SJTU BUS

测试用例

版本 <1.0>

、

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <日/月/年> | <x.x> | <详细信息> | <姓名> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

测试用例

# 安卓客户端集成测试

## 用户登录测试

前置条件：用户未登录

测试输入：输入手机号码、密码

观察点：点击“登录”

控制点：手机号码是否已注册；手机号码与密码是否匹配

预期结果：若手机号未注册，提示“该账号不存在哦~”

若手机号码与密码不匹配，提示“手机号码或密码填写不正确哦~”

若正确，提示“登陆成功~”

## 管理员/司机登录测试

前置条件：管理员未登录

测试输入：输入用户名和密码

观察点：点击“登录”

预期结果：如果用户名和密码不匹配，则提示“不存在该用户哦”

若正确，提示登陆成功

## 注册测试

前置条件：用户未登录

测试输入：输入错误的信息

观察点：点击“注册”

预期结果：若用户名未填写，提示“用户名不能为空哦~”

若手机号码未填写，提示“手机号码不能为空哦~”

若手机号码不合规范，提示“请输入正确的手机号码哦~”

若密码为空，提示“密码不能为空哦~”

若密码非6-16位，提示“请设置6-16位的密码哦~”

若两次密码输入不一致，提示“两次密码输入不一致哦~”

若验证码为空，提示“验证码不能为空哦~”

若未勾选协议，提示“请确保您已阅读并同意《SJTUBUS用户协议》哦~”

若验证码输入错误，提示“验证失败”

## 忘记密码测试

前置条件：用户进入登录界面

观察点：用户点击忘记密码

## Jaccount登录测试

前置条件：用户未登录

测试输入：输入jaccount用户名、密码以及验证码

观察点：点击“登录”

控制点：输入信息是否正确

预期结果：若正确，提示“登陆成功”

否则，显示错误提示

## 账号切换测试

前置条件：用户已登录

测试输入：用户退出登录，并由该设备登录其他身份的账号（司机/管理员/用户/jaccount用户）

预期结果：不同的用户角色看到不同的主界面

## 预约单程班车测试1

前置条件：用户已登录，若未登录提示“请先登录~”

测试输入：调整单程的起点，终点

观察点：点击“查询”

预期结果：若起点终点相同，提示“起点和终点不能相同哦~”

若起点终点为徐汇和七宝的连线，提示“没有徐汇到七宝的班车哦~”

若试图调时间到今天之前，提示“不能预约已经发出的班次哦~”

若正常，进入下一页面

## 预约单程班车测试2

前置条件：用户选择预约的线路和日期

测试输入：点击前一天，后一天，或直接调整日期

预期结果：若调到今天之前，提示“不能预约已经发出的班次哦~”

若当日班次是0，加一句“快去预约其他的班次吧~”

若点击班次信息，显示班次详细信息

若点击收藏，提示“收藏功能还不能使用哦~”

若点击预约，进入订单填写页面

## 预约单程班车测试3

前置条件：用户选择了班次

测试输入：用户调整预约提醒和特殊需求的设置

观察点：点击“预约”

预期结果：如果有座位，显示预约成功的提示框，再次进入界面查看剩余座位数减少

如果没座位或者预约失败，提示预约失败

## 预约往返班车测试1

前置条件：用户已登录，若未登录提示“请先登录~”

测试输入：调整往返的起点，终点，出发时间，返程时间

观察点：点击“查询”

预期结果：同预约单程班车测试1

若调整时间后回程早于去程，提示“回程的时间不能早于去程哦~”

## 预约单程班车测试2

前置条件：用户进入设置去程/返程界面

测试输入：调整去程的时间，选择班次

观察点：点击“预约”

预期结果：同预约单程班车测试2

若调整时间后回程早于去程，提示“回程的时间不能早于去程哦~”

若调整时间后回程和去程时间又重叠，提示“不能预约行程冲突的班次哦~”

## 班次信息测试1

前置条件：用户进入班次信息界面

测试输入：用户选择类型，点击具体的班次

观察点：点击“预约”

预期结果：默认的显示就是当天正确的类型

显示当天正常运行的班次及当天的首班车、末班车

## 班次信息测试2

前置条件：用户进入时刻表/路线站点界面

测试输入：如果是校内巴士，可选择站点查看不同站点的信息

预期结果：显示当前类型及线路条件下的出发时间，备注及站点信息

## 路线查询测试

前置条件：用户点击进入路线查询界面

观察点：用户点击路线切换并且切换线路

预期结果：界面显示为起点和终点的最优路线，路线切换后线路图随之改变

## 实时位置测试1

前置条件：用户点击实时位置进入界面

测试输入：用户对地图进行移动、放缩，点击图标

预期结果：若用户对地图进行缩放，达到一定缩放比时会显示站点中文名

用户什么都不做，巴士每隔3-5秒刷新一次当前位置

若司机打开了司机定位，则在实时位置上可以看到司机的位置及其移动

## 实时位置测试2

前置条件：用户点击实时位置进入界面

观察点：点击站点标记，点击巴士

预期结果：点击站点标记显示站点相关的所有时刻信息，上拉可以全屏显示

## 实时位置测试3

前置条件：用户点击实时位置进入界面

测试输入：用户点击右上角图标，选择任意站点

预期结果：弹出与用户点击的站点对应的时刻信息，上拉可以全屏显示

## 实时位置测试4

前置条件：用户点击实时位置进入界面

测试输入：用户点击右上角图标，并选择最近的站点

预期结果：系统自动判断出离当前用户最近的站点，弹出该站点对应的时刻信息，上拉可以全屏显示

## 管理员扫描预约码测试1

前置条件：管理员已登录，正在扫描用户预约记录中的预约码

观察点：管理员扫描预约码

预期结果：若预约码正确，则提示“验证成功”

若扫描的不是预约码，则提示“验证错误”

## 管理员扫描预约码测试2

前置条件：管理员扫描用户的预约码

预期结果：当用户的预约码是正确的，并且该车还未发出，用户未上车时，扫描后用户的预约记录状态出现变化，由预约未上车变为预约已上车

## 司机定位测试

前置条件：已登录并且身份是司机

测试输入：点击开启定位

预期结果：显示定位已开启，用户手机在实时位置上可以get到司机

司机在屏幕熄灭的情况下应该也能继续，但是退出就不能继续被定位了

## 个人信息测试1

前置条件：用户已登录

测试输入：用户点击“退出登录”

