

# REST 架构下的智能共享停车位系统<sup>①</sup>

陆晓天<sup>1</sup>, 葛 艳<sup>1,2</sup>, 郭焕杰<sup>1</sup>, 陶晟宇<sup>3</sup>, 陈 明<sup>2,4</sup>

<sup>1</sup>(上海海洋大学 信息学院, 上海 201306)

<sup>2</sup>(农业部渔业信息重点实验室, 上海 201306)

<sup>3</sup>(上海海洋大学 工程学院, 上海 201306)

<sup>4</sup>(上海海洋大学 现代信息与教育技术中心, 上海 201306)

通讯作者: 葛 艳, E-mail: yge@shou.edu.cn

**摘 要:** 为了有效缓解城市停车一位难求与大量车位利用率低下两者之间的矛盾, 本文从物联网的角度给出了智能分时停车位系统解决方案, 采用 REST 架构将用户手机 app、信息存储服务器和智能车位锁等独立设备有机连接, 支持停车位的实时发现、分时共享、在线预约、智能导航和无人监管, 形成又一个良好的共享经济案例。

**关键词:** 共享车位; REST; 物联网

引用格式: 陆晓天, 葛艳, 郭焕杰, 陶晟宇, 陈明. REST 架构下的智能共享停车位系统. 计算机系统应用, 2018, 27(8): 97-102. <http://www.c-s-a.org.cn/1003-3254/6461.html>

## Intelligent Shared Parking System Based on REST Architecture

LU Xiao-Tian<sup>1</sup>, GE Yan<sup>1,2</sup>, GUO Huan-Jie<sup>1</sup>, TAO Sheng-Yu<sup>3</sup>, CHEN Ming<sup>2,4</sup>

<sup>1</sup>(College of Information Technology, Shanghai Ocean University, Shanghai 201306, China)

<sup>2</sup>(Key Laboratory of Fisheries Information, Ministry of Agriculture, Shanghai 201306, China)

<sup>3</sup>(College of Engineering Science and Technology, Shanghai Ocean University, Shanghai 201306, China)

<sup>4</sup>(Institute of Information and Education Technology, Shanghai Ocean University, Shanghai 201306, China)

**Abstract:** In order to effectively alleviate the contradiction between the parking difficulties and a large number of parking spaces with low utilization, this paper presents a solution to the shared parking system from the perspective of the Internet of Things which adopts REST architecture to connect the independent devices such as users' mobile phone app, information storage server, intelligent parking lock, etc., supporting real time discovery of parking spaces, share time sharing, online booking, smart navigation, and unmanned supervision. This forms another case of sharing economy.

**Key words:** sharing parking; REST; Internet of Things

近年来, 随着我国经济的发展, 中国民用汽车保有量持续高速增长. 据统计<sup>[1]</sup>, 1995 至 2016 年, 中国民用汽车拥有量由 1040 万辆剧烈增加到 19 440 万辆, 年均增长率高达 14.96%. 2012 年, 中国民用汽车保有量更是首次突破 1 亿辆大关. 而大城市市中心的停车场却

已几乎接近饱和, 难以大规模扩大停车位规模. 汽车保有量持续增长与停车位数量停滞之间的矛盾急待解决.

为此, 北京、上海等多个城市推出立体车库<sup>[2]</sup>, 商场车位共享利用<sup>[3]</sup>, 老旧小区改造等提高公共资源利用效率的方案, 取得了一定成效, 但仍有不足, 例如立体车库

① 基金项目: 上海市科技创新行动计划项目(16391902900); 大学生创新项目“易停——快捷停车app”(S201710264107); 上海海洋大学教改项目(A1-0201-00-0322025)

Foundation item: Technological Innovation Project of Shanghai (16391902900); College Student Innovation Project “Easy Parking App” (S201710264107); Revolution Project of Shanghai Ocean University (A1-0201-00-0322025)

