

路边停车服务器端通信协议

1. 概述	1
1.1. 目的.....	1
1.2. 阅读对象	1
2. 协议概述	2
2.1. 基础协议	2
2.2. 协议中通信方式.....	2
2.3. 协议基本数据格式	2
3. 系统相关协议	5
3.1. 客户端机器码登记	5
3.2. 认证码获取	5
3.3. 与服务器对时	7
3.4. 客户端配置	8
4. 用户相关协议	10
4.1. 公司街道信息获取.....	10
4.2. 用户类型信息获取.....	11
4.3. 用户列表下载	12
4.4. 用户详细信息下载.....	14
4.5. 用户信息上传	16
4.6. 删除用户	18
4.7. 用户登录登出	18
4.8. 用户密码修改	19
5. 应用相关协议	21
5.1. 车辆分类列表	21
5.2. 车辆停放信息	23
5.2.1. 客户端向服务器提交数据的格式.....	23
5.2.2. 车辆停放信息的返回格式.....	24
5.3. 图片上传下载	28
6. 异常码及含义	30
7. 客户端操作流程	34
8. DETAILED REVISION HISTORY.....	35
CHANGES FROM VERSION 0.50 TO VERSION 0.60.....	35

CHANGES FROM VERSION 0.40 TO VERSION 0.50.....	35
CHANGES FROM VERSION 0.30 TO VERSION 0.40.....	36
CHANGES FROM VERSION 0.20 TO VERSION 0.30.....	36
CHANGES FROM VERSION 0.10 TO VERSION 0.20.....	37
FIRST VERSION 0.10.....	37



1. 概述

1.1. 目的

为方便路边停车 App 的移动设备端与 Web 服务器端之间的通信，制定该协议。

1.2. 阅读对象

路边停车 App 移动设备端开发人员。

路边停车服务器端开发人员。

2. 协议概述

2.1. 基础协议

本协议是建立在 HTTP 协议 1.1 版本(Hypertext Transfer Protocol -- HTTP/1.1)基础之上，制定的应用层协议；HTTP 协议 1.1 版本的具体标准参见 [rfc2616](#) 技术文档。

本协议的数据按照 JSON 数据交换格式组织；JSON 数据交换格式的标准参见 [rfc4627](#) 技术文档。

本协议中的文本数据编码格式按照 UTF-8 编码格式进行编码；UTF-8 的标准参见 [rfc3629](#) 技术文档。

本协议中涉及到的非文本数据按照 BASE64 编码格式编码成文本，再作为文本进行传输；BASE64 的相关标准参加 [rfc1421](#)、[rfc2045](#)、[rfc822](#) 技术文档。

2.2. 协议中通信方式

本协议中，通信的双方为路边停车服务器端软件和路边停车 App 移动设备端软件，分别简称为“服务器”和“客户端”。

通信中，总是由客户端主动发起请求，服务器端返回响应结果的方式进行通信。

客户端采用 POST 请求向服务器提交客户端的数据，服务器收到 POST 请求并处理后返回结果；客户端也可以采用 GET 请求获取服务器端的处理结果。

2.3. 协议基本数据格式

通信中，如果是在 HTTP 协议层出现通信错误，按照 HTTP 错误码标准返回错误原因；如果在 HTTP 层未出现通信错误，则按照如下的数据格式请求和响应数据。

POST 的数据基本格式：

```
{
  "client": "0001",
  "serial": "1234567890",
  "credit": " E2AF32258C4B1ADE398FFBD7E89FFFE271A0EEB5 ",
  "type": "enter",
  "content":
  {
    // Data to be transferred
  }
}
```

在客户端通过 POST 方法提交的数据中，均按照以上的 JSON 封装格式提交数据。其中 client 既可以为客户端机器码，也可以为客户端编号(十进制的数字串)，客户端编号由服务器一次性分配，并保持不变，直到服务器管理员手动修改才会变更。如果恰好存在一个客户端的机器码为十进制数字串，并且该数字恰好与另一个客户端的编号相同，则需要服务器管理员手动调整客户端编号，避开类似的歧义。

serial 为客户端自定义的编号，并且只能用数字、字母和下划线，字母不区分大小写；**serial** 的长度理论上不受限制。无论 **POST** 是否成功，任何两次 **POST** 请求中的 **serial** 必须不相同。

credit 为服务器端分配给客户端的认证码，该认证码由服务器分配给客户端，作为客户端合法 **POST** 请求的凭证，同一个认证码的可使用次数和有效时间是有限制的，该限制由服务器设置并管理(默认值是非常大)，客户端一般无需关心。

鉴于在不同应用环境中对安全的要求不同，服务器对认证码的验证是一个可选过程。通常情况下，服务器不对认证码的合法性进行验证，即 **credit** 可以为任意值。在特别重视安全的应用环境中，如果要验证认证码的合法性，需要由系统管理员对服务器进行设置。

type 为提交的数据类型，服务器根据该类型可以更容易辨别如何处理客户端的请求，具体的类型字段在其后的内容中逐一说明。

如果服务器端收到多于一次 **client** 和 **serial** 均相同的 **POST** 请求，将视为重复 **POST** 请求，将只接纳首次可识别的请求(服务器收到 **credit** 号码最后一个引号并且该 **credit** 号码是合法的视为可识别)，其余的重复请求均被拒绝。

服务器得到 **POST** 数据后，进行处理，然后返回处理结果，同时将处理结果保存，供客户端通过 **GET** 方法获取，因此 **POST** 的返回结果和 **GET** 的结果在格式上是一致的，并且其内容通常也相同。客户端也可以请求资源“?client=客户端&serial=序号”的方式获取服务器对 **POST** 的处理结果。比如 [GET ip?client=001&serial=123 HTTP/1.1](#)，即可获取客户端编号为 001，serial 为 123 的 **POST** 请求的处理结果。服务器对 **POST** 结果的响应数据格式如下。

服务器响应的数据基本格式：

```
{
  "client": "0001",
  "serial": "1234567890",
  "content":
  {
    // Data to be transferred
  },
  "exception":
  {
    "type": "error / warning",
    "code": "404",
    "detail": "page not find"
  }
}
```

其中 **client** 是服务器对该客户端的编号，为一个正整数；如果提交 **POST** 请求的客户端为无效的客户端，则响应中 **client** 字段为空。

serial 与客户端 **POST** 请求时所提交的 **serial** 相同。

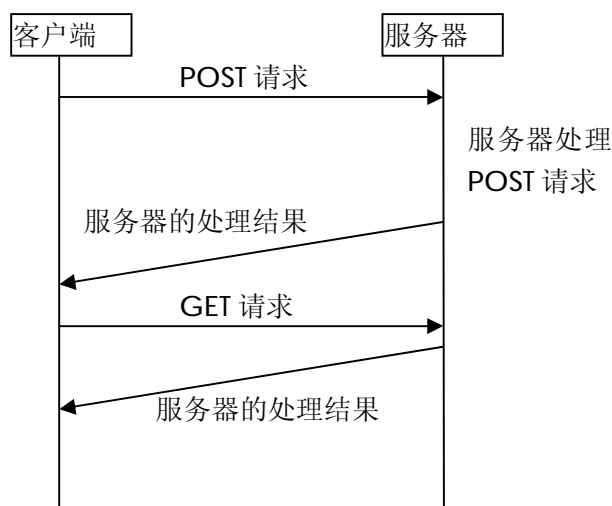
content 为返回的具体内容，**content** 内部的数据格式与 **POST** 中的 **type** 有关。如果没有任何值可返回，**content** 字段有可能不存在。

exception 为非正常 **POST** 的异常报告，其中的 **type** 可取的字段为 **error** 与 **warning**，表示异常的类型；**code** 为异常代码，任何两个不同含义的 **code** 均不相同，无论 **type** 的值是否相同；**detail** 是 **code** 的详细解释，为一个字符串。

content 字段和 **exception** 属于可选字段，即有可能某个字段不存在，但至少存在一个字段。

客户端只需遵循 **POST** 请求在 **GET** 请求之前即可，**GET** 处理结果的顺序可以是任意的。客户端可以多次对同一 **POST** 请求 **GET** 处理结果，但每次 **GET** 的结果内容可能不相同，详细信息参见各具体协议的描述。服务器只对 **POST** 请求的处理结果在较长的一段时间段内保存，之后服务器可能会删除该结果，客户端再通过 **GET** 请求获取时将会得不到任何结果。