预期结果：用户退出登录，侧拉栏显示已经退出登录

## 个人信息测试2

前置条件：用户已登录

测试输入：当用户手机号/学号/真实姓名为空时，用户点击进行编辑

预期结果：用户编辑个人信息确定后，退出重新进入界面个人信息已经更新

## 预约记录测试

前置条件：用户已登录且进入预约记录界面

测试输入：用户点击预约记录的操作，排序

预期结果：用户可以对预约记录进行排序和筛选

## 取消预约测试

前置条件：用户在预约记录界面

测试输入：用户点击取消预约

预期结果：预约记录删除，预约座位处的座位数回升一位

## 预约提醒测试

前置条件：用户在预约记录界面

测试输入：用户点击预约提醒

预期结果：弹出对话框让用户确认预约提醒，并可以选择提醒的时间，并到时推送提醒

## 班次收藏测试

前置条件：用户在预约班车界面

测试输入：用户点击收藏对具体班次进行收藏

预期结果：对话框提示收藏成功，在侧拉栏目的班次收藏界面可以看到自己收藏的班次；

如果是已经收藏过的班次，则提示班次收藏失败

## 取消班次收藏测试

前置条件：用户在班次收藏界面

测试输入：用户点击取消班次收藏

预期结果：班次收藏删除，计数器的值减一

## 快速预约测试

前置条件：用户在班次收藏界面

测试输入：用户点击快速预约并选择日期

预期结果：如果用户选择日期的班次已发出，提示“不能预约已经发出的班次哦~”

如果用户选择的日期在一周以后，提示“仅可预约一周以内的班次哦~”

如果用户选择的日期已经有其他行程，提示“不可以预约冲突的行程哦~”

正常情况下，提示预约成功，并在预约记录处添加

## 管理员/司机录入乘车信息测试

前置条件：管理员或者司机已经登录

测试输入：管理员点击录入乘车信息

预期结果：如果管理员选择了还未发出的班次，则提示“该班次还未发出”

正常情况下，管理员填写乘车信息的各项数据，提交后后端的数据库中显示出了数据，并对与没有上车的人进行信用积分的扣分。

## 公告测试

前置条件：用户在我的消息界面

测试输入：后台发送一条公告

预期结果：用户在公告界面可以收到来自后台的公告

## 意见测试

前置条件：用户在用户反馈界面

测试输入：用户输入文字或者发送图片

预期结果：在界面上保留着用户发出的文字和图片，并且退出重新进入后仍然存在

## 设置测试

前置条件：用户在设置界面

预期结果：可以设置：是否在order界面提示预约提醒；账号切换；关于与帮助；主界面显示的功能

## 断网测试

前置条件：网络已经被切断

预期结果：点进各个活动时APP不会崩溃，而会提示“网络请求失败，请检查你的网络”，并在主界面提示网络变化。

# 后端单元测试

## StringCalendarUtils测试用例1

测试StringCalendarUtils类的stringToDate方法

1. 测试输入

i)“2018-07-25”

1. 预期结果

i)值为2018-07-25的date类

## StringCalendarUtils测试用例2

测试StringCalendarUtils类的stringToTime方法

1. 测试输入

i) “2018-07-25 23:56:00”

1. 预期结果
2. 相值为2018-07-25 23:56:00应的date类

## StringCalendarUtils测试用例3

测试StringCalendarUtils类的dateToString方法

1) 测试输入

i) 值为2018-07-26的date类

1. 预期结果

i)“2018-07-26”

## StringCalendarUtils测试用例4

测试StringCalendarUtils类的timeToString方法

1)测试输入

i)值为00:00:05的date类

2)预期结果

i)“00:00:05”

## StringCalendarUtils测试用例5

测试StringCalendarUtils类的isBeforeCurrentTime方法

1)测试输入

i)”2018-07-25 23:56:00”

ii)”2018-07-26 00:00:02”

iii)”2019-01-01 12:00:00”

2)预期结果

i)true

ii)true

iii)false

## StringCalendarUtils测试用例6

测试StringCalendarUtils类的getCurrentDate方法

1)测试输入

i)无

2)预期结果

i)当前日期 eg：“2018-07-26”

## StringCalendarUtils测试用例7

测试StringCalendarUtils类的getCurrentTime方法

1)测试输入

i)无

2)预期结果

i)当前时间 eg “2018-07-26 00:00:05”

## ShiftUtils测试用例1

测试ShiftUtils类的getEngType方法

1)测试输入

i)“在校期-工作日”

2)预期结果

i)“NormalWorkday”

## ShiftUtils测试用例2

测试ShiftUtils类的getChiType方法

1)测试输入

i) “NormalWorkday”

2)预期结果

i) “在校期-工作日”

## ShiftUtils测试用例3

测试ShiftUtils类的getEngLineName方法

1)测试输入

i) "校园巴士逆时针"

2)预期结果

i) "LoopLineAntiClockwise"

## ShiftUtils测试用例4

测试ShiftUtils类的getChiLineName方法

1)测试输入

i) "LoopLineAntiClockwise"

2)预期结果

i) "校园巴士逆时针"

## AppointmentService测试用例1

测试AppointmentService类的getAppointInfo方法(预约班车界面)

1)测试输入

i) line\_name = "MinHangToQiBao"; type = "HolidayWorkday"; appoint\_date = "2018-07-27";

2)预期结果

i) 相应的AppointInfo类, size = 2, get(1).shiftid = MQHD1700

## AppointmentService测试用例2

测试AppointmentService类的getAddAppointment方法

1)测试输入

i) lineNameCN = "闵行到七宝"; line\_type = "HolidayWorkday";

time = “17:00:00”; date = “2018-07-27”

2)预期结果

i) 相应的Appointments，size = 1

## AppointmentService测试用例3

测试AppointmentService类的verifyAppointment方法

1)测试输入

i) username = "姚子航"; departure\_date = "2018-07-27"; shift\_id = "MQHD1700";

2)预期结果

i) 验证前 isNormal = false, 验证后 isNormal = true

## LineService测试用例1

测试LineService类的getStationByLineName方法

1)测试输入

i) line\_name = "MinHangToQiBao";

2)预期结果

i) 相应的stations，get(0)是闵行，get(1)是七宝

## LineService测试用例2

测试LineService类的getLineInfo方法

1)测试输入

i) line\_name = "MinHangToQiBao"; type = "HolidayWorkday";

2)预期结果

i) 相应的lineinfo类，首班车8:50，末班车17:00

## LineService测试用例3

测试LineService类的getAllLineName方法

1)测试输入

i) type = "HolidayWorkday";