收稿时间: 2017-12-05; 修改时间: 2017-12-27; 采用时间: 2018-01-04; csa 在线出版时间: 2018-07-28

的建设成本较高, 停车不便利, 对车主车技有一定要求; 老旧小区改造, 还是受有限的物理空间的限制, 缓解程度很有限; 商场车位共享在居民区相对远离商业区的情况, 可操作性并不强. 相对这些通过增加车位来缓解矛盾的方法之外, 我们还会注意到, 现实生活中还存在大量空闲车位, 因不方便被发现, 或不能分时共享而造成的变相浪费. 于是, 我们提出了一种基于 app 的共享车位系统的解决方案, 配合智能车位锁, 让车位主在车位空

闲时, 将车位信息动态发布, 供有停车需求的人发现, 通过 app 控制智能车位锁, 实现车位分时共享, 及时发现.

### 1 智能车位系统构思

为实现共享车位的信息实时发布、发现、租赁、归还和计费, 在车位主和车位需求者搭起沟通的桥梁, 我们将业务功能分割成如图 1 所示三种不同用户功能. 主要覆盖以下内容:

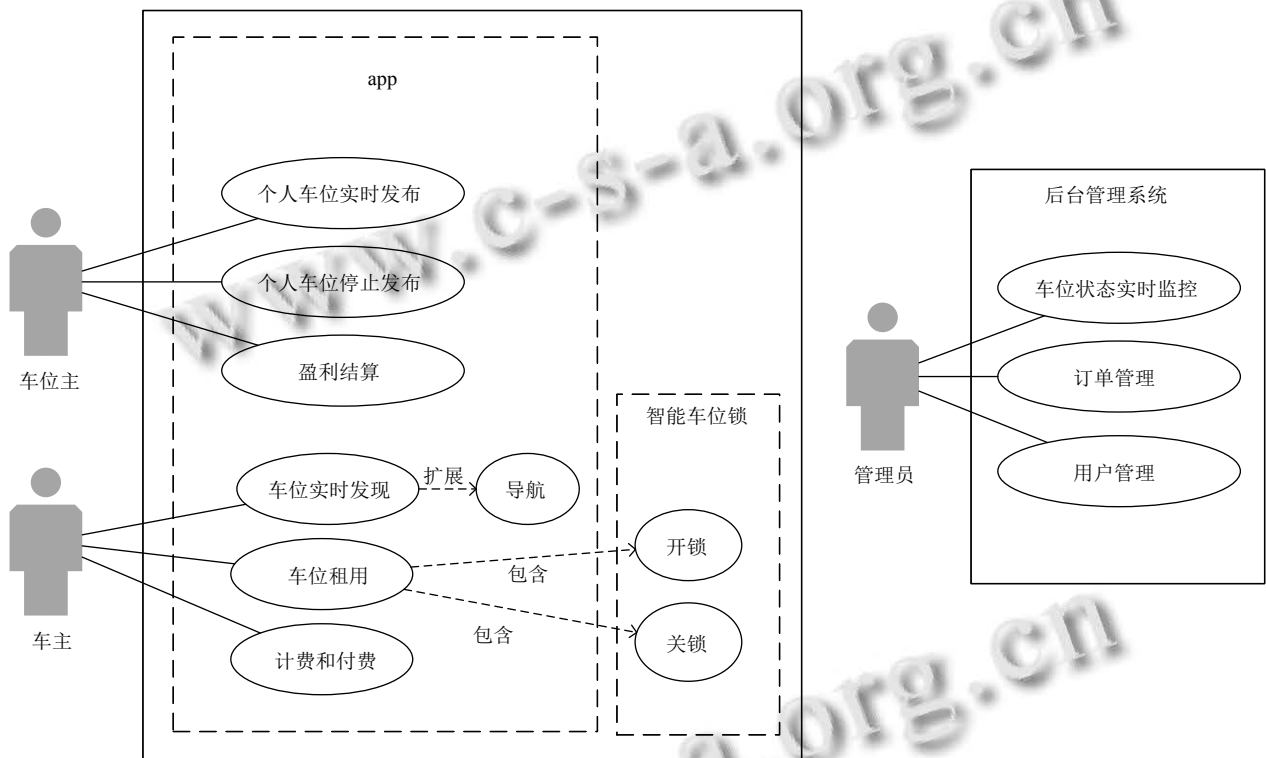


图 1 系统整体用例图

#### 1.1 车位主用例

- (1) 个人车位实时发布: 车位主可以在车位闲置期间, 将车位发布, 供有需要的车主发现可用信息.
- (2) 个人车位停止发布: 车位主不想把车位出租出去时, 在 app 上点击停止发布后, 车位将不再显示在 app 上.
- (3) 盈利结算: 按车主租用费用按一定比例付与车位拥有主.

#### 1.2 车主用例

- (1) 个人车位实时发现: 车位能够在 app 上的地图看到附近可用车位, 并导航到车位位置.
- (2) 车位租用: 车主在 app 上点击开锁即可打开智能车位锁.

- (3) 计费 and 付费: 车主在使用结束后, 在 app 上点击关锁即可锁起智能车位锁, 系统自动按使用时间乘以单价计费计算出此次租用价格, 使用时间为智能车位锁开启的时间到关闭的时间, 此后可利用微信和支付宝等电子支付工具进行支付.

#### 1.3 管理员用例

- (1) 车位状态实时监控: 管理员能够监控各个区域实时的车位情况, 剩余可用车位数量等.
- (2) 订单管理: 数据库会记录所有用户的每次租赁的情况 (例如开始时间, 结束时间, 租用车位号, 总价等等), 如产生异议, 管理员可对其追溯.