交互的过程如下图所示。



3. 系统相关协议

3.1. 客户端机器码登记

客户端只有正确的经过软件注册才可以使用，软件注册是在服务器端由管理员完成，但在客户端数量众多的情况下，手动输入客户端机器码是很不方便的，需要由客户端提交自己的机器码到服务器，方便服务器的管理员进行管理。

客户端机器码登记的 POST 提交格式：

```
{
  "client": "deviceId",
  "serial": "serial",
  "credit": "",
  "type": "register",
  "content":
  {
    "device": "deviceId"
  }
}
```

类型字段"type"固定为 register，表示客户端进行机器码登记；字段"content"只包含 device 一个子字段，为客户端机器码。

客户端机器码登记的响应格式：

```
{
  "client": "client number",
  "serial": "serial ",
  "content":
  {
    "device": "deviceId",
    "order": " client number "
  }
}
```

类型字段"type"固定为 register，表示客户端进行机器码登记。如果登记成功或之前已登记过，字段"content"包含"device"和"order"两项，分别表示客户端机器码及服务器对该客户端的编号；如果登记失败，则只返回"exception"字段。

3.2. 认证码获取

因为一些特定的原因，比如在需要特别考虑安全的环境中，或为了服务器方便验证通信的合法性，

这时客户端需要从服务器端获取认证码，用于通信中向服务器提供合法的身份证明。

鉴于在不同应用环境中对安全的要求不同，认证码获取是一个可选过程。通常情况下，服务器不对 POST 请求中的认证码进行合法性验证。

认证码获取协议只包含两部分：认证码获取请求、认证码获取请求的响应。客户端请求认证码时，通过 POST 方式向服务器提交请求数据。

客户端请求认证码 POST 提交格式：

```
{
  "client": " deviceId / client number ",
  "serial": " serial ",
  "credit": "credit code",
  "type": "credit",
  "content":
  {
    "subtype": "request",
    "limit": "1000",
    "encryption": "none"
  }
}
```

其中的 **credit** 用客户端已有的合法认证码，**type** 为 **credit**，**content** 包含三项子字段：**subtype**、**limit**、**encryption**。**subtype** 固定为 **request**，表示请求获取认证码；**limit** 是一个数字，表示客户端一次可以接纳的认证码数量，以免服务器响应请求时返回巨量认证码使得客户端没有能力处理；**encryption** 是加密的类型，指明返回的认证码使用何种加密方式进行加密，可选的字段有：**none**、**DES**、**3DES**、**AES**、**RSA** 等，其中 **none** 表示不加密，如果 **encryption** 字段不为 **none**，则客户端必须有能力正确解密服务器返回的认证码数据(比如使用相同的密码解密)。

认证码获取请求的响应格式：

```
{
  "client": "client number",
  "serial": "serial",
  "content":
  {
    "sum": "3",
    "list":
    [
      {
        "order": "01",
        "times": "18446744073709551615",
        "value": "1A9FCBF8"
      }
    ]
  }
}
```



```
    },  
    {  
        "order": "02",  
        "times": "18446744073709551615",  
        "value": "2369B092"  
    },  
    {  
        "order": "03",  
        "times": "18446744073709551615",  
        "value": "D3FB4E10"  
    }  
]  
}  
}
```

content 包含两个子项：**sum**、**list**。**sum** 字段为一个数字，指明服务器返回的验证码个数。**list** 为一个列表，其中的每个成员包含 **order**、**times** 和 **value** 三个子项，**order** 是服务器对此次返回的验证码的一个编号，从 1 开始，**times** 是该验证码还可以用多少次，**value** 是与 **order** 对应的验证码。如果获取验证码不成功，则只返回"exception"字段。

3.3. 与服务器对时

为了对停放车辆的正确计时，客户端需要与服务器保持时间一致，并将两者的时钟差异控制在一个很小的范围内（比如 0.5 秒），客户端需要从服务器获取时间，并校正本地时间。

为了方便客户端和服务器进行通信状态监控，客户端可以向服务器报告在线状态。客户端在线信号是一个可选过程，是否提交客户端在线信号由客户端自行决定，服务器无法对其强制要求。

与服务器对时 POST 提交格式：

```
{  
    "client": " deviceId / client number ",  
    "serial": " serial ",  
    "credit": "credit code",  
    "type": "time",  
    "content":  
    {  
        "time": "2012-10-03 12:01:22",  
    }  
}
```

该 POST 数据中，**type** 为 **time**，**content** 只包含 **time** 一项内容，为客户端本机时间。

与服务器对时响应格式:

```
{
  "client": "client number ",
  "serial": " serial ",
  "content":
  {
    "time": "2012-10-03 12:01:22.123456",
  }
}
```

type 为 online, content 只包含 time 一个子项。

如果服务器成功返回, 则 content 只包含 time 一个子项, 为服务器响应的即时时间, 该时间在服务器得到客户端请求时立即生成, 精度为微秒。

如果请求失败, 则只返回"exception"字段。

3.4. 客户端配置

针对不同街道, 服务器会给出一些配置信息, 这些配置信息由服务器统一管理。客户端通过向服务器请求所设置的值, 并根据所得到的值决定其他操作。

客户端记录保留时间 POST 提交格式:

```
{
  "client": " deviceID / client number ",
  "serial": " serial ",
  "credit": "credit code",
  "type": "config",
  "content":
  {
    "street": "001"
  }
}
```

该 POST 数据中, type 为 config, content 只包含 street 一项, 是街道编号。

客户端记录保留时间响应格式:

```
{
  "client": "client number ",
  "serial": " serial ",
  "content":
  {
    "record": " 2592000",
  }
}
```



```
"prepay": " 10.00"  
}  
  
}
```

如果服务器成功返回，则 **content** 包含 **record**、**prepay** 两项，分别是客户端应保留记录的最久时间(以秒为单位)，该街道预收费金额。比如：如果 **record** 为 864000，则客户端可以只保留十天以内的记录，十天之前的记录就可以删除了。

如果请求失败，则只返回**"exception"**字段。

4. 用户相关协议

为了方便通过客户端管理用户信息，客户端可以下载和上传用户信息。用户信息上传与下载的 POST 格式应该符合以下基本格式：

```
{
  "client": " deviceId / client number ",
  "serial": " serial ",
  "credit": "credit code",
  "type": "user ",
  "content":
  {
    "subtype": "list"
    // other info
  }
}
```

type 的值固定为 user，subtype 的值可以为 area、group、list、get、append、modify、delete、login、logout、password，分别表示获取公司街道信息、获取用户类型信息、获取用户列表、获取用户详细信息、添加用户详细信息、修改用户详细信息、删除用户、用户登录、用户登出、修改用户密码。content 中除 subtype 字段外，还可能还有其他字段，具体参见以下各节详细描述。