2)预期结果

i) 相应的list，size = 6

## ShiftService测试用例1

测试ShiftService类的getScedule方法

1)测试输入

i) String line\_name = "MinHangToQiBao";

String type = "HolidayWorkday";

2)预期结果

i) 相应的schedule，第一班车8:50

## ShiftService测试用例2

测试ShiftService类的getShiftInfo方法

1)测试输入

i) String shift\_id = "MQHD1700";

2)预期结果

i) 相应的shiftinfo, 出发时间17:00， 荷载50

## ShiftService测试用例3

测试ShiftService类的changeTypeToId方法

1)测试输入

i) "NormalWorkday"

ii) "NormalWeekendAndLegalHoliday"

iii) "HolidayWorkday"

2)预期结果

i) 相应的shiftinfo, 出发时间17:00， 荷载50

i) WD

ii) WE

iii) HD

## ShiftService测试用例4

测试ShiftService类的ChangeTimeToStringTime方法

1)测试输入

i) 值为17:00的time类

2)预期结果

i) “1700”

## ShiftService测试用例5

测试ShiftService类的ModifySeat方法

1)测试输入

i) modifySeat(shift\_id,100);

2)预期结果

i)返回值1，成功

## ShiftService测试用例6

测试ShiftService类的GetTimeList方法

1)测试输入

i) String lineNameCn = "闵行到徐汇";

String type = "HolidayWorkday";