- (3) 用户管理: 为了防止不付款等恶意行为, 我们

会给予每个用户信用分评价, 管理员能够将禁止分数过低的用户使用 app 上的功能。

## 2 软件系统架构选型

随着“Web 2.0”概念的提出, Web Services 已经在互联网的方方面面广泛使用, 目前 Web Services 有两种主流架构, 分别是面向服务的 SOAP 架构和面向资源的 REST 架构。

从基本原理上看, SOAP 架构是一个严格定义的信息交换协议, 用于在 Web Service 中封装成机器可读的格式化数据。REST 架构则是一种轻量级的架构, 完全基于 http 协议本身来实现<sup>[4]</sup>。

在具体的协议上, SOAP 架构采用应用程序自定义协议的方式, 其中 WS-\* 系列协议是最为广泛使用的, 但其缺乏统一的标准。REST 架构采用 http 协议。

原生提供的 GET、POST、PUT 和 DELETE 四个方法, 分别正好对应获取、创建、修改和删除四种对资源的操作。并把符合 REST 架构设计的应用编程接口称作 RESTful api。

实现方式上, SOAP 架构是是基于 wsd1 描述, 然后通过 UDDI 注册发布, 从而再被调用者发现, 实现调用。REST 架构通过 url 来定义因特网上独一无二的一个资源, 所谓资源, 就是指网络上一个实体, 或者说是网络上的一个具体信息, 通过设计 RESTful api 来使用四种 http 方法来改变这个资源的状态, 完成相关业务逻辑。

在使用场景上, SOAP 架构适合复杂的非功能性的需求以及端到端之间的交互, 例如银行 ATM 机对数据库中用户金额的读写操作。REST 架构适合无状态, 多个终端之间的简单交互, 例如对数据表单的简单增删查<sup>[5,6]</sup>。

虽然 REST 架构在处理大规模复杂问题上不如 SOAP 架构, 但 REST 架构简单, 系统性能高。数据量越大时 REST 架构的服务性能越优于 SOAP 架构<sup>[7]</sup>。此外, REST 架构具有耦合度低, 易于扩展等优点, 智能共享车位系统具有交互简单, 资源清晰, 状态明确的特点, 非常适合选用 REST 架构。