4.1. 公司街道信息获取

公司信息和街道信息由服务器统一管理，客户端在添加新用户时，需要从服务器获取公司街道信息，供客户端操作者选择。

其 POST 请求中的 content 字段如下：

```
"content":
{
  "subtype": " area"
}
```

其中仅包含 subtype 一个子字段，固定为 area。

用户类型信息获取响应格式：

```
{
  "client": "client number ",
  "serial": " serial ",
  "content":
```

```
{
  "company": "四川前途高科电子有限公司",
  "sum": "2",
  "list":
  [
    {
      "order": "001",
      "city": "四川省成都市",
      "district": "锦江区",
      "street": "春熙路"
    },
    {
      "order": "002",
      "city": "四川省成都市",
      "district": "锦江区",
      "street": "天府广场"
    }
  ]
}
```

如果服务器成功返回结果，则 **content** 中包含三个子项：**company**、**sum**、**list**。**company** 表示该客户端所属的公司名称，**sum** 表示获取的街道数量。**list** 是一个列表，其中每个成员包含四个子字段 **"order"**、**"city"**、**"district"**、**"street"**，分别代表街道编号、街道所属城市名称、街道所属行政区名称、街道名称。

如果服务器没有成功返回结果，则只返回 **"exception"** 字段。

4.2. 用户类型信息获取

用户类型信息获取用于获得服务器对用户进行了哪些分类，其 POST 请求中的 **content** 字段如下：

```
"content":
```

```
{  
    "subtype": "group"  
}
```

其中仅包含 **subtype** 一个子字段，固定为 **group**。

用户类型信息获取响应格式：

```
{  
    "client": "client number ",  
    "serial": " serial ",  
    "content":  
    {  
        "sum": "2",  
        "groups": ["manager","worker"]  
    }  
}
```

如果服务器成功返回结果，则 **content** 中包含两个子项：**sum**、**groups**。**sum** 表示获取的组数量。**groups** 是一个列表，其中每个成员代表类型名。

如果服务器没有成功返回结果，则只返回"exception"字段。

4.3. 用户列表下载

用户列表下载用于客户端获取指定类型用户的一些基本信息，以方便用户登录和客户端离线管理用户登录，其 POST 请求中的 **content** 字段如下：

```
"content":  
{  
    "subtype": " list ",  
    "groups": ["manager ", " worker "]  
}
```

其中 **subtype** 子字段固定为 **list**，**groups** 子字段是一个列表，指出要获取哪些类型的用户列表。

用户列表下载响应格式：

```
{  
    "client": "client number ",  
    "serial": " serial ",
```



```
"content":
{
  "sum": "3",
  "list":
  [
    {
      "user": "me",
      "street": "001",
      "group": "manager",
      "password": " 576E3DEFD5CB258D0514F59F89B91571 ",
      "hash": " D36E060779E9EAF26E9F7DAD28AD8554 ",
      "photo":
      TWFuIGlZlGRpc3Rpbmd1aXNoZWQslG5vdCBvbmh5IGJ5IGhpcyByZWZzb24slGJ1dCBieSB0aGlz
      IHNpbmd1bGFyIHh3Npb24gZnJvbSBvdGhlcjBhbmItYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2Yg
      dGhlIG1pbmQslHRoYXQgYnkgySBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpbIB0aGUy29udGlu
      dWVkiGFuZCBpbmRlZmF0aWdhYmxiIGdlbmVyyXRpb24gb2Yga25vd2xlZGdlLCBleGNIZWRzIHRO
      ZSBzaG9ydCB2ZWwhbWVuY2Ugb2YgYW5lGNhcm5hbCBwbGVhc3VyZS"
    },
    {
      "user": "you",
      "street": "001",
      "group": " worker ",
      "password": " 5AA5AD3BBB153D091D00AE67D51B7DC1",
      "hash": " 5A7069D04E6E04F4A62B52B1327681F1",
      "photo":
      TWFuIGlZlGRpc3Rpbmd1aXNoZWQslG5vdCBvbmh5IGJ5IGhpcyByZWZzb24slGJ1dCBieSB0aGlz
      IHNpbmd1bGFyIHh3Npb24gZnJvbSBvdGhlcjBhbmItYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2Yg
      dGhlIG1pbmQslHRoYXQgYnkgySBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpbIB0aGUy29udGlu
      dWVkiGFuZCBpbmRlZmF0aWdhYmxiIGdlbmVyyXRpb24gb2Yga25vd2xlZGdlLCBleGNIZWRzIHRO
      ZSBzaG9ydCB2ZWwhbWVuY2Ugb2YgYW5lGNhcm5hbCBwbGVhc3VyZS"
    },
    {
      "user": "other",
      "street": "001",
      "group": " worker ",
      "password": " 641E0234E16895DAC578AEBAA4241EDA",
      "hash": " C40D0967B0DA6A4B6F7A47F3DA2D21A6",
      "photo":
      TWFuIGlZlGRpc3Rpbmd1aXNoZWQslG5vdCBvbmh5IGJ5IGhpcyByZWZzb24slGJ1dCBieSB0aGlz
      IHNpbmd1bGFyIHh3Npb24gZnJvbSBvdGhlcjBhbmItYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2Yg
```

```
dGhllG1pbmQslHRoYXQgYnkgySBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpbIB0aGUgY29udGlu
dWVkiGFuZCBpbmRlZmF0aWdhYmxiIGdlbmVyyYXRpb24gb2Yga25vd2xIZGdlLCBleGNIZWRzIHRO
ZSBzaG9ydCB2ZWlhbWVuY2Ugb2YgYW55IGNhcm5hbCBwbGVhc3VyZS"

    }
  ]
}

}
```

如果服务器成功返回结果，则 **content** 中包含两个子项：**sum**、**list**。**sum** 表示获取的用户数量。**list** 是一个列表，其中每个成员包含 **user**、**street**、**group**、**password**、**hash**、**photo** 六个子项，分别代表用户名、用户所服务的街道、用户类型、加密后的用户密码、用户照片校验值、经 BASE64 编码的用户照片。如果该用户没有照片，则 **photo** 和 **hash** 字段为空。

如果服务器没有成功返回结果，则只返回"exception"字段。

4.4. 用户详细信息下载

用户详细信息下载用于客户端获取指定用户的详细信息，以方便通过客户端核对用户的详细信息，其 POST 请求中的 **content** 字段如下：

```
"content":
{
  "subtype": "get ",
  "operator": "me",
  "users": ["me", "you ", " other "]
}
```

其中 **subtype** 子字段固定为 **get**, **operator** 子字段为已成功登录的用户; **users** 子字段是一个列表，指出要获取哪些用户的详细信息。

