**测试方案和用例**

整个项目分为后端、web端、小程序端，我们对此分别展开测试。

目录

[web端测试 1](#_Toc10999026)

[模块一：登陆 1](#_Toc10999027)

[模块二：用户 1](#_Toc10999028)

[模块三：请假 2](#_Toc10999029)

[请假申请 2](#_Toc10999030)

[请假管理 3](#_Toc10999031)

[模块四：加班 3](#_Toc10999032)

[加班申请列表 3](#_Toc10999033)

[模块五：工时 4](#_Toc10999034)

[工时记录 4](#_Toc10999035)

[模块六：部门 4](#_Toc10999036)

[部门编辑 4](#_Toc10999037)

[模块七：人脸 5](#_Toc10999038)

[更改人脸信息 5](#_Toc10999039)

[人脸信息管理 5](#_Toc10999040)

[模块八：排班 6](#_Toc10999041)

[排班信息列表 6](#_Toc10999042)

[模块九：签到 7](#_Toc10999043)

[小程序端 7](#_Toc10999044)

[一、登陆模块 7](#_Toc10999045)

[二、主页模块 8](#_Toc10999046)

[三、查看班次安排模块 11](#_Toc10999047)

[四、申请/查看请假模块 12](#_Toc10999048)

[五、申请加班模块 14](#_Toc10999049)

[六、查看个人信息模块 15](#_Toc10999050)

[七、查看系统消息模块 20](#_Toc10999051)

[后端测试 22](#_Toc10999052)

[用户相关 22](#_Toc10999053)

[功能点1：用户认证 23](#_Toc10999054)

[功能点2：添加用户 26](#_Toc10999055)

[功能点3：获取指定用户信息 32](#_Toc10999056)

[功能点3：用户列表 37](#_Toc10999057)

[功能点4：修改用户 42](#_Toc10999058)

[功能点5：删除用户 50](#_Toc10999059)

[部门相关 51](#_Toc10999060)

[功能点6：添加部门 52](#_Toc10999061)

[功能点7：修改部门 57](#_Toc10999062)

[功能点8：获取指定部门信息 62](#_Toc10999063)

[功能点9：获取部门列表 67](#_Toc10999064)

[功能点10：删除部门 70](#_Toc10999065)

[工时相关 72](#_Toc10999066)

[功能点11：获取工时记录 72](#_Toc10999067)

[排班相关 81](#_Toc10999068)

[功能点12：添加排班 82](#_Toc10999069)

[功能点13：部门排班 86](#_Toc10999070)

[功能点14：获取排班列表 89](#_Toc10999071)

[功能点15：删除排班 98](#_Toc10999072)

[请假相关 101](#_Toc10999073)

[功能点16：申请请假 101](#_Toc10999074)

[功能点17：获取指定用户请假 104](#_Toc10999075)

[功能点18：获取请假列表 112](#_Toc10999076)

[功能点19：审批请假 117](#_Toc10999077)

[功能点20：销假 121](#_Toc10999078)

[加班相关 123](#_Toc10999079)

[功能点21：申请加班 123](#_Toc10999080)

[功能点22：获取指定用户加班 125](#_Toc10999081)

[功能点23：获取加班申请列表 131](#_Toc10999082)

[功能点24：审批加班 136](#_Toc10999083)

[人脸相关 140](#_Toc10999084)

[功能点25：更新指定用户人脸信息 140](#_Toc10999085)

[功能点26：获取指定用户可用的人脸信息 143](#_Toc10999086)

[功能点27：获取人脸列表 146](#_Toc10999087)

[功能点28：审批人脸信息 149](#_Toc10999088)

[签到相关 151](#_Toc10999089)

[功能点29：获取二维码 151](#_Toc10999090)

[功能点30：获取当前用户签到状态 153](#_Toc10999091)

[功能点31：二维码签到 157](#_Toc10999092)

[功能点32：人脸签到 160](#_Toc10999093)

[信息相关 160](#_Toc10999094)

[功能点33：获取信息列表 161](#_Toc10999095)

[功能点34：获取指定信息 170](#_Toc10999096)