## 3 基于 REST 架构的共享车位系统逻辑框架

结合架构选型, 综合考虑系统待完成的功能, 本文设计的系统架构如图 2 所示。架构将整个系统的逻辑

结构分为四个组成部分, 分别是: 客户端 app, 后台管理系统, 智能车位锁和中央服务器。

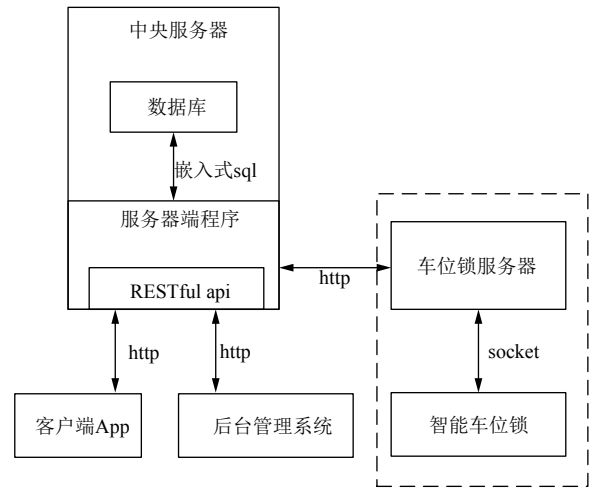


图 2 系统整体逻辑框架

### 3.1 客户端 app

客户端 app 作为提供给车位主和车主的主要终端交互方式, 具有注册登录, 地图上标记车位, 导航, 控制智能车位锁打开/关闭以及显示计费等主要功能。

### 3.2 后台管理系统

后台管理系统是提供给管理员的终端, 实现管理员对车位主, 车主的集中管理。界面实时显示车位状态, 可用车位数量, 订单记录, 用户记录, 并建立相应的数据库保存所有历史记录, 方便运营维护和责任追究。

### 3.3 智能车位锁

智能车位锁是由车位锁服务器和车位锁两个部分组成的。

车位锁采用 Aithinker7 模块, 可以进行 GSM/GPS/GPRS 通信, 实时上传车位信息, 包括设备号 (如 0001), 经度, 纬度, 锁的状态 (0,1), 是锁否被锁定 (0,1)。车位锁的通讯情况较为复杂, 且若在信号不好的地方 (如地下室等) 较容易断开连接, 此时需要重连。

为了降低耦合度, 车位锁不与中央服务器直接通信, 而是与车位锁服务器建立基于 TCP/IP 的 socket 长连接通信。车位锁服务器是车位锁和中央服务器的中介, 将连接的具体细节屏蔽后, 提供开锁和关锁的 RESTful api 给中央服务器调用, 将复杂的通信变为请求-等待-回应的简单模型, 即中央服务器向车位锁服务器请求开锁或关锁, 等待处理, 车位锁服务器返回结果 (成功或失败)。

### 3.4 中央服务器

中央服务器是全系统的核心,提供了数十个 RESTful api 实现与 app、后台管理系统以及车位锁服务器的连接.为了保持数据和状态的统一性,各部分都只和中央服务器交互,由中央服务器统一将数据保存在数据库中.

在 app 侧的 RESTful api 主要实现用户的登陆、短信验证码获取及注册、获得该用户所有订单历史记录、查询附近车位并计算距离、发布车位、取消发布车位、开启智能车位锁并创建订单和关闭智能车锁并结束订单等功能.

在后台管理系统侧,管理员拥有比普通用户更高的权限,这一侧的 RESTful api 主要实现查看订单记录并进行筛选、监视车位状态和查看用户历史记录等功能.