用户详细信息下载响应格式：

```
{
  "client": "client number ",
  "serial": " serial ",
  "content":
  {
    "sum": "3",
    "list":
    [
      {
        "user": "me",
        "number": "001",
```




```
"name": "张三",
"id": "510104201201011810",
"company": "宝马公司",
"street": "春熙路",
"group": "manager",
"hash": "1A7A023D0C186CCF3E5468E97BD75AB9CAEC96B8",
"photo": "
TWfUlGzlGRpc3Rpbmd1aXNoZWQslG5vdCBvbmh5IGJ5IGhpcyByZWZzb24slGJ1dCBieSB0aGlz
IHNPbmd1bGFyIHh3Npb24gZnJvbSBvdGhlcjBhbmItYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2Yg
dGhlIG1pbmQslHRoYXQgYnkgYSBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpbIB0aGUy29udGlu
dWVklGFuZCBpbmRlZmF0aWdhYmxlGdlbmVYXRpb24gb2Yga25vd2xlZGdlLCBleGNIZWRzIHRo
ZSBzaG9ydCB2ZWhlbWVuY2Ugb2YgYW5lIGNhcm5hbCBwbGVhc3VyZS"
},
{
"user": "you",
"number": "002",
"name": "李四",
"id": "510104201201011810",
"company": "宝马公司",
"street": "春熙路",
"group": "worker",
"hash": "1A7A023D0C186CCF3E5468E97BD75AB9CAEC96B8",
"photo": "
TWfUlGzlGRpc3Rpbmd1aXNoZWQslG5vdCBvbmh5IGJ5IGhpcyByZWZzb24slGJ1dCBieSB0aGlz
IHNPbmd1bGFyIHh3Npb24gZnJvbSBvdGhlcjBhbmItYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2Yg
dGhlIG1pbmQslHRoYXQgYnkgYSBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpbIB0aGUy29udGlu
dWVklGFuZCBpbmRlZmF0aWdhYmxlGdlbmVYXRpb24gb2Yga25vd2xlZGdlLCBleGNIZWRzIHRo
ZSBzaG9ydCB2ZWhlbWVuY2Ugb2YgYW5lIGNhcm5hbCBwbGVhc3VyZS"
},
{
"user": "other",
"number": "003",
"name": "王五",
"id": "510104201201011810",
"company": "宝马公司",
"street": "春熙路",
"group": "worker",
"hash": "1A7A023D0C186CCF3E5468E97BD75AB9CAEC96B8",
"photo": "
TWfUlGzlGRpc3Rpbmd1aXNoZWQslG5vdCBvbmh5IGJ5IGhpcyByZWZzb24slGJ1dCBieSB0aGlz
```

```
        IHNpbmd1bGFyIH8hcn3Npb24gZnJvbSBvdGhlcjBhbmltYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2Yg
        dGhlIG1pbmQsIH8hcn3Npb24gZnJvbSBvdGhlcjBhbmltYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2Yg
        dWVklGFuZCBpbmRlZmF0aWdhYm9lGdlbmVhYXRpb24gb2Yga25vd2lZGdlLCBleGNlZWZlHRo
        ZSBzaG9ydCB2ZW50bWVhY2Ugb2YgYW5lGNhcm5hbCBwbGVhc3VyZS"
    }
  ]
}
}
```

如果服务器成功返回结果，**content** 中包含两个子项：**sum**、**list**。**sum** 表示获取的用户数量。**list** 是一个列表，其中每个成员包含 **user**、**number**、**name**、**id**、**company**、**street**、**group**、**hash**、**photo** 九个子项，分别代表用户名、用户工号、用户姓名、用户有效证件号码、用户所在公司名称、用户服务的街道名称、用户类型、用户照片大小、用户照片校验值、经 **BASE64** 编码的用户照片。

如果该用户没有照片，则 **hash** 和 **photo** 字段为空字符串。

如果服务器没有成功返回结果，则只返回"exception"字段。

4.5. 用户上传信息

如果在客户端新增或修改了用户信息，需要向服务器上传，其 **POST** 请求中的 **content** 字段如下：

```
"content":
{
  "subtype": "append / modify ",
  "operator": "me",
  "users":
  [
    {
      "user": "me",
      "number": "001",
      "name": "张三",
      "id": "510104201201011810",
      "company": "宝马公司",
      "street": "春熙路",
      "group": "manager",
      "hash": "1A7A023D0C186CCF3E5468E97BD75AB9CAEC96B8",
      "photo":
        TWFuIGlZGRpc3Rpbmd1aXNoZWQsIG5vdCBvbmx5IGJ5IGhpcyByZWZzbn24slGJ1dCBleSB0aGlz
        IHNpbmd1bGFyIH8hcn3Npb24gZnJvbSBvdGhlcjBhbmltYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2Yg
        dGhlIG1pbmQsIH8hcn3Npb24gZnJvbSBvdGhlcjBhbmltYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2Yg
        dWVklGFuZCBpbmRlZmF0aWdhYm9lGdlbmVhYXRpb24gb2Yga25vd2lZGdlLCBleGNlZWZlHRo
        ZSBzaG9ydCB2ZW50bWVhY2Ugb2YgYW5lGNhcm5hbCBwbGVhc3VyZS"
```

```
    },  
    {  
        "user": "you",  
        "number": "002",  
        "name": "李四",  
        "id": "510104201201011810",  
        "company": "宝马公司",  
        "street": "春熙路",  
        "group": " worker ",  
        "hash": "1A7A023D0C186CCF3E5468E97BD75AB9CAEC96B8",  
        "photo": "  
            TWFuIGZlGRpc3Rpbmd1aXNoZWQsIG5vdCBvbmh5IGJ5IGhpcyByZWZzb24sIGJ1dCBleSB0aGlz  
            IHNpbmd1bGFyIHh3c3Npb24gZnJvbSBvdGhlcilBhbmltYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2Yg  
            dGhllIG1pbmQsIHRoYXQgYnkgYSBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpbIB0aGUgY29udGlu  
            dWVkiGFuZCBpbmRlZmF0aWdhYmxiIGdlbmV5YXRpb24gb2Yga25vd2xIZGdlLCBleGNIZWRzIHRo  
            ZSBzaG9ydCB2ZWwhbWVuY2Ugb2YgYW55IGNhcm5hbCBwbGVhc3VyZS"  
    },  
    {  
        "user": "other",  
        "number": "003",  
        "name": "王五",  
        "id": "510104201201011810",  
        "company": "宝马公司",  
        "street": "春熙路",  
        "group": " worker ",  
        "hash": "1A7A023D0C186CCF3E5468E97BD75AB9CAEC96B8",  
        "photo": "  
            TWFuIGZlGRpc3Rpbmd1aXNoZWQsIG5vdCBvbmh5IGJ5IGhpcyByZWZzb24sIGJ1dCBleSB0aGlz  
            IHNpbmd1bGFyIHh3c3Npb24gZnJvbSBvdGhlcilBhbmltYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2Yg  
            dGhllIG1pbmQsIHRoYXQgYnkgYSBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpbIB0aGUgY29udGlu  
            dWVkiGFuZCBpbmRlZmF0aWdhYmxiIGdlbmV5YXRpb24gb2Yga25vd2xIZGdlLCBleGNIZWRzIHRo  
            ZSBzaG9ydCB2ZWwhbWVuY2Ugb2YgYW55IGNhcm5hbCBwbGVhc3VyZS"  
    }  
]  
}
```

其中 subtype 子字段为 append 或 modify，分别代表新增用户和修改现有用户信息；operator 子字段为已成功登录的用户；users 子字段是一个列表，指出要添加哪些用户或修改哪些用户的详细信息，users 子字段各项的含义与“用户详细信息下载”返回值中 list 子字段的含义相同。

添加的新用户没有设置密码，新用户可以用任何密码登录成功。

用户信息上传响应格式和“用户详细信息下载”响应格式相同，返回的是完整的用户详细信息。

如果只有部分用户的信息添加或修改成功，则返回值中只包含操作成功的那部分用户，并且会在 `exception` 中给出警告。

4.6. 删除用户

当删除用户时，需要向服务器提交删除用户的请求，其 POST 请求中的 `content` 字段如下：

```
"content":
{
  "subtype": "delete",
  "operator": "me",
  "users": ["me", "you", "other"]
}
```

其中 `subtype` 子字段固定为 `delete`，`operator` 子字段为已成功登录的用户；`users` 子字段是一个列表，指出要获取哪些用户的详细信息。

删除用户响应格式：

```
{
  "client": "client number ",
  "serial": " serial ",
  "content":
  {
    "sum": "3",
    "users": ["me", "you", "other"]
  }
}
```

如果服务器成功返回结果，`content` 中包含两个子项：`sum`、`list`。`sum` 表示删除的用户数量。`users` 是一个列表，代表已经成功删除的用户的用户名。

如果只有部分用户被删除，则返回值中只包含操作成功的那部分用户的用户名，并且会在 `exception` 中给出警告。

4.7. 用户登录登出

用户要在客户端软件上操作某些功能，需要向服务器登录；用户从客户端退出时，需要向服务器发送登出信息，服务器以此作为判断用户操作某些功能的权限之一，并记录用户的登录情况。