# web端测试

## 模块一：登陆

前置条件：处于登陆页面

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 用例名称 | 操作步骤 | 预期结果 | 测试结果 |
| 0001 | 输入符合长度要求的账号和正确密码 | 1. 点击账号，正常输入账号 2. 点击密码，输入账号对应密码 3. 点击登陆 | 正常登入系统 | 正常登入系统，并跳转到系统界面 |
| 0002 | 输入符合长度要求的账号和错误密码 | 1. 点击账号，正常输入账号 2. 点击密码，输入错误密码 3. 点击登陆 | 出现错误提示 | 弹出气泡提示“账号或密码错误” |
| 0003 | 输入不符合长度要求的账号 | 1. 点击账号，输入长度不符合要求的账号 2. 点击密码，准备输入 | 出现错误提示 | 账号栏下方提示“账号长度应大于2个字” |
| 0004 | 输入符合长度要求的账号和不符合长度要求的密码 | 1. 点击账号，正常输入账号 2. 点击密码，输入长度小于4位的密码 | 出现错误提示 | 密码栏下方提示“密码长度应大于4位” |
| 0005 | 账号为空，输入密码 | 点击密码，输入密码 | 出现错误提示 | 账号栏下方提示“账号不为空” |
| 0006 | 正确输入账号，密码为空 | 1. 点击账号，正确输入账号 2. 点击登陆 | 出现错误提示 | 密码栏下方提示“密码不为空” |

## 模块二：用户

前置条件：管理员登陆，处于用户管理页面

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 用例名称 | 操作步骤 | 预期结果 | 测试结果 |
| 0001 | 查询用户，输入正确的用户名称 | 1. 在导航栏内输入想要查询的用户的姓名 2. 点击查询 | 下方列表显示对应用户 | 下方列表显示对应用户 |
| 0002 | 查询用户，输入错误的用户名称 | 1. 在导航栏内输入错误的用户姓名 2. 点击查询 | 不会显示任何用户 | 下方列表显示暂无数据 |
| 0003 | 新增用户 | 1. 点击导航栏的新增按钮，弹出新增窗口 2. 输入对应的信息 3. 点击提交 4. 点击确认提交窗口中的确认按钮 | 成功添加新的用户 | 添加新的用户，并跳转到用户管理页面，用户列表中出现新添加的用户 |
| 0004 | 新增用户，但是姓名或密码为空 | 1. 点击导航栏的新增按钮，弹出新增窗口 2. 输入除姓名以及密码外其它信息 | 出现错误提示 | 姓名栏下方提示“姓名不为空”；密码栏下方提示“密码不为空” |
| 0005 | 新增用户，但是姓名或密码长度不符合要求 | 1. 点击导航栏的新增按钮，弹出新增窗口 2. 输入长度不符合要求的姓名以及密码 | 出现错误提示 | 姓名栏下方提示“姓名长度应大于2个字”；密码栏下方提示“密码长度应大于4位” |
| 0006 | 编辑用户 | 1. 选择想要编辑的用户，点击对应用户右侧的编辑按钮 2. 编辑信息 3. 点击提交 4. 点击确认提交窗口中的确认按钮 | 成功编辑用户 | 下方列表显示的对应用户信息发生改变 |
| 0007 | 删除用户 | 1. 选择想要删除的用户，点击对应用户右侧的删除按钮 2. 点击确认窗口中的确认 | 成功删除用户 | 下方列表对应的用户消失 |

## 模块三：请假

### **请假申请**

**前置条件：普通用户登陆，处于请假申请页面**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 用例名称 | 操作步骤 | 预期结果 | 测试结果 |
| 0001 | 申请请假，输入加班信息 | 1. 选择开始时间和结束时间 2. 输入请假理由 3. 点击提交 | 弹出窗口提示申请成功 | 弹出窗口提示申请成功 |

### **请假管理**

**前置条件：管理员登陆，处于请假审批页面**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 用例名称 | 操作步骤 | 预期结果 | 测试结果 |
| 0001 | 查询请假信息，输入正确用户姓名 | 1. 在导航栏输入正确的用户姓名 2. 点击查询 | 显示用户的请假信息 | 下方列表显示对应用户请假信息 |
| 0002 | 查询请假信息，输入错误用户姓名 | 1. 在导航栏输入错误的用户姓名 2. 点击查询 | 不会显示任何用户的请假信息，或显示其它用户（对应输入id）的请假信息 | 不会显示任何用户的请假信息，或显示其它用户（对应输入id）的请假信息 |
| 0003 | 审核请假 | 1. 点击审核按钮 2. 选择相应审核状态 | 成功审核请假信息，状态更改 | 状态更改成功 |
| 0004 | 翻页 | 点击翻页按钮 | 显示相应页信息 | 显示相应页信息 |

