Park接口

—7.12

web 三个客户端

android ?

IOS ?

[普通用户]

1.登录

2.注册

3.用户名是否可用

4.短信验证

5.请求用户基本信息

6.修改用户基本信息

7.消费记录(先忽略,数据挖掘包含这个功能)

8.按范围查询停车场基本信息

9.按街道查询停车场基本信息

10.按停车场id查询停车场

11.车位预定

12.按照语音卡号&时间查询

13.按照车牌号&时间查询

14.新建工单

15.工单获取

16.工单提交

17.完结工单

18.评价工单

[管理员用户]

1.登录

2.请求admin基本信息

3.修改admin基本信息

4.创建停车场

5.按范围查询停车场基本信息

6.按街道查询停车场基本信息

7.修改停车场

8.删除停车场

9.按照语音卡号&时间查询

10.按照车牌号&时间查询

11.获取普通用户信息

12.获取未受理工单

13.获取正在处理工单

14.工单提交

15.查询历史工单

[停车场管理员接口]

1.登录

2.上送停车场数据

3.开始停车

4.预定请求(websocket)

5.解除预定请求(websocket)

6.预定—>已停

7.停车结束

附录1:测试数据

附录2:项目建立(局域网)相关问题:服务器ip: 192.168.1.110

方式: post

通信数据格式（请求+返回）：json

说明:

1.(user和admin)登录是一个url,其他是一个url.

2. IP: 局域网:192.168.1.110

3.不返回报错信息,如果是false,提示重新登录,一般就是被挤下去了.

4.关于距离单位,百度地图上面做法是,>1公里用公里,<1公里用米,所以我在传 的时候使用公里做单位,会保留小数点后3位,也就是米.

获取cookie(username…) —> Smart.getCookie(“username”);

[普通用户]

1.登录

web的url : /ParkT/login

Flex,android : http://IP/Park/clientLogin

请求数据格式:

{

“cmd” : “login”,

“username” : “xxxx”,

“password” : “abc”

}

返回数据格式:

{

“cmd” : “login”,

“res” : “true”

}

2.注册

web的url : / ParkT /rest

Flex,android : http://IP/Park/clientRest

请求数据格式:

{

“cmd” : “register”,

“username” : “xxxx”,

“phone” : “xxxx”

“password” : “abc”

}

返回数据格式:

{

“cmd” : “register”,

“res” : “true”

}

3.用户名是否可用

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式:

{

“cmd” : “ValidUsername”,

“username” : “xxxx”

}

返回数据格式:

{

“cmd” : “ValidUsername”,

“res” : “true” //表示可用

}

4.短信验证(只有发送功能,cookie验证)

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式:

{

“cmd” : “ValidUser”,

“phone” : “xxxx”

}

返回数据格式:

{

“cmd” : “ValidUser”,

“res” : “true”

}

5.请求用户基本信息

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式:

{

“cmd” : “getUser”,

“username” : “xxxx”

}

返回数据格式:

//没有数据就没有返回

{

“cmd” : “getUser”,

“res” : “true”,

“id” : xxxx,

“username” : “xxxx”,

“password” : “xxxx”,

“phone” : “xxxx”,

“realname” : “xxxx”, //真实姓名

“IDcard” : “xxxx”, //身份证

“bankcard” : “xxxx”, //银行卡

“mail” : “xxxx”, //邮箱

“registerTime” : “xxxx”//注册时间

“parkTime” : “xxxx” //停车总时长

“money” : “xxxx” //消费总金额

“bindList” : [

{

“carPlate” : “xxxx”, //车牌号

“voiceCard” : “xxxx” //语音卡

}

…

]

}

6.修改用户基本信息(没修改也进行填写)

//处理方式:将getUser返回数据存储为obj,然后修改obj,返回.

//不需要重新生成对象(注册时间什么的都可以直接返回)

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式:

{

“cmd” : “modifyUser”,

“id” : xxxx, //不能修改,但要上传

“username” : “xxxx”,

“password” : “xxxx”,

“phone” : “xxxx”,

“realname” : “xxxx”, //真实姓名

“IDcard” : “xxxx”, //身份证

“bankcard” : “xxxx”, //银行卡

“mail” : “xxxx”, //邮箱

“bindList” : [

{

“carPlate” : “xxxx”, //车牌号

“voiceCard” : “xxxx” //语音卡

}

…

]

}

返回数据格式:

{

“cmd” : “modifyUser”,

“res” : “true”,

}

7.消费记录(先忽略,数据挖掘包含这个功能)

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

//这个是查这个用户所有的停车记录

请求数据格式:

{

“cmd” : “charging”,

“username” : “xxxx”,

“carPlate” : “xxxx”, //车牌号(具体查那个车牌的,只能是自己的)

“paginationNum” : xx , //页码

“paginationPage” : xx //每页的容量

}

返回数据格式:

{

“cmd” : “charging”,

“res” : “true”,

“chargingList” : [

{

“id” : xxxx, //订单号

“carPlate” : ”粤B88888”, //车牌号

“voiceCard” : ”0088” //语音卡

”name” : ”xxxx”, //停车场名

”citycode” : ”xxxx”, //城市编码

”address” : ”xxxx”, //停车场地址

“reserveTime” : “xxxx”, //预定时间

“startTime”:”xxxx”, //入场时间

“endTime”:”xxxx”, //出场时间

“reserveHour” : xxxx , //预定时长(不满一小时按一小时算)

“parkHour” : xxxx , //停车时长(不满一小时按一小时算)

“reservePrice” : “xxxx”, //预定单价

“parkPrice”: ”xxxx”, //停车单价

“consumption”: ”xxxx” //收费

“state” : “xxxx” //订单状态(false:未完成.true:已完成)

}

…

]

}

8.按范围查询停车场基本信息

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式：

{

“cmd” : “rangeQueryPark”,

“username” : “xxxx”,

“longitude” : 34.123 // double，定位点经度(数据类型要选择对)

“latitude” : 35.0 //double，定位点纬度

“range”:10 //int，范围，单位：公里

}

返回数据格式：

{

“cmd” : “rangeQueryPark”,

“res” : “true”,

“parkList” : [

{

“id” : xxxx, //停车场id

“name” : “xxxx”, //停车场名字

“address” : “xxxx” //停车场地址

“citycode” : “xxxx”,

“longitude” : xxxx // double，定位点经度(数据类型要选择对)

“latitude” : xxxx //double，定位点纬度

“total” : xxxx, //总共车位

“canReserve” : xxxx,//可预定车位

“distance” : xxxx //距离

“reservePrice” : “xxxx”, //预定单价

“parkPrice”: ”xxxx”, //停车单价

“updateTime” //这条数据更新的时间,要显示在web上

}

…

]

}

9.按城市编码查询停车场基本信息

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式：

{

”cmd”:”citycodeQueryPark”,

“username” : “xxxx”,

”citycode”:”110000”, //城市编码

}

返回数据格式：

{

”cmd”:”citycodeQueryPark”,

“res” : “true”,

“parkList” : [

{

“id” : xxxx, //停车场id

“name” : “xxxx”, //停车场名字

“address” : “xxxx”, //停车场地址

“citycode” : “xxxx”,

“longitude” : xxxx, // double，定位点经度(数据类型要选择对)

“latitude” : xxxx, //double，定位点纬度

“total” : xxxx, //总共车位

“canReserve” : xxxx,//可预定车位

“reservePrice” : “xxxx”, //预定单价

“parkPrice”: ”xxxx”, //停车单价

“updateTime” //这条数据更新的时间,要显示在web上

}

…

]

}

10.按停车场id查询停车场（间隔轮询）

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式：

{

”cmd”:”IdQueryPark”,

“username” : “xxxx”,

“id” : xxxx

}

返回数据格式：

{

”cmd”:”IdQueryPark”,

“res” : “true”,

“id” : xxxx, //停车场id

“name” : “xxxx”, //停车场名字

“address” : “xxxx” //停车场地址

“longitude” : xxxx // double，定位点经度(数据类型要选择对)

“latitude” : xxxx //double，定位点纬度

“total” : xxxx, //总共车位

“canReserve” : xxxx,//可预定车位

“updateTime” : xxxx //这条数据更新的时间,要显示在web上

//在服务端更新的时间,查询并不能改变时间.

“reservePrice” : “xxxx”, //预定单价

“parkPrice”: ”xxxx”, //停车单价

}

11.车位预定

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

//“我的预定”这个功能不需要上传也不需要查询,直接在cookie中进行储存

请求数据格式:

{

“cmd” : “reservePark”,

“username” : “xxxx”,

“carPlate” : “xxxx”, //车牌号(主要)

“voiceCard” : “xxxx”, //语音卡

“parkId” : “xxxx” //停车场id

}

返回数据格式:

{

“cmd” : “reservePark”,

“res” : “true”,

}

12.按照语音卡号&时间查询

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式：

{

”cmd”:”voiceCardRecord”

“username” : “xxxx”

“voiceCard” : ”0088” //语音卡

“startTime” : ”20141028” //开始时间

“endTime” : ”20141029” //结束时间

“paginationNum” : xx , //页码

“paginationPage” : xx //每页的容量

}

返回数据格式：{

”cmd”:”carPlateRecord”

