4 请求示例

{

“roleId”:1,

“grantId”:1

}

#### 3.6.3.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.6.3.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respTime”:” 2020-06-12 10:53:13”

}

### 3.6.4 获取用户授权车辆授权的角色

#### 3.6.4.1 接口说明

获取用户对授权车辆的授权角色

#### 3.6.4.2 接口url

GET /v1/userVeh/getGrantRole

#### 3.6.4.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| grantId | long | Y | 车辆授权id |

#### 3.6.4.4 请求示例

<http://10.210.100.18:8888/tspUser/v1/userVeh/getGrantRole?grantId=1>

#### 3.6.4.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| roleId | long | Y | 角色id |
| roleName | string | Y | 角色名称 |
| roleDesp | string | N | 角色描叙 |

#### 3.6.4.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:

{

“roleId”:1,

“roleName”:”A00000”,

“roleDesp”:” 2020-06-12 10:53:13”

},

“respTime”:” 2020-06-12 10:53:13”

}

## 用户安全相关

### 3.7.1 用户绑定手机

### 3.7.2 用户解绑手机

### 3.7.3 用户换绑手机

### 3.7.4 用户绑定邮箱

### 3.7.5 用户解绑邮箱

### 3.7.6 用户换绑邮箱

### 3.7.7 用户绑定pin

### 3.7.8 用户解绑pin

### 3.7.9 用户安全项绑定查询

#### 3.7.9.1 接口说明

查询当前用户绑定的安全项

#### 3.7.9.2 接口url

GET /v1/userSecurity/checkBind

#### 3.7.9.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.7.9.4 请求示例

http://10.210.100.18:8888/tspUser/userSecurity/checkBind

#### 3.7.9.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| phoneBind | bool | Y | 是否绑定手机 |
| emailBind | bool | Y | 是否绑定邮箱 |
| pinBind | bool | Y | 是否绑定pin码 |
| pinLastUpdate | date | Y | pin码上次更新时间 |

#### 3.7.9.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:{

“phoneBind”:true,

“emailBind”:true,

“pinBind”:false,

“pinLastUpdate”:” 2020-06-12 10:53:13”

},

“respTime”:” 2020-06-12 10:53:13”

}

### 3.7.10 获取系统安全问题

#### 3.7.10.1 接口说明

查询当前系统安全问题

#### 3.7.10.2 接口url

GET /v1/userSecurity/question/list

#### 3.7.10.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.7.10.4 请求示例

http://10.210.100.18:8888/tspUser/userSecurity/question/list

#### 3.7.10.5 接口出参

问题集合列表，问题对象结构如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| id | long | Y | 问题id |
| question | string | Y | 问题 |

#### 3.7.10.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:[

{

“id”:1,

“question”:”我爸爸叫什么？”

},

{

“id”:2,

“question”:”我最喜欢什么动物？”

}

],

“respTime”:” 2020-06-12 10:53:13”

}

### 3.7.11 用户添加安全问题和答案

#### 3.7.11.1 接口说明

用户选择问题，并且设置答案，用于进行安全校验等

#### 3.7.11.2 接口url

POST /v1/userSecurity/answer/save

#### 3.7.11.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| questionId | long | Y | 问题id |
| answer | string | Y | 问题答案 |

#### 3.7.11.4 请求示例

{

“questionId”:1,

“answer”:”xxx”

}

#### 3.7.11.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.7.11.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”,

“respTime”:” 2020-06-12 10:53:13”

}

## 用户证件信息维护

### 3.8.1 用户行驶证信息添加或修改

#### 3.8.1.1 接口说明

用户录入行驶证信息，当id不为空时进行更新，当为空时进行添加

#### 3.8.1.2 接口url

POST /v1/drivingLic/edit

#### 3.8.1.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| licNo | string | N | 行驶证号；插入时不为空，更新时可不填 |
| licDocNo | string | N | 档案号；插入时不为空，更新时可不填 |
| issueOrg | string | N | 签发机关；插入时不为空，更新时可不填 |
| engineNo | string | N | 发动机号；插入时不为空，更新时可不填 |
| vin | string | N | 车架号；插入时不为空，更新时可不填 |
| expireDate | date | N | 到期时间；插入时不为空，更新时可不填 |
| id | long | N | 当id为空时，进行插入；当id不为空时，进行更新 |