## 模块四：加班

### **加班申请列表**

**前置条件：管理员登陆，处于加班申请列表页面**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 用例名称 | 操作步骤 | 预期结果 | 测试结果 |
| 0001 | 查询加班信息，输入正确用户id | 1. 在导航栏输入正确的用户id 2. 点击查询 | 显示用户的加班信息 | 下方列表显示对应用户请假信息 |
| 0002 | 查询加班信息，输入错误用户id | 1. 在导航栏输入错误的用户id 2. 点击查询 | 不会显示任何用户的加班信息，或显示其它用户（对应输入id）的加班信息 | 不会显示任何用户的加班信息，或显示其它用户（对应输入id）的加班信息 |
| 0003 | 审核加班 | 1. 点击审核按钮 2. 选择相应审核状态 | 成功审核加班信息，状态更改 | 状态更改成功 |
| 0004 | 翻页 | 点击翻页按钮 | 显示相应页信息 | 显示相应页信息 |

## 模块五：工时

### **工时记录**

**前置条件：管理员登陆，处于工时记录页面**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 用例名称 | 操作步骤 | 预期结果 | 测试结果 |
| 0001 | 页面加载 | 用户进入工时记录页面 | 页面显示所有工时记录 | 页面显示所有工时记录 |
| 0002 | 翻页 | 点击翻页按钮 | 显示相应页信息 | 显示相应页信息 |
| 0003 | 查询指定时间范围内工时记录 | 1. 选择开始时间以及结束时间 2. 点击查询按钮 | 页面列表显示指定时间范围内的工时记录 | 页面列表显示指定时间范围内的工时记录 |

## 模块六：部门

### **部门编辑**

**前置条件：管理员登陆，处于部门编辑页面**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 用例名称 | 操作步骤 | 预期结果 | 测试结果 |
| 0001 | 查询部门信息，输入正确的部门名称 | 1. 在导航栏内输入想要查询的部门的名称 2. 点击查询 | 下方列表显示对应部门信息 | 下方列表显示对应部门信息 |
| 0002 | 查询用户，输入错误的部门名称 | 1. 在导航栏内输入错误的部门名称 2. 点击查询 | 不会显示任何部门信息 | 下方列表显示暂无数据 |
| 0003 | 新增部门 | 1. 点击导航栏的新增按钮，弹出新增窗口 2. 输入对应的信息 3. 点击提交 4. 点击确认提交窗口中的确认按钮 | 成功添加新的部门 | 添加新的部门，并跳转到部门管理页面，部门列表中出现新添加的部门 |
| 0004 | 新增部门，但是姓名或密码为空 | 1. 点击导航栏的新增按钮，弹出新增窗口 2. 输入除姓名以及密码外其它信息 | 出现错误提示 | 姓名栏下方提示“姓名不为空”；密码栏下方提示“密码不为空” |
| 0005 | 新增用户，但是姓名或密码长度不符合要求 | 1. 点击导航栏的新增按钮，弹出新增窗口 2. 输入长度不符合要求的姓名以及密码 | 出现错误提示 | 姓名栏下方提示“姓名长度应大于2个字”；密码栏下方提示“密码长度应大于4位” |
| 0006 | 编辑用户 | 1. 选择想要编辑的用户，点击对应用户右侧的编辑按钮 2. 编辑信息 3. 点击提交 4. 点击确认提交窗口中的确认按钮 | 成功编辑用户 | 下方列表显示的对应用户信息发生改变 |
| 0007 | 删除用户 | 1. 选择想要删除的用户，点击对应用户右侧的删除按钮 2. 点击确认窗口中的确认 | 成功删除用户 | 下方列表对应的用户消失 |

## 模块七：人脸

### **更改人脸信息**

**前置条件：普通用户登陆，处于人脸信息更新页面**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 用例名称 | 操作步骤 | 预期结果 | 测试结果 |
| 0001 | 打开摄像头 | 点击打开摄像头 | 调用电脑相机，屏幕左侧出现相机 | 屏幕左侧出现相机 |
| 0002 | 拍照 | 1. 点击打开摄像头 2. 点击拍照 | 屏幕右侧出现拍摄的照片 | 屏幕右侧出现拍摄的照片 |
| 0003 | 上传人脸照片 | 1. 点击打开摄像头 2. 点击拍照 3. 点击使用照片 | 拍摄的照片上传到后端 | 拍摄的照片上传到后端 |