“res” : “true”

”recordList”:[

{

//没有这个值,下面也会没有

“id” : xxxx, //订单号

“carPlate” : ”粤B88888”, //车牌号

“voiceCard” : ”0088” //语音卡

”name” : ”xxxx”, //停车场名

”citycode” : ”xxxx”, //城市编码

”address” : ”xxxx”, //停车场地址

“reserveTime” : “xxxx”, //预定时间

“startTime”:”xxxx”, //入场时间

“endTime”:”xxxx”, //出场时间

“reserveHour” : xxxx , //预定时长(不满一小时按一小时算)

“parkHour” : xxxx , //停车时长(不满一小时按一小时算)

“reservePrice” : “xxxx”, //预定单价

“parkPrice”: ”xxxx”, //停车单价

“consumption”: ”xxxx” //收费

“state” : “xxxx” //订单状态(false:未完成.true:已完成)

}…

]

}

13.按照车牌号&时间查询

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式：

{

”cmd”:”carPlateRecord”,

“username” : “xxxx”,

“carPlate” : ”0088”, //车牌号

“startTime” : ”20141028” //开始时间

“endTime” : ”20141029” //结束时间

“paginationNum” : xx , //页码

“paginationPage” : xx //每页的容量

}

返回数据格式：{

”cmd”:”carPlateRecord”

“res” : “true”

”recordList”:[

{

//没有这个值,下面也会没有

“id” : xxxx, //订单号

“carPlate” : ”粤B88888”, //车牌号

“voiceCard” : ”0088” //语音卡

”name” : ”xxxx”, //停车场名

”citycode” : ”xxxx”, //城市编码

”address” : ”xxxx”, //停车场地址

“reserveTime” : “xxxx”, //预定时间

“startTime”:”xxxx”, //入场时间

“endTime”:”xxxx”, //出场时间

“reserveHour” : xxxx , //预定时长(不满一小时按一小时算)

“parkHour” : xxxx , //停车时长(不满一小时按一小时算)

“reservePrice” : “xxxx”, //预定单价

“parkPrice”: ”xxxx”, //停车单价

“consumption”: ”xxxx” //收费

“state” : “xxxx” //订单状态(false:未完成.true:已完成)

}…

]

}

14.新建工单

//这个也可以按步骤来弄

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式:

{

“cmd” : “userNewWorkOrder”,

“username” : “xxxx”,

“title” : “xxxx” //工单标题

}

返回数据格式:

{

“cmd” : “userPutWorkOrder”,

“id” : xxxx, //工单编号

“res” : “true”

}

15.工单获取(时间查询)

//获取的都是自己的工单

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式:

{

“cmd” : “userGetWorkOrder”,

“username” : “xxxx”,

//以开始时间作为标准

“startTime” : ”20141028” //开始时间

“endTime” : ”20141029” //结束时间

}

返回数据格式:

{

“cmd” : “userGetWorkOrder”,

“res” : “true”,

workOrderList : [

{

“id” : “xxxx”, //工单编号

“answer” : “xxxx”, //回答者

“startTime” : “xxxx”, //工单开始时间

“endTime” : “xxxx”, //工单结束时间(没结束就没有endTime)

“assess” : “xxxx”, //评价

“state” : “xxxx”, //工单状态(0:新建,1:提交,2:处理中,3:完成,4已评价)

“messageList” : [

{

“name” : “xxxx”, //消息发送者

//type是为了前台上色方便

“type” : “xxxx”, //用户类型(user&admin两种)

//时间有排序作用

“time” : “xxxx” //发送时间

“message” : “xxxx” //发送内容

},

{

“name” : “xxxx”,

“type” : “xxxx”,

“time” : “xxxx”

“message” : “xxxx”

},

…

]

}

…

}

16.工单提交

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式:

{

“cmd” : “userPutWorkOrder”,

“username” : “xxxx”,

“id” : “xxxx”, //工单编号

“message” : “xxxx” //提交内容

}

返回数据格式:

{

“cmd” : “userPutWorkOrder”,

“res” : “true”

}

17.完结工单

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式:

{

“cmd” : “FinishWorkOrder”,

“username” : “xxxx”,

“id” : “xxxx”, //工单编号

}

返回数据格式:

{

“cmd” : “FinishWorkOrder”,

“res” : “true”

}

18.评价工单

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式:

{

“cmd” : “AssessWorkOrder”,

“username” : “xxxx”,

“id” : “xxxx”, //工单编号

“assess” : “xxxx” //评价

}

返回数据格式:

{

“cmd” : “AssessWorkOrder”,

“res” : “true”