2)预期结果

i)相应的times，size = 2

# 后端集成测试

## …/appointment/search

调用者：后台管理系统调用

调用类型：get

调用操作：管理员在后台点击查询预约

参数输入：线路名，线路时间，发车时间，预约日期

结果返回：返回所有符合结果的预约记录

## …/appointment/infos

调用者：安卓前台调用

调用类型：get

调用操作：用户点击预约班车并选择了指定日期

参数输入：线路名，线路时间，预约日期

结果返回：返回当日所有未发出的可预约班次

## …/appointment/record

调用者：安卓前台调用

调用类型：get

调用操作：用户点击预约记录时获取后端数据

参数输入：用户名，用户角色

结果返回：返回该用户所有的预约记录，包括已发车的和未发车的

## …/appointment/appoint

调用者：安卓前台调用

调用类型：post

调用操作：用户在订单填写页面点击预约并确定

参数输入：用户名，用户角色，预约日期，班次id，线路名，提交时间，备注

结果返回：提示用户是否预约成功，若成功则在数据库中和用户的预约记录中显示

## …/appointment/verify

调用者：安卓前台调用

调用类型：post

调用操作：管理员扫描用户的预约码

参数输入：用户名，预约日期，班次id

结果返回：如果扫描成功则提示成功，并修改用户预约记录的状态

## …/appointment/cancel

调用者：安卓前台调用

调用类型：post

调用操作：用户在预约记录界面选择取消预约

参数输入：用户名，用户角色，预约日期，班次id

结果返回：提示用户是否取消成功，如果发出就不能取消，成功就在数据库中删除数据

## …/collection/addCollect

调用者：安卓前台调用

调用类型：post

调用操作：用户在班车显示界面选择收藏班次

参数输入：用户id，用户名，班次id

结果返回：提示用户是否添加收藏成功，如果已经收藏过了就提示失败

## …/collection/infos

调用者：安卓前台调用

调用类型：get

调用操作：用户点击班次收藏界面

参数输入：用户名

结果返回：获取该用户收藏的所有班次

## …/collection/deleteCollect

调用者：安卓前台调用

调用类型：post

调用操作：用户在班次收藏界面选择取消收藏

参数输入：用户名，用户id，班次id

结果返回：删除用户选择取消收藏的班次

## …/line/infos

调用者：安卓前台调用

调用类型：get

调用操作：用户点击班次信息

参数输入：线路类型

结果返回：显示当前线路类型下运行的所有班次

## …/line/stations

调用者：安卓前台调用

调用类型：get

调用操作：用户点击具体线路查看信息

参数输入：线路名

结果返回：显示该线路的所有时刻及站点信息

## …/location/locate

调用者：安卓前台调用

调用类型：post

调用操作：司机在司机定位界面位置移动

参数输入：经度，纬度

结果返回：更新司机位置的经纬度信息

## …/location/location

调用者：安卓前台调用

调用类型：get

调用操作：用户在实时位置界面实时获取司机的位置

结果返回：显示司机的位置

## …/message/add

调用者：后台管理系统调用

调用类型：post

参数输入：信息类型，信息标题，信息内容，开始时间，结束时间

结果返回：添加消息/公告

## …/message/all

调用者：安卓前台调用

调用类型：get

调用操作：用户点击公告按钮

结果返回：获取到当前已经有的所有公告信息

## …/ridebusinfo/import

调用者：安卓前台调用

调用类型：post

调用操作：管理员/司机填写某班次的乘车信息

结果返回：填写该次乘车信息并对预约未到的用户进行扣分

## …/shift/search\_schedule

调用者：安卓前台调用

调用类型：get

结果返回：获取当前类型和线路的时刻表

## …/shift/infos

调用者：安卓前台调用

调用类型：get

结果返回：获取班次信息详情

## …/shift/search\_schedule\_loopline

调用者：安卓前台调用

调用类型：get

结果返回：获取校园环线的时刻表

## …/shift/add

调用者：后台管理系统调用

调用类型：post

结果返回：添加班次

## …/shift/search\_shift

调用者：后台管理系统调用

调用类型：get

结果返回：查询班次

## …/shift/delete

调用者：后台管理系统调用

调用类型：post

结果返回：删除班次

## …/shift/search\_time

调用者：后台管理系统调用

调用类型：post

结果返回：查询班次的时刻表

## …/shift/modify

调用者：后台管理系统调用

调用类型：post

结果返回：修改班次

## …/oauth/jaccount

调用者：安卓前台调用

调用类型：post

结果返回：判断用户在前台输入的jaccount账号密码是否合法且匹配，并反馈

## …/account/login

调用者：安卓前台调用

调用类型：post

结果返回：用户前台登录

## …/account/admin

调用者：安卓前台调用

调用类型：post

结果返回：管理员前台登录

## …/account/register

调用者：安卓前台调用

调用类型：post

结果返回：用户前台注册

## …/account/logout

调用者：安卓前台调用

调用类型：post

结果返回：用户前台登出

## …/account/profile

## …/account/update\_infos

调用者：安卓前台调用

调用类型：post

结果返回：用户前台更新个人数据

## …/administrator/login

调用者：后台管理系统调用

调用类型：post

结果返回：管理员后台登录

## …/administrator/logout

调用者：后台管理系统调用