用户登录登出 POST 请求中的 `content` 字段如下：

```
"content":
{
  "subtype": "login ",
  "operator": "me",
  "password": "CF3E5468E97BD75AB9C"
}
```

其中 subtype 子字段为 login 或 logout，分别代表用户登录和用户登出；password 为 operator 的登录密码加密后的值，登出时，password 可以为空。

用户登录登出响应格式：

```
{
  "client": "client number ",
  "serial": " serial ",
  "content":
  {
    "detail": "success"
  }
}
```

如果服务器成功返回结果，content 只包含 detail 一个子项，detail 的值为 success，代表登录/登出成功。

如果服务器没有成功返回结果，则只返回"exception"字段。

4.8. 用户密码修改

用户登录成功后，可以修改密码，客户端将密码修改请求发送至服务器，服务器返回操作结果。

新用户添加成功后，新用户也应当及时修改密码，否则有账户被别人盗用的风险。

用户密码修改 POST 请求中的 content 字段如下：

```
"content":
{
  "subtype": "password",
  "operator": "me",
  "password":
  {
    "old": "CF3E5468E97BD75AB9C",
    "new": "2C2BB2CBC4C010EE1E1"
  }
}
```

}

其中 **subtype** 子字段固定为 **password**，**operator** 子字段为已成功登录的用户，如果用户未登录则不会得到成功的返回结果；**password** 包含 **old** 和 **new** 两个子项，分别为 **operator** 的旧登录密码加密后的值和新登录密码加密后的值。

用户密码修改响应格式和用户登录登出响应格式相同。

5. 应用相关协议

5.1. 车辆分类列表

客户端通过向服务器提交查询信息，可以得到满足某些条件的车辆信息。为了得到指定条件的车辆信息，客户端需要向服务器提交查询条件。

车辆分类列表 POST 提交格式：

```
{
  "client": " deviceId / client number ",
  "serial": " serial ",
  "credit": "credit code",
  "type": " sort",
  "content":
  {
    "street": "001",
    "after": "2012-12-31 23:59:59",
    "limit": "100",
    "photos": "0",
    "exit": "yes/no",
    "debt": "yes/no",
  }
}
```

该 POST 数据中，type 为 sort，content 包含 street、after、limit、photos、exit、debt 六项；street 是街道编号，after 指晚于该时间进入车位的车辆，limit 指定客户端可接收的最大记录数量，服务器返回的条目数应不大于该数值，photos 指定拥有几个拍照记录，exit 指定车辆是否离开车位，debt 指定是否结清费用。exit 和 debt 的取值为 yes 和 no，exit 为 yes 或 no 分别表示车辆已离开车位或车辆还在车位上，debt 为 yes 或 no 分别表示停车费用已结清或停车费用未结清(欠费或未足额找零)。

车辆分类列表响应格式：

```
{
  "client": "client number ",
  "serial": " serial ",
  "content":
  {
    "street": "001",
    "sum": "3",
    "list":
    [
```

```
{
  "station": "122",
  "enter": "2013-01-01 00:00:00",
  "exit": "2013-01-01 01:00:00",
  "debt ": "8.5"
  "hash": "1A7A023D0C186CCF3E5468E97BD75AB9CAEC96B8",
  "photo": "
    TWFuGlzlGRpc3Rpbmd1aXNoZWQslG5vdCBvbmh5IGJ5IGhpcyByZWZb24slGJ1dCBieS0aGlz
    IHNpbmd1bGFyIHh3Npb24gZnJvbSBvdGhlcjBhbmItYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2Yg
    dGhlIG1pbmQslHRoYXQgYnkgySBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpb0aGUyY29udGlu
    dWVkiGFuZCBpbmRlZmF0aWdhYm9lGdlbmVhYXRpb24gb2Yga25vd2xlZGdlLCBleGNIZWRzIHRO
    ZSBzaG9ydCB2ZW50bWVhY2Ugb2YgYW55IGNhcm5hbCBwbGVhc3VyZS"
},
{
  "station": "123",
  "enter": "2013-01-01 00:03:00",
  "exit": "2013-01-01 00:58:00",
  "debt": "-0.5"
  "hash": "1A7A023D0C186CCF3E5468E97BD75AB9CAEC96B8",
  "photo": "
    TWFuGlzlGRpc3Rpbmd1aXNoZWQslG5vdCBvbmh5IGJ5IGhpcyByZWZb24slGJ1dCBieS0aGlz
    IHNpbmd1bGFyIHh3Npb24gZnJvbSBvdGhlcjBhbmItYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2Yg
    dGhlIG1pbmQslHRoYXQgYnkgySBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpb0aGUyY29udGlu
    dWVkiGFuZCBpbmRlZmF0aWdhYm9lGdlbmVhYXRpb24gb2Yga25vd2xlZGdlLCBleGNIZWRzIHRO
    ZSBzaG9ydCB2ZW50bWVhY2Ugb2YgYW55IGNhcm5hbCBwbGVhc3VyZS"
},
{
  "station": "138",
  "enter": "2013-01-01 00:02:00",
  "exit": "2013-01-01 02:00:00",
  "debt": "18.5"
  "hash": "1A7A023D0C186CCF3E5468E97BD75AB9CAEC96B8",
  "photo": "
    TWFuGlzlGRpc3Rpbmd1aXNoZWQslG5vdCBvbmh5IGJ5IGhpcyByZWZb24slGJ1dCBieS0aGlz
    IHNpbmd1bGFyIHh3Npb24gZnJvbSBvdGhlcjBhbmItYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2Yg
    dGhlIG1pbmQslHRoYXQgYnkgySBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpb0aGUyY29udGlu
    dWVkiGFuZCBpbmRlZmF0aWdhYm9lGdlbmVhYXRpb24gb2Yga25vd2xlZGdlLCBleGNIZWRzIHRO
    ZSBzaG9ydCB2ZW50bWVhY2Ugb2YgYW55IGNhcm5hbCBwbGVhc3VyZS"
}
]
```



```
}
```

```
}
```

如果服务器成功返回，则 **content** 包含 **street**、**sum**、**list** 三项，**street** 是街道编号，**sum** 是 **list** 中所包含的成员数量，**list** 是一个列表，反应了满足查询条件的车辆基本信息，列表中每个成员均包括 **station**、**enter**、**exit**、**debt**、**hash**、**photo** 六项，分别代表车位号、进入时间、离开时间、所欠费用(为负表示应当找零的金额)、最近拍摄照片的校验值、最近所拍摄照片的 **BASE64** 编码值。其中 **station**、**enter** 的值不为空，其他几项的值根据具体情况可能为空。

如果请求失败，则只返回"exception"字段。

5.2. 车辆停放信息

当有车辆进入或离开停车位、或对车辆拍照、收费时，收费员应当通过客户端向服务器提交相关信息；当需要服务器计算停车费或查询之前收费记录时，需要向服务器提交一些必要的信息。