在智能车位锁一侧,也是采用 REST 架构对开锁、关锁这两个核心交互的业务逻辑进行封装和抽象,从而实现在 app 和智能车位锁之间达到了双向透明传输的效果.于是,app 不需要知道智能车位锁具体的通信方式,智能车位锁也不需要知道 app 上层的业务逻辑.以 app 上打开车锁并创建一个订单开始计费的一个智能车位锁侧的 RESTful api 为例,其定义如下:

创建订单的 api

/order [POST]

·请求

-head

·Authorization [必填][token]

-body

·parkingSpace\_id [必填][车位 id]

·回应

-状态码 200: 示例返回数据格式:

```
{
  "result": [ {
    "id": 48,
    "user_id": 2,
    "parkingSpace_id": 1,
    "parkingSpace_name": "1 号",
    "address": "第二小区左侧",
    "price": 10,
    "status": 1,
    "status_name": "已创建",
    "start_time": "2017-07-19
```

22:23:43"

}]

}

-状态码 401 错误 token 错误

-状态码 404 错误 没有这个车位

-状态码 403 错误 车位状态不是空闲

## 4 REST 架构下的核心代码实现

几乎所有主流开发语言都支持 REST 架构,其中 nodejs 是基于事件循环的异步非阻塞 I/O 模式,在以被访问为主要功能的服务器中效率较高,因此我们选用的服务器端程序是基于 nodejs 的 express 框架,在这种框架下,一个 RESTful api 恰好对应一个中间件,每个中间件都是一个独立的模块.

在上述 RESTful api 定义基础上,以下给出部分 api 代码实现.

### 4.1 车位信息获取 api

车位信息获取 api 是在 app 加载完成后自动调用,从而获取信息的.服务器会根据用户请求的区域,以及用户提供的经度和纬度,在数据库查询相应的车位并计算距离,然后按距离从小到大的排序,返回查询结果.

代码示例:

```
var querySql='Select... From ...
```

```
Order By
```

```
calculateDisdance(request.latitude,request.
longitude,latitude,longitude) ;;
```

//根据车位和用户距离排序的 sql 代码,查询部分具体代码省略

```
var [result]= await mysqlQuery(querySql);return
response(result);
```

### 4.2 创建订单的 api

在 app 上点车位开锁,首先 app 会发一个 http 请求给中央服务器,此时中央服务器进行一些业务处理(例如检查是否登陆,检查车位是不是已经被占用了)之后,向车位锁服务器发一个开锁的 http 请求,这时候车位锁服务器会向智能地锁发一个 tcp 包检查当前智能地锁是否在线,是否能打开,如果正确打开,智能地锁向车位锁服务器回发一个 tcp 包,车位锁服务器再向中央服务器返回一个成功的 http 状态,中央服务器将这次行为包装成一个订单后记录进数据库后,向 app 返回一个成功的信息.如果有任何一个环节出错,就会



向 app 返回一个拒绝的信息, 不会开始收费, 这样保障了系统的健壮性.

代码示例:

```
//查询车位状态
var querySql='Select... From ...';
Where parkingSpace.id=request.parkingSpace_id'
var [parkingSpace]=await mysqlQuery(querySql);
//具体没有该车位 id, 返回错误
if(!parkingSpace){
return response('no such parkingSpace');
}
//具体该车位状态不是空闲, 返回错误
if(!parkingSpace.status!=='1'){
return response('status invalid');
}
//向车位锁服务器发送开锁请求
try {await middleServerPost(parkingSpace.lock_
id,'open')}
```

//如果开锁失败, 返回错误

```
catch(e){return response('fail to open')};
```

//向数据库插入新的数据

```
var insertSql='Insert Into order ... Values ...;
Select ... From ... Where id=last_insert_id()';
```

```
var [result]= await mysqlQuery(insertSql);
```

//如果一切正常, 返回正确结果

```
return response(result);
```

#### 4.3 后台管理系统获取地区状态的 api

管理员登录后会定时轮询这个 api 以获得最新的动态情况, 在服务器中计算出每个地区的车位数量及可用车位数量后返回给浏览器.