调用类型：post

结果返回：管理员后台登出

## …/administrator/judgestate

调用者：后台管理系统调用

调用类型：post

结果返回：管理员后台判断状态

## …/administrator/register

调用者：后台管理系统调用

调用类型：post

结果返回：管理员后台注册

## …/driver/search

调用者：后台管理系统调用

调用类型：post

结果返回：后台查找匹配的司机信息

## …/driver/add

调用者：后台管理系统调用

调用类型：post

结果返回：后台添加新司机

## …/driver/delete

调用者：后台管理系统调用

调用类型：post

结果返回：后台删除司机

## …/driver/modify

调用者：后台管理系统调用

调用类型：post

结果返回：后台修改司机的信息

## …/driver/login

调用者：安卓调用

调用类型：post

结果返回：司机前台登录

# 后台管理系统集成测试

## 管理员登录测试

前置条件：输入网站url后自动跳转至登录界面

测试输入：输入管理员账号及密码

控制点：账号是否存在，密码是否与账号匹配

预期结果：若账号存在且密码匹配则能正常使用其他所有功能，否则提示输入信息有误，等待重新输入

## 查找用户测试

前置条件：按照目录提示进入查找用户界面

测试输入：输入用户相关信息

预期结果：将所有信息中含有关键字的用户以列表形式呈现，若无则表格为空

## 查询校内巴士时刻表测试

前置条件：按照目录提示进入查询校内巴士时刻表界面

观察点：点击选择框选择想查询的班次的行驶方向、是否寒暑假、是否节假日

预期结果：根据选择的属性将所有符合条件的班次筛选出并以表格形式展现

## 查询校区巴士班次表测试

前置条件：按照目录提示进入查询校区巴士班次表界面

观察点：点击选择框选择想查询的班次的始发站、终点站、是否寒暑假、是否节假日

预期结果：根据选择的属性将所有符合条件的班次筛选出并以表格形式展现

## 查询校区巴士预约信息测试

前置条件：按照目录提示进入查询校区巴士预约信息界面

观察点：点击选择框选择想查询的班次的出发日期、始发站、终点站、线路类型、发车时间

预期结果：根据选择的属性将所有符合条件的预约信息筛选出并以表格形式展现

## 添加用户测试

前置条件：按照目录提示进入添加用户界面

测试输入：输入新用户的姓名、密码、确认密码、电话号码

观察点：点击选择框选择用户积分和身份，点击添加按钮

预期结果：成功将新用户添加至数据库，并有消息框提示添加成功

## 修改用户测试

前置条件：按照目录提示进入修改用户界面

测试输入：输入用户相关信息

观察点：在相应用户的信息行点击修改

测试输入：修改可修改的属性并保存

预期结果：表格展示修改后的结果，数据库的数据被更新

## 删除用户测试

前置条件：按照目录提示进入删除用户界面

测试输入：输入用户相关信息

观察点：在相应用户的行点击删除

预期结果：表格中没有该用户的信息，数据库的数据成功被删除

## 添加校内巴士班次测试

前置条件：按照目录提示进入添加校内巴士班次界面

观察点：点击选择框选择巴士类型、方向、始发站、终点站、出发时刻、是否节假日、巴士，点击添加按钮

预期结果：成功将新校内巴士班次添加至数据库，并有消息框提示添加成功

## 添加校区巴士班次测试

前置条件：按照目录提示进入添加校区巴士班次界面

观察点：点击选择框选择巴士类型、方向、始发站、终点站、出发时刻、是否节假日、是否工作日、预留座位数、巴士，点击添加按钮

预期结果：成功将新校区巴士班次添加至数据库，并有消息框提示添加成功

## 修改班次测试

前置条件：按照目录提示进入修改巴士班次界面

测试输入：输入班次相关信息

观察点：在相应班次的信息行点击修改

测试输入：修改可修改的属性并保存

预期结果：表格展示修改后的结果，数据库的数据被更新

## 删除班次测试

前置条件：按照目录提示进入删除巴士班次界面

测试输入：输入班次相关信息

观察点：在相应班次的行点击删除

预期结果：表格中没有该班次的信息，数据库的数据成功被删除

## 添加司机测试

前置条件：按照目录提示进入添加司机界面

测试输入：输入新司机的姓名、密码、确认密码、电话号码

观察点：点击选择框选择司机身份，点击添加按钮

预期结果：成功将新司机添加至数据库，并有消息框提示添加成功

## 修改司机测试

前置条件：按照目录提示进入修改司机信息界面

测试输入：输入司机相关信息

观察点：在相应司机的信息行点击修改

测试输入：修改可修改的属性并保存

预期结果：表格展示修改后的结果，数据库的数据被更新

## 删除司机测试

前置条件：按照目录提示进入删除司机界面

测试输入：输入司机相关信息

观察点：在相应用户的行点击删除

预期结果：表格中没有该司机的信息，数据库的数据成功被删除

## 发布新公告测试

前置条件：按照目录提示进入发布新公告界面

观察点：选择公告类型、公布的日期范围

测试输入：输入公告标题、公告内容

观察点：点击发布按钮

预期结果：新公告被存入数据库，用户收到新公告的提醒，消息框提示发布成功

## 统计预约信息测试

前置条件：按照目录提示进入统计预约信息界面

观察点：选择统计方式、统计时间段、始发站、终点站、线路类型、出发时刻，点击生成数据

预期结果：统计分析出数据总量、平均预约座位数、最大预约座位数、最小预约座位数并绘制折线图

## 统计已发车次乘客测试

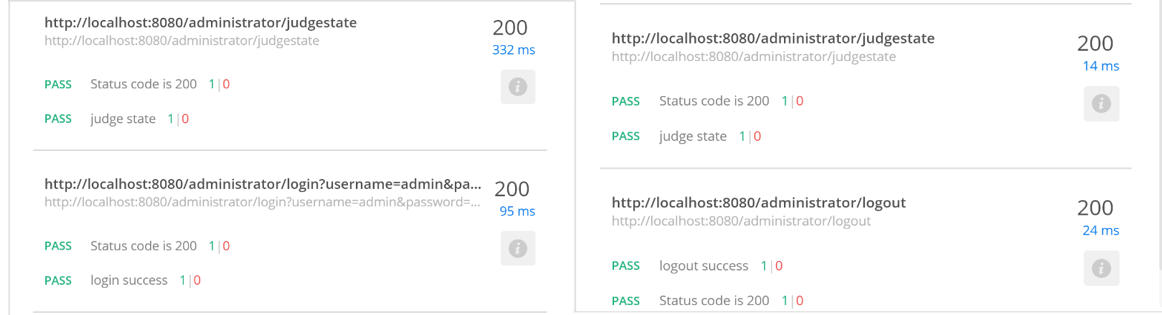
前置条件：按照目录提示进入统计已发车次乘客信息

观察点：选择统计方式、统计时间段、始发站、终点站、线路类型、出发时刻，点击生成数据

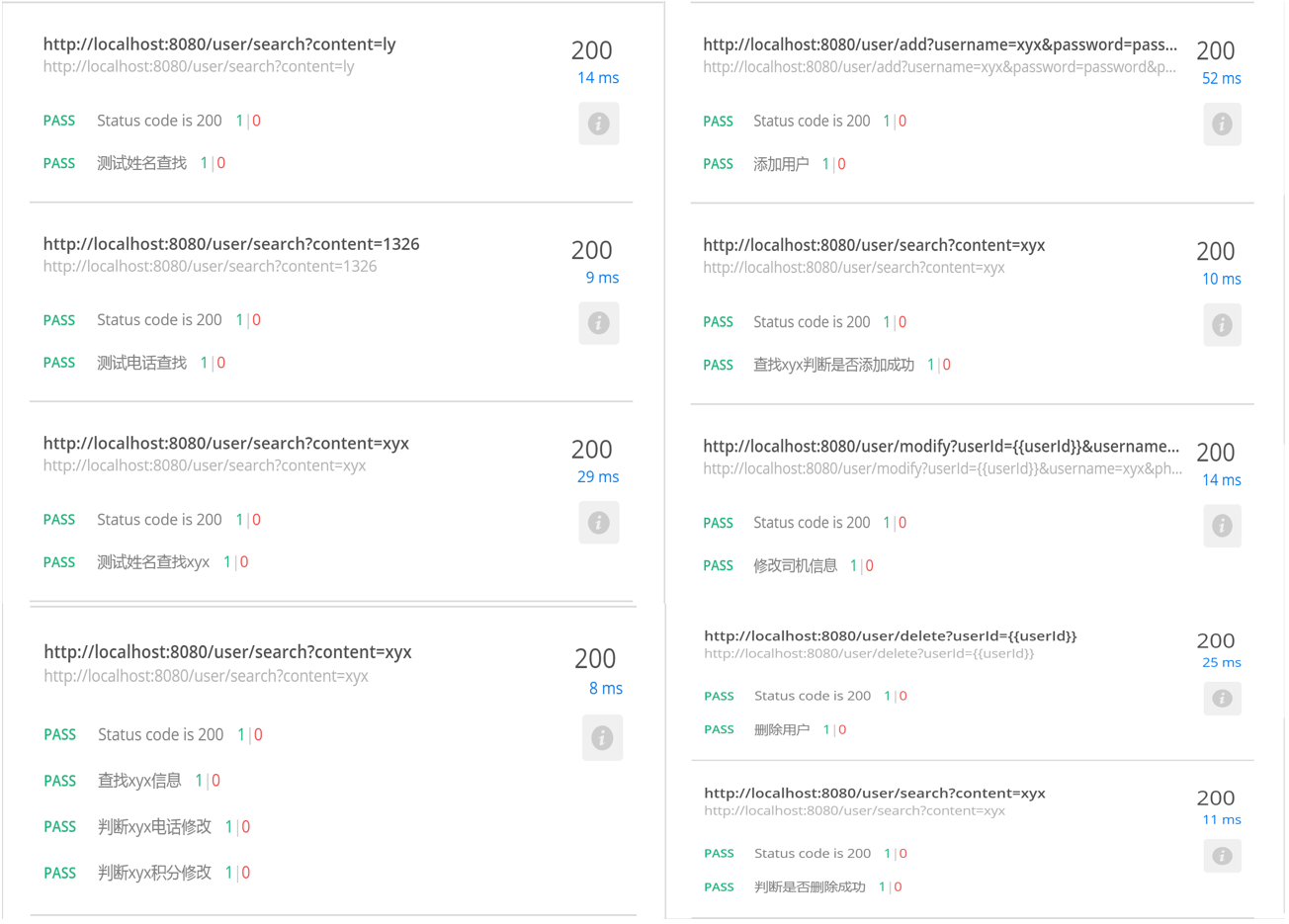
预期结果：统计分析出平均实际上车人数、教工平均乘坐人数、非教工平均乘坐人数及最大最小人数等，并绘制饼图

# 后台管理系统接口测试

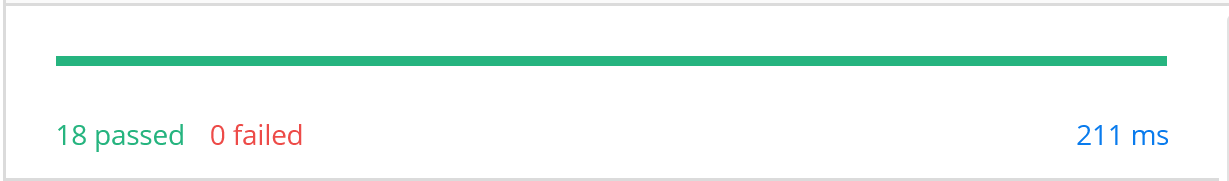
## 和administrator相关的接口测试（测试用例数：4个 测试结果：全部通过）