}

[管理员用户] (admin:123,admin123:123)

1.登录

web的url : /Park/login

Flex,android : http://IP/Park/clientLogin

请求数据格式:

{

“cmd” : “adminLogin”,

“username” : “xxxx”,

“password” : “abc”

}

返回数据格式:

{

“cmd” : “adminLogin”,

“res” : “true”

}

2.请求admin基本信息

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式:

{

“cmd” : “getAdmin”,

“username” : “xxxx”,

}

返回数据格式:

{

“cmd” : “getUser”,

“res” : “true”,

“username” : “xxxx”,

“password” : “xxxx”,

“phone” : “xxxx”

“realname” : “xxxx”, //真实姓名

}

3.修改admin基本信息

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式:

{

“cmd” : “modifyAdmin”,

“username” : “xxxx”,

“password” : “xxxx”,

“phone” : “xxxx”

“realname” : “xxxx”, //真实姓名

}

返回数据格式:

{

“cmd” : “modifyAdmin”,

“rest” : “true”,

}

4.创建停车场

//可以用步骤的形式 (3级联动)—> (停车场名,地址)—> (地图上面确定经纬度)—>

//也可以在创建停车场的时候将停车场管理员也一起创建好了.

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式:

{

”cmd” : ”createPark”

“username” : “xxxx”,

”name” : ”xxxx”, //停车场名

”longitude” : 34.678, //double，经度

”latitude” : 35.456, //double，纬度

”citycode” : ” 110000”, //注意：参照中国省市县编码

”address” : ”xxxx”, //地址

“reservePrice” : “xxxx”, //预定单价

“parkPrice”: ”xxxx”, //停车单价

}

返回数据格式：

{

”cmd” : ”createPark”

”res” : ”true”

}

5.按范围查询停车场基本信息(user有)

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式：

{

“cmd” : “rangeQueryPark”,

“username” : “xxxx”,

“longitude” : 34.123 // double，定位点经度(数据类型要选择对)

“latitude” : 35.0 //double，定位点纬度

“range”:10 //int，范围，单位：公里

}

返回数据格式：

{

“cmd” : “rangeQueryPark”,

“res” : “true”,

“parkList” : [

{

“id” : xxxx, //停车场id long

“name” : “xxxx”, //停车场名字

“address” : “xxxx” //停车场地址

“citycode” : “xxxx”, //层级划分

“longitude” : xxxx //double，定位点经度(数据类型要选择对)

“latitude” : xxxx //double，定位点纬度

“total” : xxxx, //总共车位

“remain” : xxxx, //剩下车位

“updateTime” //这条数据更新的时间,要显示在web上

“reservePrice” : “xxxx”, //预定单价

“parkPrice”: ”xxxx”, //停车单价

}

…

]

}

6.按城市编码查询停车场基本信息

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式：{

”cmd”: ”citycodeQueryPark”,

“username” : “xxxx”,

”citycode”:”110000” //层级划分

}

返回数据格式：

{

”cmd”:”citycodeQueryPark”,

“res” : “true”,

“parkList” : [

{

“id” : xxxx, //停车场id

“name” : “xxxx”, //停车场名字

“address” : “xxxx” //停车场地址

“citycode” : “xxxx”, //层级划分

“longitude” : xxxx, // double，定位点经度(数据类型要选择对)

“latitude” : xxxx, //double，定位点纬度

“total” : xxxx, //总共车位

“remain” : xxxx, //剩下车位

“updateTime” //这条数据更新的时间,要显示在web上

“reservePrice” : “xxxx”, //预定单价

“parkPrice”: ”xxxx”, //停车单价

}

…

]

}

7.修改停车场

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式：

{

”cmd” : ”modifyPark”,

“id” : xxxx, //停车场id long

“username” : “xxxx”,

”name” : ”xxxx”, //停车场名

”citycode” : ” 110000”, //参照中国省市县编码(通过联动获取了code在改)

”longitude” : 34.678, //double，经度

”latitude” : 35.456, //double，纬度

”address” : ”xxxx” //地址

“reservePrice” : “xxxx”, //预定单价

“parkPrice”: ”xxxx”, //停车单价

}

返回数据格式:

{

”cmd”:”modifyPark”,

”res”:”true”

}

8.删除停车场

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式：

{

”cmd” : “deletePark”,

“username” : “xxxx”,

“id” : xxxx //停车场id long

}

返回数据格式：

{

”cmd” : ”deletePark”,

”res” : ”true”

}

9.按照语音卡号&时间查询

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式：

{

”cmd” : ”adminVoiceCardRecord”

“username” : “xxxx”