代码示例:

```
var querySql='Select totalParkingSpace,
availableParkingSpace ... From ...
Where ... Group By region.id';
var [result]= await mysqlQuery(querySql);return
response(result);
```

## 5 原型系统展示

以车主用户为例, 打开 app 后, 附近的可用停车位用 P 图标在地图上标识, 如图 3(a) 初始可用车位标记页面所示.

此时用户可以选择预停的车位, 点击地图上的图标后将自动生成导航路线, 如图 3(b) 所示, 待用户到达指定车位后, 可以看到智能车位锁处于图 4(b) 点击开锁图标即可进入开锁界面, 如图 3(c) 所示. 点击开锁后, 智能车位锁将会自动打开, 如图 4(c) 所示, 用户此时可以实现停车.



图 3 原型 app 界面

停车结束后, 用户将车开走, 然后点击 app 的结束停车图标, 如图 4(a) 所示, 智能车位锁将会自动关闭, 如图 4(b) 所示, 同时结算本次计费, 对接用户账户完成扣款.



图 4 停车界面及车位锁变化

以体育馆门前某一个停车位为例, 原来没有任何信息化管理, 该车位长期处于空闲状态, 增加智能停车锁后, 统计该车位一天内的使用情况, 如图 5 所示, 可

以看出,使用频次明显增加,这在无形中就拓展了可用车位,缓解了车位难求的困境。

订单列表

预订id	用户id	用户名	车位地址	状态	总价	开始时间	结束时间
164	2	yui	体育馆	已完成	11.33	2017-12-20 16:35:24	2017-12-20 18:51:24
163	3	lao	体育馆	已完成	1.61	2017-12-20 15:46:24	2017-12-20 16:05:44
162	5	bob	体育馆	已完成	2.49	2017-12-20 15:06:34	2017-12-20 15:36:24
161	16	sg	体育馆	已完成	7.94	2017-12-20 13:02:35	2017-12-20 14:37:49
160	13	wy	体育馆	已完成	7.78	2017-12-20 10:32:21	2017-12-20 12:05:43

图5 后台管理系统界面

## 6 结束语

本文利用 REST 架构模块化,易扩展,无状态等优点,提供了基于 REST 架构的智能车位锁物联网解决方案,可以将分时可用车位信息集中发布,并共享给手机用户,用户借用手机 app 可以发现,查找,预约和导航到对应车位,并可以实现车位锁开启和闭合控制等操作,有效提升了车位管理水平和车位使用效率。

该系统在今年的全国移动创新大赛,全国 iCAN 大赛决赛以及上海市的创造杯大赛中都取得不错的成绩。在第十九届中国国际工业博览会上展示,也得到了来自物业、停车场行业相关企业以及市民的广泛好评

和热切关注。

## 参考文献

- 胡华龙,李建松,蒋子龙,等.基于地理国情监测数据的城市停车位与潜力估算方法研究.地球信息科学报,2017,19(10):1287-1297.
- 付翠玉,关景泰.立体车库发展的现状与挑战.机械设计与制造,2005,(9):156-157. [doi: 10.3969/j.issn.1001-3997.2005.09.076]
- 康正宁.共享停车的商业模式创新与政策需求分析.科学发展,2017,(5):107-112. [doi: 10.3969/j.issn.1674-6171.2017.05.013]
- Rodriguez A. Restful web services: The basics. Amon, New York: IBM, 2008.
- Zur Muehlen M, Nickerson JV, Swenson KD. Developing web services choreography standards—the case of REST vs. SOAP. Decision Support Systems, 2005, 40(1): 9-29. [doi: 10.1016/j.dss.2004.04.008]
- 王建斌,胡小生,李康君,等. REST 风格和基于 SOAP 的 Web Services 的比较与结合.计算机应用与软件,2010,27(9):297-300. [doi: 10.3969/j.issn.1000-386X.2010.09.093]
- 高攀攀,王健,黄颖,等.互联网上基于 SOAP 和 REST 的 Web 服务的对比分析.小型微型计算机系统,2015,36(11):2417-2421. [doi: 10.3969/j.issn.1000-1220.2015.11.001]