5.2.1. 客户端向服务器提交数据的格式

车辆停放 POST 提交格式：

```
{
  "client": " deviceId / client number ",
  "serial": " serial ",
  "credit": "credit code",
  "type": "stay",
  "content":
  {
    "subtype":
    "place/move/displace/escape/photo/confirm/prepay/pay/ detail/price",
    "operator": "me",
    "time": "2013-02-06 12:08:00",
    "street": "001",
    "station": "123",
    "enter": "2013-02-06 12:02:02",
    "refer": "122",
    "during": "2013-02-06 12:01:02",
    "judge": "川 A88886",
    "trust": "99.86",
    "plate": "川 A88888",
    "price": "25.00",
    "hash": "1A7A023D0C186CCF3E5468E97BD75AB9CAEC96B8",
    "photo":
```

```
TWFuIGZlGRpc3Rpbmd1aXNoZWQsIG5vdCBvbm5lIGJ5IGhpcyByZWZzb24sIGJ1dCBleSB0aGlz
```

```
IHNpbmd1bGFyIHh3Npb24gZnJvbSBvdGhlcilBhbmItYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2Yg
dGhlIG1pbmQsIHRob3R0YXQgYnkgYSBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpbIB0aGUgY29udGlu
dWVklGFuZCBpbmRlZmF0aWdhYm9lGdlbmVvYXRpb24gb2Yga25vd2xIZGdlLCBleGNIZWRzIHRob3R0
ZSBzaG9ydCB2ZWwhbWVvY2Ugb2YgYW5lGNhcm5hbCBwbGVhc3VyZS*
}
}
```

该 POST 请求中 **type** 为 **stay**, **content** 中至少包含 **subtype**、**operator**、**street**、**station**、**enter** 五项, **street** 是车辆所在街道编号, **station** 是车辆当前所在车位编号, **enter** 是车辆进入当前车位的时间, 服务器根据 **street**、**station**、**enter** 三项唯一确定所提交的信息是针对哪一辆车次。

subtype 可能的取值有: **place**、**move**、**displace**、**escape**、**photo**、**confirm**、**prepay**、**pay**、**detail**、**price**, 分别代表手动确认车辆进入车位(未收到传感器检测结果的情况下)、车辆从一个车位换到另一个车位、手动确认车辆离开车位(未收到传感器检测结果的情况下)、车辆逃费、对车辆拍照、对车辆预收费、对车辆收费、查询车辆详情(出入记录及收费)、查询应补收多少费用。对于 **subtype** 不同的取值, 除 **subtype**、**operator**、**street**、**station**、**enter** 之外, 还必须有某些字段, 多余的字段可有可无, 多余的字段会被服务器忽略。

subtype 为 **place**、**displace**、**escape** 时, 必须有 **time** 字段, 表示车辆进入、离开、逃费的时间。**subtype** 为 **place** 时, **time** 字段的值应当和 **enter** 字段的值相同。

subtype 为 **move** 时, 必须有 **time**、**refer**、**during** 字段, 分别代表操作员确认车辆换位置的时间、车辆之前停放的车位号、车辆在上一车位停放时段内的某一时间; 比如车辆在 2013-02-06 11:01:22 进入 122 号车位, 于 2013-02-06 12:01:22 离开 122 号车位, 并于 2013-02-06 12:02:22 进入 123 号车位, 收费员于 2013-02-06 12:08:00 确认车辆从 122 号车位移动到 123 号车位, 则 **time** 的值为 2013-02-06 12:08:00, **refer** 的值为 122, **during** 的取值不小于 2013-02-06 11:01:22 且不大于 2013-02-06 12:01:22(即车辆在 122 号车位停放时段内的某一个时间)。

subtype 为 **photo** 时, 必须有 **time**、**judge**、**trust**、**hash**、**photo** 字段, 如果车牌被操作员确认, 则还应有 **plate** 字段; **time** 代表拍照的时间, **judge** 代表软件识别出的车牌号码, **trust** 代表软件识别车牌号码的可信度, 以百分比表示(无百分号符号), **hash** 代表所拍摄照片的校验值, **photo** 是所拍摄照片的 BASE64 编码值, **plate** 是经操作员确认后的车牌号码。

subtype 为 **confirm** 时, 必须有 **time**、**plate** 字段, 分别代表确认车牌的时间, 所确认的车牌号。

subtype 为 **prepay**、**pay** 时, 必须有 **time**、**price** 字段, 分别代表预收或收费的时间, 收取的金额。

subtype 为 **detail**、**price** 时, 可以只有必要的 **subtype**、**operator**、**street**、**station**、**enter** 五项。

5.2.2. 车辆停放信息的返回格式

对于 POST 提交格式中 **subtype** 为 **place**、**move**、**displace**、**escape**、**photo**、**confirm**、**prepay**、**pay** 的情况, 均是客户端向服务器提交信息, 在服务器正确接纳的情况下, 服务器只需确认所提交信息

的正确性，因此服务器只需返回如下信息：

```
{
  "client": "client number ",
  "serial": " serial ",
  "content":
  {
    "detail": "success"
  }
}
```

如果服务器成功返回结果，**content** 只包含 **detail** 一个子项，**detail** 的值为 **success**，代表信息提交成功。

如果服务器没有成功返回结果，则只返回**exception**字段。

对于 POST 提交格式中 **subtype** 为 **detail**、**price** 的情况，是客户端向服务器查询某些信息，服务器需要返回客户端所需的信息。

当 POST 提交格式中 **subtype** 为 **price** 时，服务器按如下格式返回：

```
{
  "client": "client number ",
  "serial": " serial ",
  "content":
  {
    "price": "25.00"
  }
}
```

所返回的信息中只包括 **price** 一项，代表实际还应该补收的费用(如果 **price** 为负值，表示应当找零)。

当 POST 提交格式中 **subtype** 为 **detail** 时，服务器按如下格式返回：

```
{
  "client": "client number ",
  "serial": " serial ",
  "content":
  {
    "street": "001",
    "sum": "9",
    "list":
    [
      {
```

26 / 37



dWVklGFuZCBpbmRlZmF0aWdhYmxllGdlbmVYXRpb24gb2Yga25vd2xIZGdlCBleGNlZWZlHRo

ZSBzaG9ydCB2ZWlibWVuY2Ugb2YgYW5lIGNhcm5hbCBwbGVhc3VyZS"

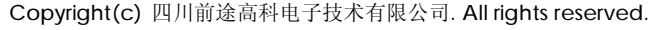
```
},
{
  "subtype": "move ",
  "operator": "me",
  "time": "2013-02-06 12:01:22",
  "station": "123",
  "refer": "122",
  "judge": "",
  "trust": " ",
  "plate": "",
  "price": " ",
  "hash": "",
  "photo":""
},
{
  "subtype": "confim ",
  "operator": "me",
  "time": "2013-02-06 12:02:22",
  "station": "123",
  "refer": "",
  "judge": "",
  "trust": "",
  "plate": "川A88888",
  "price": "",
  "hash": "",
  "photo":""
},
{
  "subtype": "exit/displace/escape ",
  "operator": "me",
  "time": "2013-02-06 13:01:22",
  "station": "123",
  "refer": "",
  "judge": "",
  "trust": "",
  "plate": "",
  "price": "",
  "hash": "",
```

subtype 可能的取值为 enter、exit、place、move、displace、escape、photo、confirm、prepay、pay，enter 和 exit 分别代表系统自动检测到车辆进入和离开，其他值的含义与 POST 中 subtype 的取值的含义相同。

5.3. 图片上传下载

图片传输 POST 提交格式:

28 / 37



6. 异常码及含义

在客户端所提交的信息中，如果服务器正常响应，既无警告也无错误，则服务器会按照“协议基本数据格式”返回包括 **content** 但不包括 **exception** 字段的内容。

如果服务器正常响应，但包括警告信息，则服务器会按照“协议基本数据格式”返回既包括 **content** 字段又包括 **exception** 字段的内容。

如果服务器未能正常响应，则服务器会按照“协议基本数据格式”返回不包括 **content** 字段但包括 **exception** 字段的内容。

协议对异常代码进行了规定，当异常原因以警告形式出现时，异常代码为正值，当异常原因以错误形式出现时，异常代码为对应的负值。比如异常代码 1 所表示的含义为“没有客户端字段”，则以警告形式出现时，**exception** 字段为：

```
"exception":  
{  
  "type": "warning",  
  "code": "1",  
  "detail": "没有客户端字段"  
}
```