“voiceCard” : ”0088” //语音卡

“startTime” : ”20141028” //开始时间

“endTime” : ”20141029” //结束时间

“paginationNum” : xx , //页码

“paginationPage” : xx //每页的容量

}

返回数据格式：{

”cmd”:”adminVoiceCardRecord”

“res” : “true”

”recordList”:[

{

//没有这个值,下面也会没有

“id” : xxxx, //订单号

“carPlate” : ”粤B88888”, //车牌号

“voiceCard” : ”0088” //语音卡

”name” : ”xxxx”, //停车场名

”citycode” : ”xxxx”, //城市编码

”address” : ”xxxx”, //停车场地址

“reserveTime” : “xxxx”, //预定时间

“startTime”:”xxxx”, //入场时间

“endTime”:”xxxx”, //出场时间

“reserveHour” : xxxx , //预定时长(不满一小时按一小时算)

“parkHour” : xxxx , //停车时长(不满一小时按一小时算)

“reservePrice” : “xxxx”, //预定单价

“parkPrice”: ”xxxx”, //停车单价

“consumption”: ”xxxx” //收费

“state” : “xxxx” //订单状态(false:未完成.true:已完成)

}…

]

}

10.按照车牌号&时间查询

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式：

{

”cmd”:”adminCarPlateRecord”,

“username” : “xxxx”,

“carPlate” : ”0088”, //车牌号

“startTime” : ”20141028” //开始时间

“endTime” : ”20141029” //结束时间

“paginationNum” : xx , //页码

“paginationPage” : xx //每页的容量

}

返回数据格式：{

”cmd”:”adminCarPlateRecord”

“res” : “true”

”recordList”:[

{

//没有这个值,下面也会没有

“id” : xxxx, //订单号

“carPlate” : ”粤B88888”, //车牌号

“voiceCard” : ”0088” //语音卡

”name” : ”xxxx”, //停车场名

”citycode” : ”xxxx”, //城市编码

”address” : ”xxxx”, //停车场地址

“reserveTime” : “xxxx”, //预定时间

“startTime”:”xxxx”, //入场时间

“endTime”:”xxxx”, //出场时间

“reserveHour” : xxxx , //预定时长(不满一小时按一小时算)

“parkHour” : xxxx , //停车时长(不满一小时按一小时算)

“reservePrice” : “xxxx”, //预定单价

“parkPrice”: ”xxxx”, //停车单价

“consumption”: ”xxxx” //收费

“state” : “xxxx” //订单状态(false:未完成.true:已完成)

}…

]

}

11.获取普通用户信息(查看)

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式:

{

“cmd” : “getAllUser”,

“username” : “xxxx”,

“paginationNum” : xx , //页码 int

“paginationPage” : xx //每页的容量 int

}

返回数据格式:

{

“cmd” : “getAllUser”,

“res” : “true”

“userList” : [

{

“username” : “xxxx”,

“phone” : “xxxx”,

“realname” : “xxxx”, //真实姓名

“IDcard” : “xxxx”, //身份证

“bankcard” : “xxxx”, //银行卡

“registerTime” : “xxxx” //注册时间

“parkTime” : “xxxx” //停车总时长

“money” : “xxxx” //消费总金额

“bindList” : [

{

“carPlate” : “xxxx”, //车牌号

“voiceCard” : “xxxx” //语音卡

}

…

]

}

…

]

}

12.获取未受理工单(没有管理员去处理)

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式:

{

“cmd” : “adminGetNoAccept”,

“username” : “xxxx”

}

返回数据格式:

{

“cmd” : “adminGetNoAccept”,

“res” : “true”;

“workOrderList” : [

{

“id” : “xxxx”, //工单编号

“startTime” : “xxxx”, //工单开始时间

“state” : “xxxx”, //工单状态(1:提交)

“messageList” : [

{

“name” : “xxxx”, //消息发送者

//type是为了前台给上色方便

“type” : “xxxx”, //用户类型(user&admin两种)

//时间有排序作用

“time” : “xxxx” //发送时间

“message” : “xxxx” //发送内容

},

{

“name” : “xxxx”,

“type” : “xxxx”,

“time” : “xxxx”

“message” : “xxxx”

},

…

]

}

…

]

}

13.获取正在处理工单(这种工单别的管理员看不见)

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式:

{

“cmd” : “adminGetAccept”,

“username” : “xxxx”,

}

返回数据格式:

{

“cmd” : “adminGetAccept”,

“res” : “true”;

“workOrderList” : [

{

“id” : “xxxx”, //工单编号

“startTime” : “xxxx”, //工单开始时间

“state” : “xxxx”, //工单状态(2:处理中)