#### 3.8.1.4 请求示例

{

“licNo”:”xxx”,

“licDocNo”:”xxx”,

“issueOrg”:”xxx”,

“engineNo”:”xxx”,

“vin”:”xxx”,

“expireDate”:”xxx”,

“id”:1,

}

#### 3.8.1.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.8.1.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”,

“respTime”:” 2020-06-12 10:53:13”

}

### 3.8.2 用户车辆行驶证信息查询

#### 3.8.2.1 接口说明

查询用户车辆对应的行驶证信息

#### 3.8.2.2 接口url

GET /v1/ drivingLic /get

#### 3.8.2.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| vin | string | Y | 车辆唯一标识 |

#### 3.8.2.4 请求示例

http://10.210.100.18:8888/tspUser/v1/drivingLic/get

#### 3.8.2.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| licNo | string | Y | 行驶证号 |
| licDocNo | string | Y | 档案号 |
| issueOrg | string | Y | 签发机关 |
| engineNo | string | Y | 发动机号 |
| vin | string | Y | 车架号 |
| expireDate | date | Y | 到期时间 |
| id | long | Y | id |

#### 3.8.2.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:{

“licNo”:”xxx”,

“licDocNo”:”xxx”,

“issueOrg”:”xxx”,

“engineNo”:”xxx”,

“vin”:”xxx”,

“expireDate”:”xxx”,

“id”:1,

},

“respTime”:” 2020-06-12 10:53:13”

}

## 生物识别

### 3.9.1 用户生物识别信息添加

#### 3.9.1.1 接口说明

添加用户生物识别信息，包括人脸，声纹，指纹等等

#### 3.9.1.2 接口url

POST /v1/biologicalInfo/save

#### 3.9.1.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.9.1.4 请求示例

#### 3.9.1.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.9.1.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”,

“respTime”:” 2020-06-12 10:53:13”

}

### 3.9.2 用户生物识别信息校验

#### 3.9.2.1 接口说明

生物识别信息校验

#### 3.9.2.2 接口url

POST /v1/biologicalInfo/save

#### 3.9.2.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.9.2.4 请求示例

#### 3.9.2.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.9.2.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:null

}

## 三方绑定相关

### 3.10.1 三方授权登录接口（非登录态）

#### 3.10.1.1 接口说明

授权服务端获取用户三方信息，并校验是否进行三方绑定；当用户已经绑定三方账号则自动完成登录操作；否则提示三方未绑定错误码，客户端接收到客户码后需要将页面跳转到账户绑定页。

#### 3.10.1.2 接口url

POST /v1/third/authLogin

#### 3.10.1.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| code | string | Y | 三方授权码 |
| state | string | N | 用于保持请求和回调的状态，授权请求后原样带回给第三方。该参数可用于防止 csrf 攻击（跨站请求伪造攻击），建议第三方带上该参数，可设置为简单的随机数加 session 进行校验 |
| clientId | string | Y | 第三方申请的clientId |
| thirdType | int | Y | 三方类型;1-微信 |

#### 3.10.1.4 请求示例

{

“code”:”xxxx”,

“state”:”xxx”,

“clientId”:”xxx”,

“thirdType”:1

}

#### 3.10.1.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| thirdToken | string | N | 用于绑定和三方登录，当返回三方未绑定的错误码时此字段必填 |
| accessToken | string | N | 资源访问令牌，当用户已经绑定三方账号时，此字段必填 |
| refreshToken | string | N | 认证通过发放的令牌，用于刷新accessToken当用户已经绑定三方账号时，此字段必填 |

#### 3.10.1.6 返回示例

已经绑定三方账号返回实体：

{

“respCode”:”00000”,

“respMsg”:”Success”,

“respData”:{

“accessToken”:”xxxx”,

“refreshToken”:”xxxx”

},

“respTime”:” 2020-06-12 10:53:13”