当以错误形式出现时，**exception** 字段为：

```
"exception":  
{  
  "type": "error",  
  "code": "-1",  
  "detail": "没有客户端字段"  
}
```

返回的 **exception** 字段中既包括异常代码，也包括异常代码的含义，异常代码及含义如下表所示：

异常代码	含义	适用场合
±1	没有客户端字段	所有
±2	客户端值为空	所有
±3	客户端未登记	所有(除“机器码登记”请求)
±4	客户端被禁用	所有
±5	客户端未归属任何公司	所有(除“机器码登记”请求)
±101	没有请求序号字段	所有
±102	请求序号为空	所有
±103	请求序号重复	所有
±104	请求序号含有非法字符	所有
±105	请求序号太长	所有
±201	没有请求类型字段	所有



±202	请求类型为空	所有
±203	无法识别的请求类型	所有
±301	软件未注册	所有(除“机器码登记”请求)
±302	软件已过期	所有
±303	软件被禁用	所有
±401	没有认证码字段	所有
±402	无效的认证码	所有
±403	认证码已超过使用次数	所有
±501	没有请求内容字段	所有
±502	请求内容为空	所有
±503	没有子类型字段	需要有 subtype 的请求
±504	子类型为空	需要有 subtype 的请求
±505	无法识别的子类型	需要有 subtype 的请求
±506	没有操作者字段	需要有 operator 的请求
±507	操作者为空	需要有 operator 的请求
±508	无法识别的操作者	需要有 operator 的请求
±509	操作者被停用	需要有 operator 的请求
±510	操作者未登录	需要有 operator 的请求
±511	操作者无权执行请求的操作	需要有 operator 的请求
±512	没有照片字段	需要上传照片的请求
±513	没有照片校验码字段	需要上传照片的请求
±514	照片字段为空	需要上传照片的请求
±515	照片校验码字段为空	需要上传照片的请求
±516	不可识别的照片校验码	需要上传照片的请求
±517	照片与校验码不匹配	需要上传照片的请求
±601	没有客户端可接纳的结果数字段	需要给出最大接纳数目的请求
±602	客户端可接纳的结果数为空	需要给出最大接纳数目的请求
±603	客户端可接纳的结果数不为一个数字	需要给出最大接纳数目的请求
±604	客户端可接纳的结果数为零	需要给出最大接纳数目的请求
±701	没有街道编号字段	含有街道编号的请求
±702	无法识别的街道编号	含有街道编号的请求
±801	没有符合条件的结果	向服务器查询的请求
±1001	没有加密类型字段	认证码获取
±1002	不可识别的加密类型	认证码获取
±1101	没有时间字段	与服务器对时
±1101	时间为空或不可识别	与服务器对时
±1201	没有用户类型字段	用户相关协议-用户列表下载
±1202	用户类型字段为空	用户相关协议-用户列表下载

±1203	含有不可识别的用户类型	用户相关协议-用户列表下载
±1204	没有用户列表字段	用户(详细信息下载/信息上传/删除)
±1205	用户列表字段为空	用户(详细信息下载/信息上传/删除)
±1206	含有不可识别的用户名	用户(详细信息下载/信息上传/删除)
±1207	含有缺少员工编号字段的用户信息	用户相关协议-用户信息上传
±1208	含有缺少员工姓名字段的用户信息	用户相关协议-用户信息上传
±1209	含有缺少员工证件号字段的用户信息	用户相关协议-用户信息上传
±1210	含有缺少公司名称字段的用户信息	用户相关协议-用户信息上传
±1211	含有缺少服务街道字段的用户信息	用户相关协议-用户信息上传
±1212	含有缺少员工类型字段的用户信息	用户相关协议-用户信息上传
±1213	含有不可识别员工编号的用户信息	用户相关协议-用户信息上传
±1214	含有不可识别员工姓名的用户信息	用户相关协议-用户信息上传
±1215	含有不可识别员工证件号的员工信息	用户相关协议-用户信息上传
±1216	含有不可识别公司名称的用户信息	用户相关协议-用户信息上传
±1217	含有不可识别服务街道的用户信息	用户相关协议-用户信息上传
±1218	含有不可识别员工类型的用户信息	用户相关协议-用户信息上传
±1219	只有部分用户信息被修改或删除	用户信息上传/删除用户
±1220	没有密码或旧密码字段	用户登录登出/用户密码修改
±1221	加密后的密码或旧密码为空	用户登录登出/用户密码修改
±1222	密码或旧密码错误	用户登录登出/用户密码修改
±1301	没有操作时间字段	车辆停放信息
±1302	操作时间为空	车辆停放信息
±1303	没有当前车位编号字段	车辆停放信息
±1304	当前车位编号为空或不可识别	车辆停放信息
±1305	没有进入当前车位时间字段	车辆停放信息
±1306	进入当前车位的时间为空	车辆停放信息
±1307	没有前继车位编号字段	车辆停放信息
±1308	前继车位编号为空或不可识别	车辆停放信息
±1309	没有前继车位参考时间字段	车辆停放信息
±1310	前继车位参考时间为空或该时间此车位无车	车辆停放信息
±1311	没有软件识别的车牌号字段	车辆停放信息
±1312	软件识别的车牌号为空	车辆停放信息
±1313	没有软件识别车牌的可信度字段	车辆停放信息
±1314	软件识别车牌的可信度为空或为非法值	车辆停放信息
±1315	没有校验车牌字段	车辆停放信息
±1316	校验车牌为空	车辆停放信息
±1317	没有收费金额字段	车辆停放信息
±1318	收费金额为空或不为数字	车辆停放信息
±1319	没有 after 字段	车辆分类列表