“messageList” : [

{

“name” : “xxxx”, //消息发送者

//type是为了前台给上色方便

“type” : “xxxx”, //用户类型(user&admin两种)

//时间有排序作用

“time” : “xxxx” //发送时间

“message” : “xxxx” //发送内容

},

{

“name” : “xxxx”,

“type” : “xxxx”,

“time” : “xxxx”

“message” : “xxxx”

},

…

]

}

…

]

}

14.工单提交

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式:

{

“cmd” : “adminPutWorkOrder”,

“username” : “xxxx”,

“id” : xxxx, //工单编号

“message” : “xxxx” //提交的内容

}

返回数据格式:

{

“cmd” : “adminPutWorkOrder”,

“res” : “true”

}

15.查询历史工单(查询的订单的状态都是完成状态)

web的url : /Park/rest

Flex,android : http://IP/Park/clientRest

请求数据格式:

{

“cmd” : “adminQueryHisWO”, //这个太长了缩写一下

“username” : “xxxx”,

//以工单开始时间作为标准

“startTime” : ”20141028”, //开始时间

“endTime” : ”20141029” , //结束时间

“paginationNum” : xx, //页码

“paginationPage” : xx //每页的容量

}

返回数据格式:

{

“cmd” : “adminQueryHisWorkOrder”,

“res” : “true”,

workOrderList : [

{

“id” : “xxxx”, //工单编号

“question” : “xxxx”, //提问者

“answer” : “xxxx”, //回答者

“assess” : “xxxx”, //评价(已评价才会有评价)

“startTime” : “xxxx”, //工单开始时间

“endTime” : “xxxx”, //工单结束时间

“state” : “xxxx”, //工单状态(3:完成,4:已评价)

“messageList” : [

{

“name” : “xxxx”, //消息发送者

//type是为了前台给上色方便

“type” : “xxxx”, //用户类型(user&admin两种)

//时间有排序作用

“time” : “xxxx” //发送时间

“message” : “xxxx” //发送内容

},

{

“name” : “xxxx”,

“type” : “xxxx”,

“time” : “xxxx”

“message” : “xxxx”

},

…

]

}

…

}

[停车场管理员接口] (最简系统)(通过车牌号确定唯一性)

1.登录(返回停车场数据)

web的url : /Park/login

请求数据格式:

{

“cmd” : “parkAdminLogin”,

“username” : “xxxx”,

“password” : “abc”

}

返回数据格式:

{

“cmd” : “login”,

“parkId” : xxxx, //上送停车场数据要用到的(我在cookie中存的string类型)

“res” : “true”

}

2.上送停车场数据(由web端来完成,也不需要检测仪器)

//在登录完成后,使用这个接口将已停和可预定上传,预定没有

web的url : /Park/rest

请求数据格式:

{

“cmd” : “parkData”,

“username” : “xxxx”,

“parkId” : xxxx, //long

“total” : xxxx, //总共车位

“maxReserve” : xxxx, //最大预定数车位

}

返回数据格式:

{

“cmd” : “parkData”,

“res” : “true”

}

3.开始停车(没预定车辆开始停车)

web的url : /Park/rest

请求数据格式:

{

“cmd” : “startPark”,

“username” : “xxxx”, //停车场管理员的用户名

“parkId” : xxxx

“reserveName” : “xxxx”, //停车者的用户名

“carPlate” : “xxxx”, //车牌号

“voiceCard” : “xxxx” //语音卡

}

返回数据格式:

{

“cmd” : “startPark”,

“res” : “true”,

}

4.预定请求(websocket)

服务器推送数据:

{

“cmd” : “reserveRequest”,

“reserveName” : “xxxx”, //请求的人的用户名

“parkId” : xxxx,

“carPlate” : “xxxx”, //车牌号

“voiceCard” : “xxxx” //语音卡

}

5.解除预定请求(websocket)

服务器推送数据:

{

“cmd” : “removeReserve”,

“reserveName” : “xxxx”, //请求的人的用户名

“parkId” : xxxx, //Long

“carPlate” : “xxxx”, //车牌号

“voiceCard” : “xxxx” //语音卡

}

6.预定—>已停(预定车辆开始停车)

web的url : /Park/rest

请求数据格式:

{

“cmd” : “reserveStartPark”,

“username” : “xxxx”,

“parkId” : xxxx,

“carPlate” : “xxxx”, //车牌号

“voiceCard” : “xxxx” //语音卡

}

返回数据格式:

{

“cmd” : “reserveStartPark”,

“res” : “true”

}

7.停车结束

web的url : /Park/rest

请求数据格式:

{

“cmd” : “parkFinish”,

“username” : “xxxx”,

“parkId” : xxxx,

“carPlate” : “xxxx”, //车牌号

“voiceCard” : “xxxx” //语音卡

}

返回数据格式:

{

“cmd” : “parkFinish”,

“res” : “true”

}

附录1:我的测试数据

经度 , 纬度 总共 可预定

112.183692,30.341042  公院 100 40

112.177691,30.33858 医专 50 20

112.22178,30.338861 长大 50 20

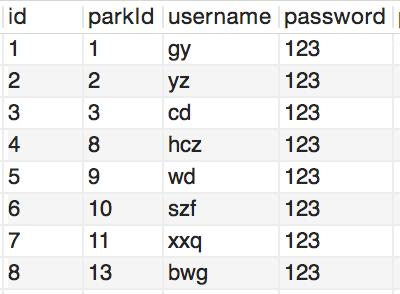
112.219624,30.374511 火车站 50 20

112.231711,30.343536 万达 100 50

112.245855,30.341292 市政府 50 20

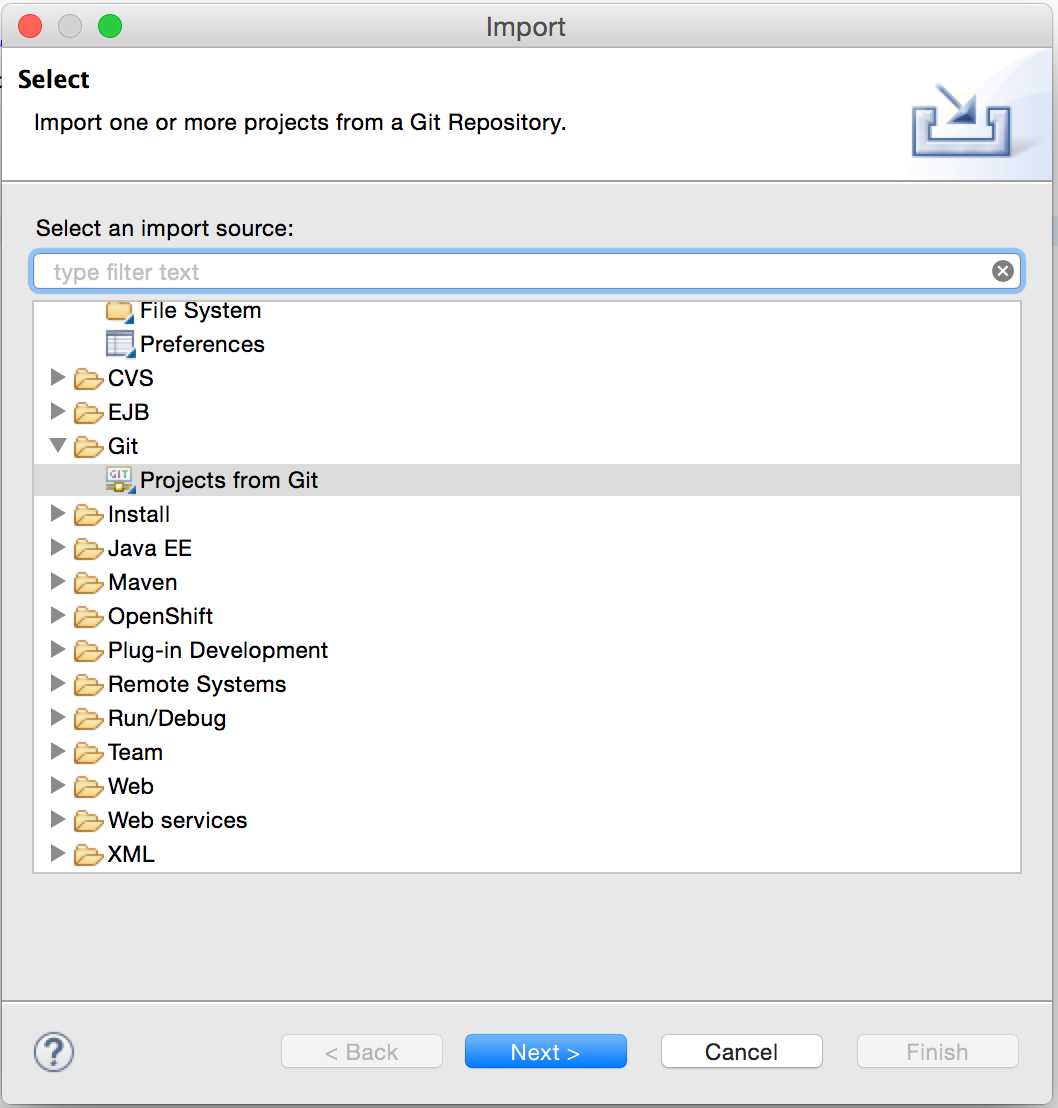
112.156096,30.360801 西校区 50 20

112.187429,30.358807 博物馆 50 20

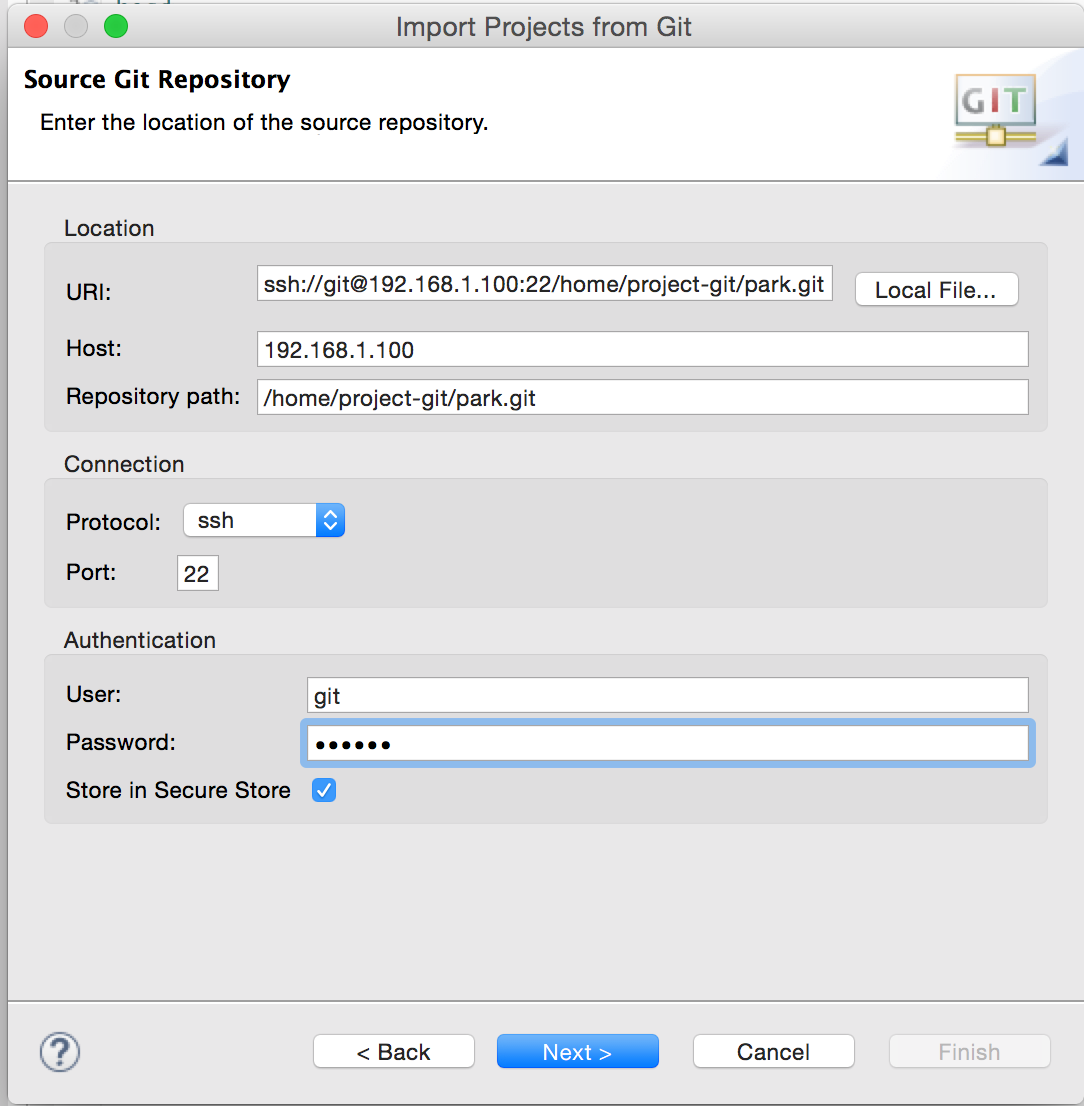
停车场管理员

附录2: 项目创建

git地址: ssh://git@192.168.1.110:22/home/project-git/park.git (用户名git,密码123456)

在项目文件夹空白处 import

在其中选择git项目,next,选择url.



将ssh://git@192.168.1.100:22/home/project-git/park.git复制到url中,在passsword填写123456,勾上记住密码.next

选择项目路径,其他默认,next,选择项目名字(Park),finish

在park项目右键,选择configure,选择转maven.