}

未绑定三方账号返回实体

{

“respCode”:”A1102”,

“respMsg”:”三方账号未绑定”,

“respData”:{

“thirdToken”:”xxxx”,

},

“respTime”:” 2020-06-12 10:53:13”

}

### 3.10.2 三方绑定登录接口(非登录态)

#### 3.10.2.1 接口说明

三方绑定接口,当客户端调用三方授权后如何返回账号没有绑定，则调用此接口进行绑定。

服务端处理流程：

1. 校验是否注册，未注册进行注册
2. 进行绑定
3. 完成登录

#### 3.10.2.2 接口url

POST /v1/third/bindLogin

#### 3.10.2.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| account | string | Y | 用户名（手机号） |
| authcode | string | Y | 验证码 |
| thirdToken | string | Y | 三方auth时返回的token |

#### 3.10.2.4 请求示例

{

“account”:”18310411954”,

“authcode”:”123456”,

“thirdToken”:”xxxxx”

}

#### 3.10.2.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| accessToken | string | Y | 资源访问令牌 |
| refreshToken | string | Y | 认证通过发放的令牌，用于刷新accessToken |

#### 3.10.2.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”,

“respTime”:” 2020-06-12 10:53:13”

}

### 3.10.3 查询用户是绑定三方账号（登录态）

#### 3.10.3.1 接口说明

查询当前登录用户绑定的所有三方账号

#### 3.10.3.2 接口url

GET /v1/third/bindCheck

#### 3.10.3.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.10.3.4 请求示例

http://10.210.100.18:8888/tspUser[/v1/third/bindCheck](http://localhost:8889/swagger-ui.html#/operations/%E6%9E%81%E5%85%89%E6%8E%A8%E9%80%81%E6%8E%A5%E5%8F%A3/bindRegistrationIdUsingPOST)

#### 3.10.3.5 接口出参

返回thirdType集合;1-微信

#### 3.10.3.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”

“respData”:{

“thirdTypes”:[1,2]

}

}

### 3.10.4 三方解绑（登录态）

#### 3.10.4.1 接口说明

三方解绑接口，要求登录态下进行三方账号解绑

#### 3.10.4.2 接口url

POST /v1/third/unbind

#### 3.10.4.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| thirdType | int | Y | 三方类型1-微信 |
| authcode | string | Y | 验证码；发送验证码用6-thirdUnbind;进行二次校验 |

#### 3.10.4.4 请求示例

{

“thirdType”:1,

“authcode”:”123456”

}

#### 3.10.4.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| account | string | Y | 用户名，用于提醒用户记住用户名 |

#### 3.10.4.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”,

“respData”:{

“account”:”18310411954”

},

“respTime”:” 2020-06-12 10:53:13”

}

### 3.10.5 三方绑定（登录态）

#### 3.10.5.1 接口说明

三方绑定接口，用户进行登录态下三方账号的绑定

#### 3.10.5.2 接口url

POST /v1/third/bind

#### 3.10.4.3 接口入参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| code | string | Y | 三方授权码 |
| state | string | N | 用于保持请求和回调的状态，授权请求后原样带回给第三方。该参数可用于防止 csrf 攻击（跨站请求伪造攻击），建议第三方带上该参数，可设置为简单的随机数加 session 进行校验 |
| clientId | string | Y | 第三方申请的clientId |
| thirdType | int | Y | 三方类型;1-微信 |

#### 3.10.4.4 请求示例

{

“code”:”xxxx”,

“state”:”xxx”,

“clientId”:”xxx”,

“thirdType”:1

}

#### 3.10.4.5 接口出参

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |

#### 3.10.4.6 返回示例

{

“respCode”:”00000”,

“respMsg”:”Success”,

“respTime”:” 2020-06-12 10:53:13”

}

# 错误码

|  |  |
| --- | --- |
| 错误码 | 说明 |
| A1001 | 用户被禁用 |
| A1002 | 用户未绑定车辆 |
| A1003 | 用户已解绑该车辆 |
| A1004 | 用户昵称已存在 |
| A1005 | token无效 |
| A1100 | 三方配置未找到 |
| A1101 | 三方账号未绑定 |
| A1102 | 三方账号已绑定 |