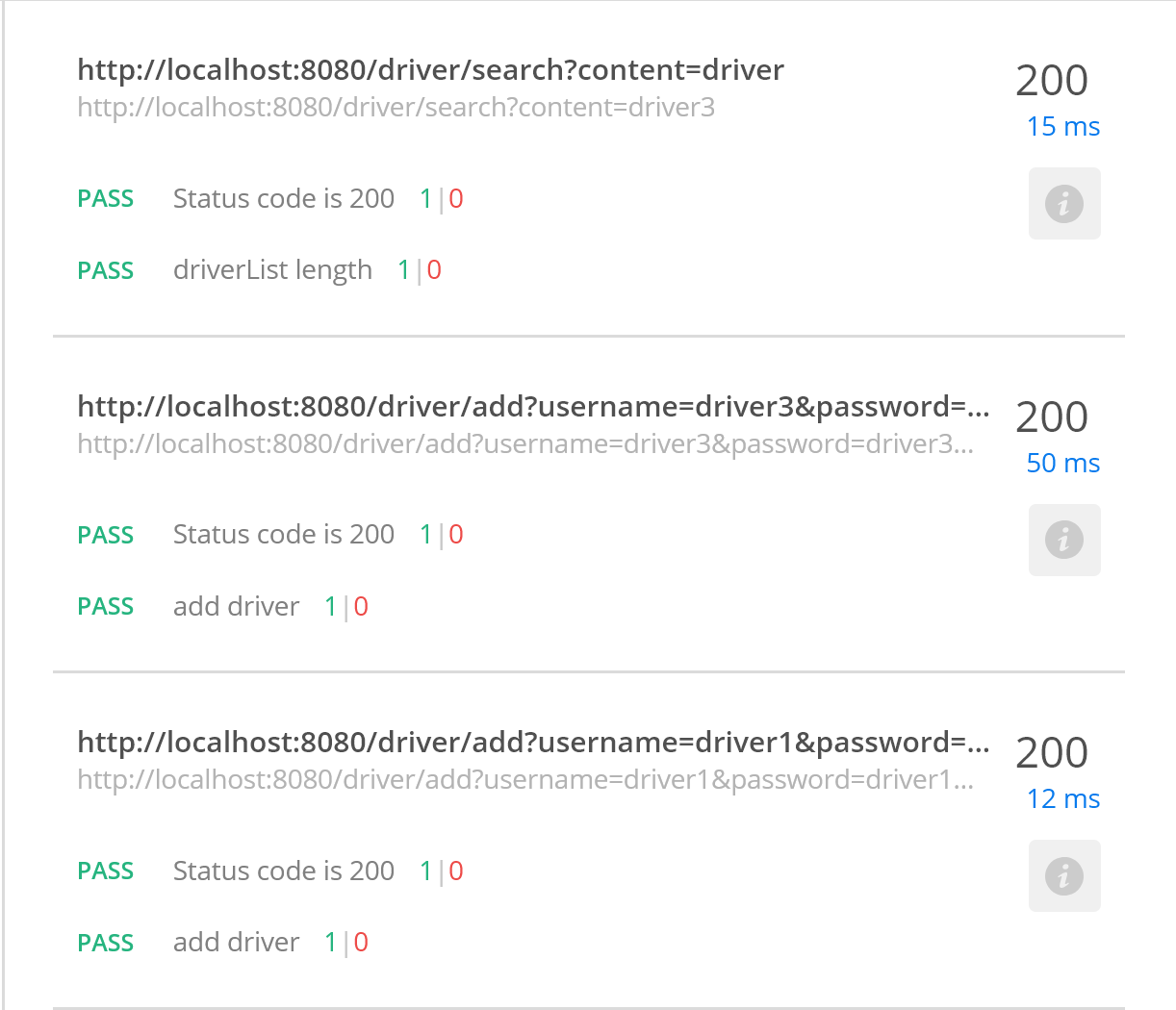
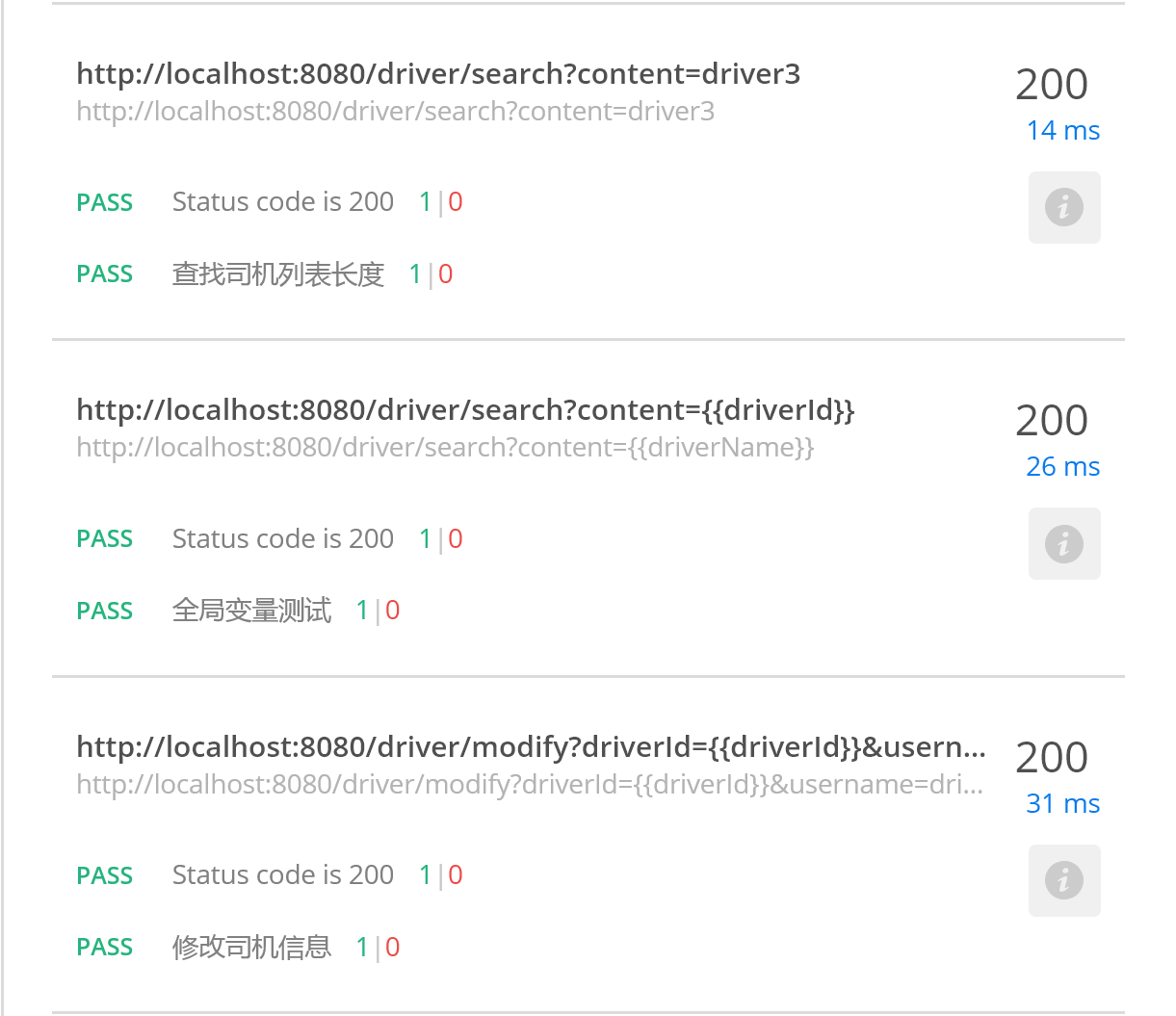