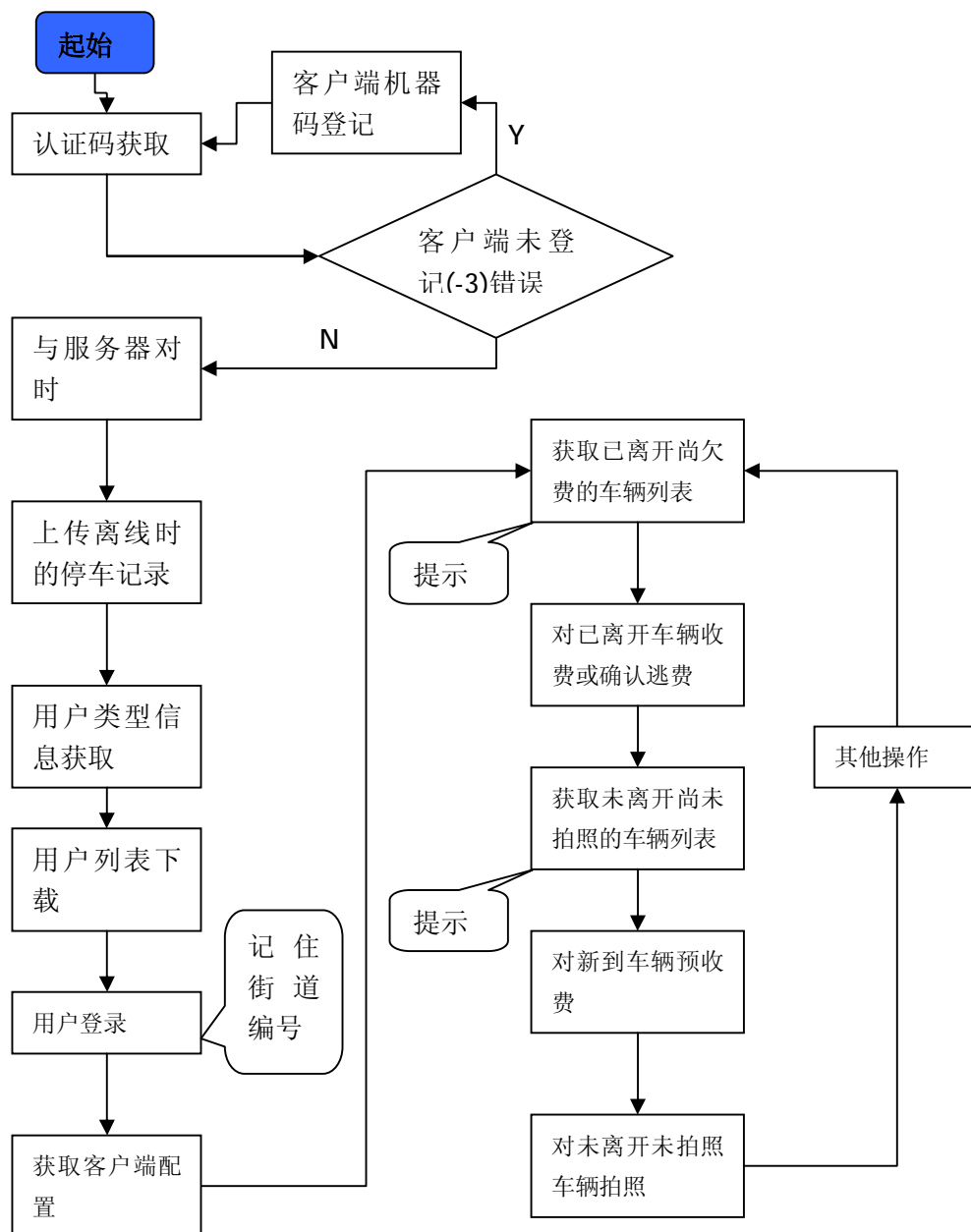


±1320	after 为空或不可识别	车辆分类列表
±1321	没有 photos 字段	车辆分类列表
±1322	photos 为空或不为数字	车辆分类列表
±1323	没有 exit 字段	车辆分类列表
±1324	exit 为空或不符格式	车辆分类列表
±1325	没有 debt 字段	车辆分类列表
±1326	debt 为空或不符格式	车辆分类列表
±1401	没有 bytes 字段	图片传输
±1402	bytes 为空或不是数字或为零	图片传输
±1403	bytes 值与照片实际值不相符	图片传输
±1404	没有所请求的照片内容	图片传输
±1405	所请求的照片字节数太大，无法传输	图片传输

7. 客户端操作流程

根据协议规定的内容，客户端需要按照一定的操作流程，才能尽可能的避免不必要的错误发生，顺利的完成预期的操作，并节约时间和网络流量。

合理的客户端操作流程如下：



8. Detailed Revision History

Changes from Version 0.50 to Version 0.60

- Ø Author: 石洪兴
- Ø Date: 2013-01-18
- Ø Approver: 石洪兴
- Ø Change (+/-) Description
 - a) (/) 根据新增的功能, 改变该协议文档。
 - b) (-) 删除 2.1. 节 “基础协议” 中对传感器协议的描述。
 - c) (-) 删除 3.1. 节 “客户端软件注册”。
 - d) (+) 增加 “客户端机器码登记”。
 - e) (-) 删除 3.3. 节 “客户端在线信号”。
 - f) (+) 增加 “与服务器对时”。
 - g) (+) 增加 “客户端配置”。
 - h) (+) 增加 “公司街道信息获取” 为 4.1.节, 其余章节编号自动调整。
 - i) (+) 增加 “用户类型信息获取” 为 4.2.节, 其余章节编号自动调整。
 - j) (+) 在 4.3.节 “用户列表下载” 的响应格式中增加 “photo” 子字段。
 - k) (-) 删除 “传感器信息传递协议” 章节, 其余章节编号自动调整。
 - l) (/) 重写 “应用相关协议”。
 - m) (/) 修改 “图片上传下载”, 限制照片大小与 bytes 所示一致。
 - n) (+) 增加 “异常码及含义” 为第 6 章, 其余章节编号自动调整。
 - o) (+) 增加 “客户端操作流程” 为第 7 章, 其余章节编号自动调整。

Changes from Version 0.40 to Version 0.50

- Ø Author: 石洪兴
- Ø Date: 2012-12-06
- Ø Approver: 石洪兴
- Ø Change (+/-) Description
 - a) (/) 修改 2.2.节 “协议中通信方式” 中服务器端返回结果的方式。
 - b) (/) 修改 2.3.节 “协议基本数据格式” POST 格式中的 client 格式, 使得允许以机器码为 client。
 - c) (/) 修改 2.3.节 “协议基本数据格式” POST 格式中的 credit 限制, 使得默认情况下客户端无需对 credit 的合法性负责。
 - d) (/) 修改 2.3.节 “协议基本数据格式” 格式的内容, 添加 exception 字段, 并允许 content 字段为可选字段。
 - e) (/) 修改 2.3.节 “协议基本数据格式” 中数据交互示意图。

- f) (/) 修改 3.1. 节“客户端软件注册”部分内容，并允许软件注册过程为客户端的一个可选过程。
- g) (/) 修改 3.1.2. 节“注册请求”和 3.1.4.节“注册确认”部分内容，更改了软件注册过程中验证码的来源方式。
- h) (-) 删除了 3.3.节“客户端连接与断连”部分，其余部分章节的编号同时自动调整。
- i) (/) 修改 3.3. 节“客户端在线信号”的返回数据格式，并且允许“客户端在线信号”为一个可选过程。
- j) (/) 重写“用户相关协议”章节，将用户列表、用户详细信息、用户登录、密码修改等归类为 **user 类型**。
- k) (/) 重写“传感器信息传递协议”章节。
- l) (/) 修改 6.1. 节“车辆进出信息”为“车辆拍照信息”，并修改其 POST 格式及响应结果字段的含义。
- m) (/) 修改 6.3. 节“车辆计费信息”中 POST 格式和响应结果格式，修改后的功能允许对车辆多次收费，直到收费正确为止。

Changes from Version 0.30 to Version 0.40

- Ø Author: 石洪兴
- Ø Date: 2012-11-16
- Ø Approver: 石洪兴
- Ø Change (+/-) Description
 - a) (/) 修改 6.3 节的 POST 格式中 price 字段的单位(“分”变为“元”)。

Changes from Version 0.20 to Version 0.30

- Ø Author: 石洪兴
- Ø Date: 2012-11-13
- Ø Approver: 石洪兴
- Ø Change (+/-) Description
 - a) (+) 在 4.1.2 节的 POST 格式中增加“operator”子字段。
 - b) (/) 修改 4.1.2 节的响应格式中 photo 字段为 photo_bytes 和 hash。
 - c) (-) 合并 4.3 节“登录登出”的两个类型为一个“action”类型，并添加 subtype 以区分具体动作。
 - d) (+) 在 6.1 节的 POST 格式中增加“operator”子字段。
 - e) (+) 在 6.3 节的 POST 格式中增加“operator”子字段。
 - f) (/) 修改 6.4 节的 POST 格式中对 subtype 为 download 的描述，之前的描述是错误的。



Changes from Version 0.10 to Version 0.20

- Ø Author: 石洪兴
- Ø Date: 2012-11-10
- Ø Approver: 石洪兴
- Ø Change (+/-) Description
 - a) (/) 修改 2.3 节括号中对“可识别的请求”的描述。
 - b) (/) 修改 2.3 节对 GET 请求格式的描述。

First Version 0.10

- Ø Author: 石洪兴
- Ø Date: 2012-10-15
- Ø Approver: 石洪兴
- Ø Change (+/-) Description
 - a) (+) 创建文档，起始版本号 0.10