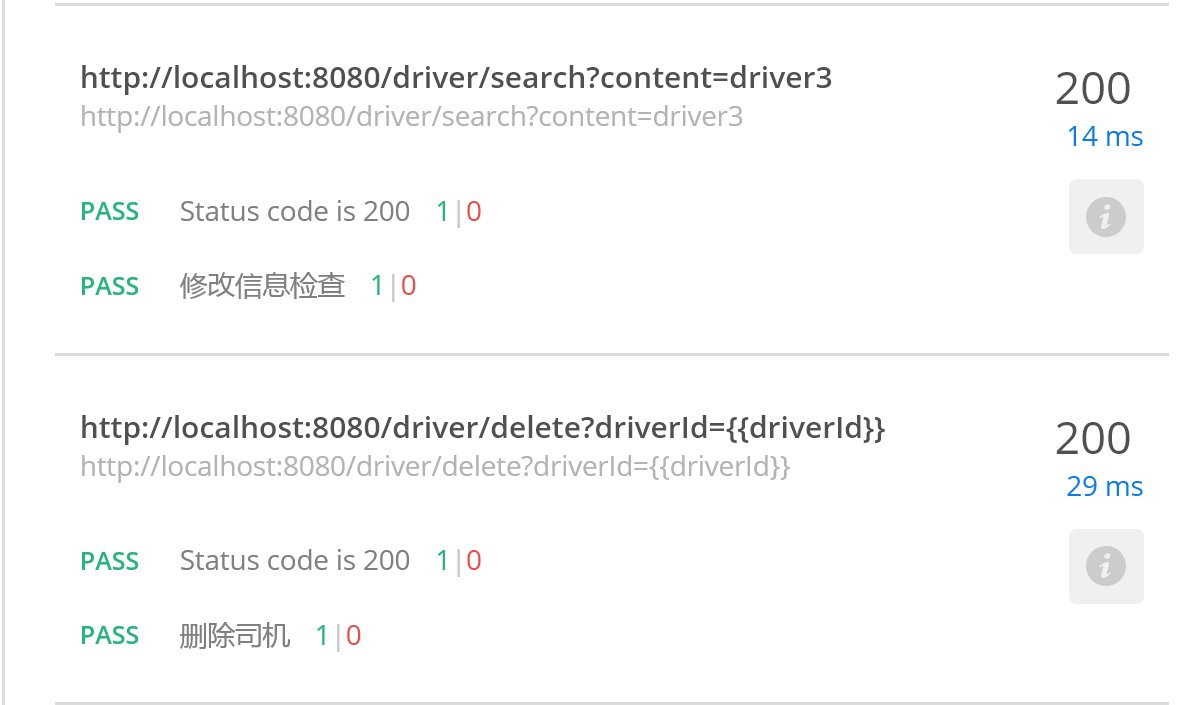
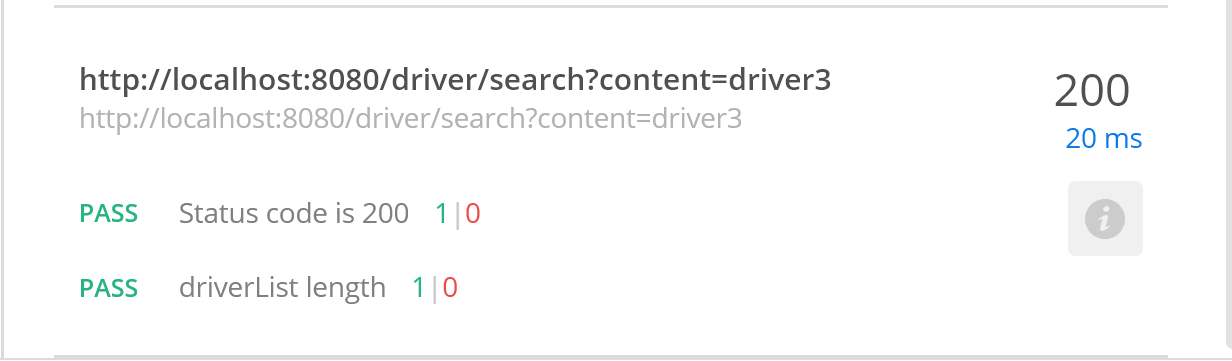
## 和user相关的接口测试（ 测试用例数：9个 测试结果：全部通过）



## 和driver相关的接口测试（测试用例数：9个 测试结果：全部通过）

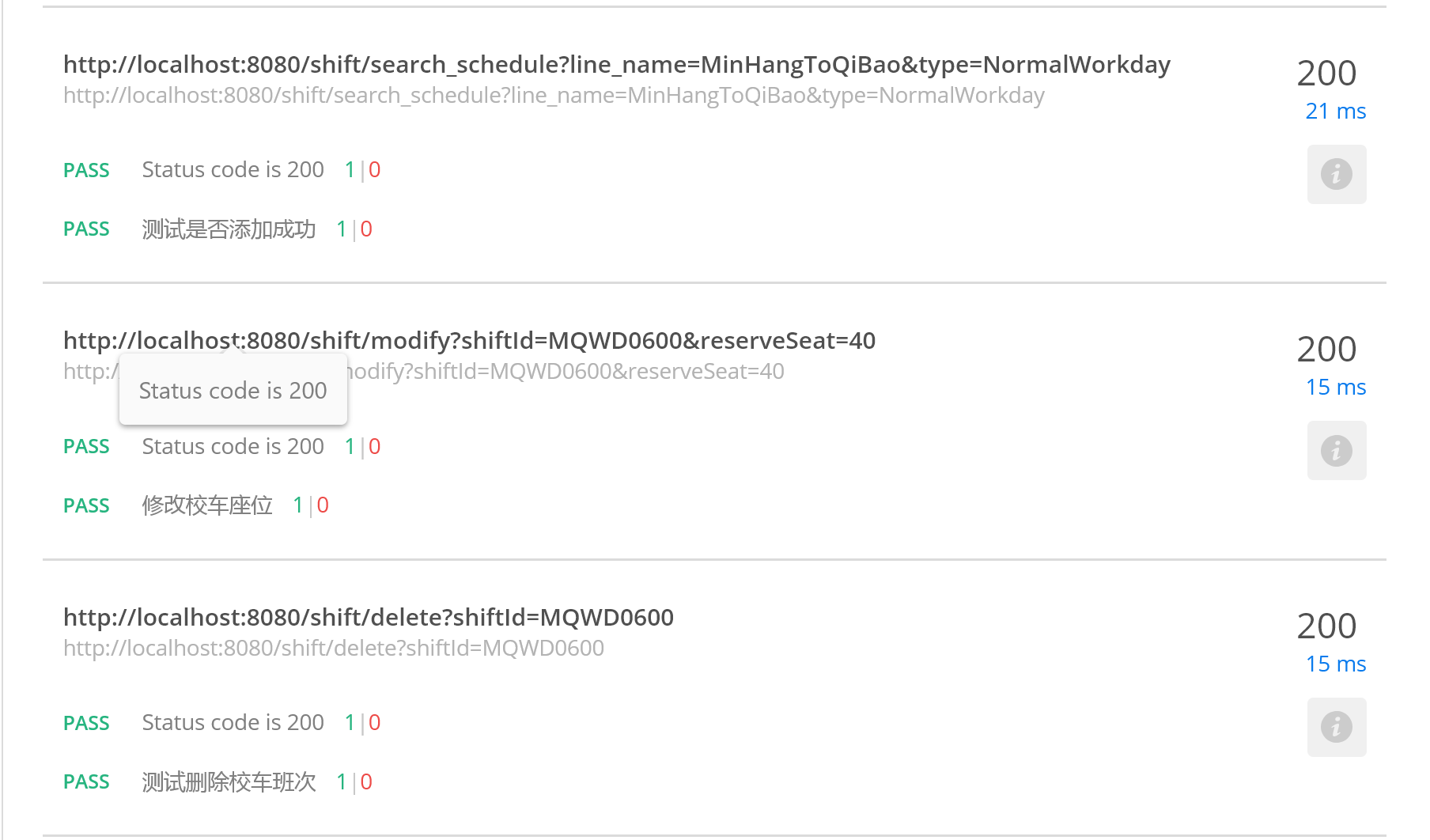
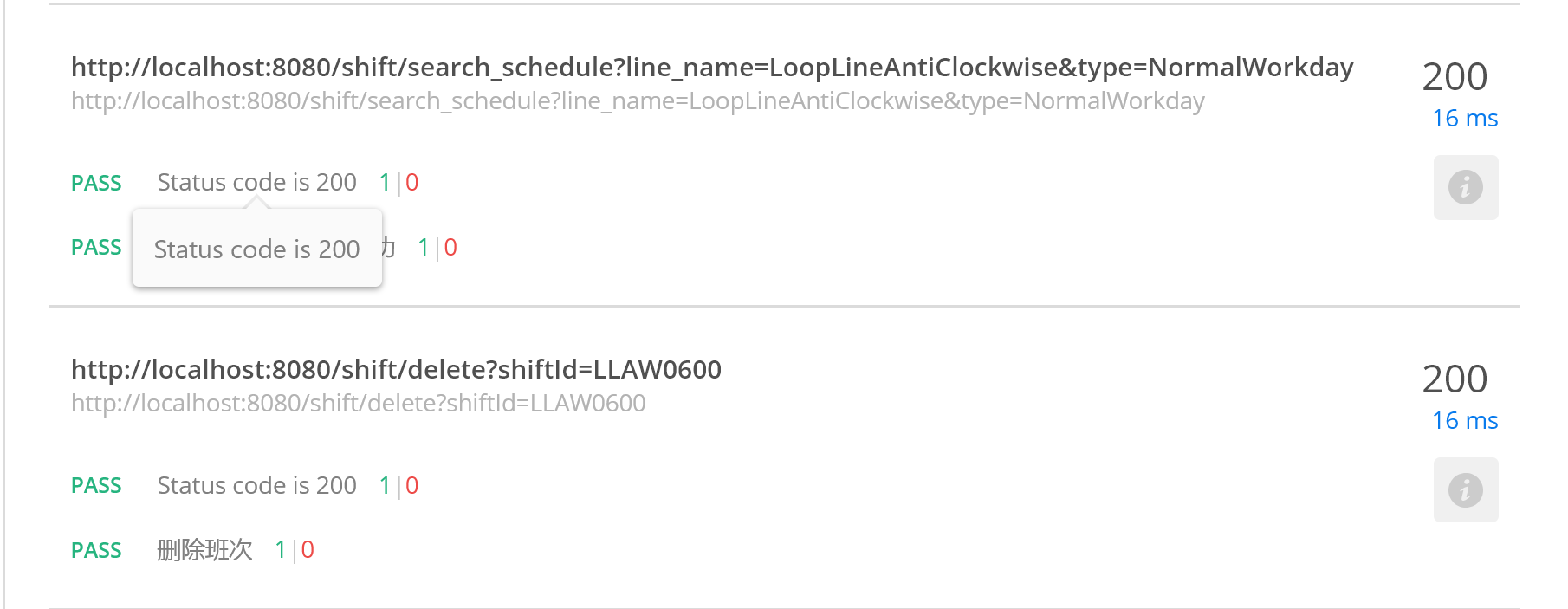


## 和shift相关的接口测试（测试用例数：12 测试结果：全部通过）





# 7．压力测试

压力测试采用了四种并发线程组来模拟多用户同时在线的情况。分别为20个并发线程、50个并发线程、100个并发线程和200个并发线程.

测试指标有4个：平均响应时间（Average Time）、90%响应时间线（90% line）、错误率（error rate）、吞吐率（每秒响应请求数、throughput）

测试接口选择了较为频繁使用、比较有代表性的三个接口：

1. 预约信息接口：<http://106.14.181.49:8080/appointment/infos>
2. 班次时刻表接口：<http://106.14.181.49:8080/shift/search_schedule>
3. 线路站点接口：[http://106.14.181.49:8080/line/stations](http://106.14.181.49:8080/line/stations?line_name=MinHangToXuHui)

以下是测试结果

经过测试我们发现某些接口的吞吐率较低，而由于巴士线路班次信息较为固定，所以决定添加在后端添加Cache策略来优化各项指标。

下图是优化后的各指标，由于预约信息实时性较强，故没有使用cache。

错误率本身为0%，无需进一步优化

可以看出，在添加缓存优化后，响应时间有所提升、而之前吞吐率偏低的班次信息接口从每秒处理约50请求升到每秒处理大于100个请求，达到了我们的期望。