### **人脸信息管理**

**前置条件：管理员登陆，处于加班审批页面**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 用例名称 | 操作步骤 | 预期结果 | 测试结果 |
| 0001 | 查询指定用户人脸信息，输入正确的用户id | 1. 在导航栏输入正确的用户id 2. 点击查询 | 显示用户的所有可用人脸信息 | 下方列表显示用户的所有可用人脸信息 |
| 0002 | 查询指定用户人脸信息，输入错误的用户id | 1. 在导航栏输入错误的用户id 2. 点击查询 | 不会显示任何用户的人脸信息，或显示其它用户（对应输入id）的人脸信息 | 不会显示任何用户的人脸信息，或显示其它用户（对应输入id）的人脸信息 |
| 0003 | 查看人脸信息 | 点击对应人脸信息的查看图片按钮 | 可以查看人脸图片 | 弹出人脸图片，便于审核 |
| 0004 | 审核人脸信息 | 1. 点击对应人脸信息的审核按钮 2. 选择相应审核状态 | 成功审核人脸信息，状态更改 | 状态更改成功 |

## 模块八：排班

### **排班信息列表**

**前置条件：管理员登陆，处于排班信息列表页面**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 用例名称 | 操作步骤 | 预期结果 | 测试结果 |
| 0001 | 查询排班信息，输入正确部门名称 | 1. 弹出窗口提示添加成功在导航栏输入正确的部门名称 2. 点击查询 | 显示用户的排班信息 | 下方列表显示对应用户请假信息 |
| 0002 | 查询排班信息，输入错误部门名称 | 1. 在导航栏输入错误的部门名称 2. 点击查询 | 不会显示任何用户的排班信息 | 下方列表显示暂无数据 |
| 0003 | 添加排班信息 | 1. 点击添加排班信息 2. 填入排班信息 3. 点击提交按钮 | 弹出窗口提示添加成功 | 弹出窗口提示添加成功 |
| 0004 | 添加部门排班 | 1. 点击添加部门排班 2. 填入部门排班信息 3. 点击提交按钮 | 弹出窗口提示添加成功 | 弹出窗口提示添加成功 |
| 0004 | 翻页 | 点击翻页按钮 | 显示相应页信息 | 显示相应页信息 |

## 模块九：签到

**前置条件：用户登陆**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 用例名称 | 操作步骤 | 预期结果 | 测试结果 |
| 0001 | 已签过到，再次点击签到 | 点击右上角用户头像，在下拉列表中点击签到 | 显示用户已签到 | 弹出气泡提示用户已签到 |
| 0002 | 未签过到，点击签到 | 1. 点击右上角用户头像，在下拉列表中点击签到 2. 在弹出页面完成人脸签到以及二维码签到 | 签到成功，签到状态改为已签到 | 签到成功，签到状态改为已签到 |

# 小程序端

## 一、登陆模块

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 模块 | 用例描述 | 前置条件 | 操作步骤 | 预期结果 | 测试结果 |
| 1.1 | **登**  **陆** | 不规范账号和规范密码 | 在“登陆页面” | 1.在“账号输入框”，键入长度小于二位的账号；  2.在“密码输入框”，键入长度大于四位的密码；  3.点击“登陆”。 | 1.弹出“账号格式不正确”的提示框；  2.输入框内的账号、密码清空；  3.留在“登陆页面”。 |  |
| 1.2 | 不规范密码和规范账号 | 在“登陆页面” | 1.在“账号输入框”，键入长度大于二位的账号；  2.在“密码输入框”，键入长度小于四位的密码；  3.点击“登陆”。 | 1.弹出“密码格式不正确”的提示框；  2.输入框内的账号、密码清空；  3.留在“登陆页面”。 |  |
| 1.3 | 不规范账号和不规范密码 | 在“登陆页面” | 1.在“账号输入框”，键入长度小于二位的账号；  2.在“密码输入框”，键入长度小于四位的密码；  3.点击“登陆”。 | 1.弹出“账号格式不正确”的提示框；  2.输入框内的账号、密码清空；  3.留在“登陆页面”。 |  |
| 1.4 | 密码显示与隐藏 | 在“登陆页面” | 1.在“密码输入框”键入大于零位的密码；  2.点击“密码输入框”右侧“显示”图标。 | 1.在点击“显示”图标之前，密码为黑色圆点（不可见）；  2.点击“显示”图标之后，密码为输入密码（可见）。 |  |
| 1.5 | 不正确的规范账号和规范密码 | 在“登陆页面” | 1.在“账号输入框”，键入长度大于二位的账号；  2.在“密码输入框”，键入长度大于四位的密码；  （所输入的账号密码为非正确员工所有）  3.点击“登陆”。 | 1.弹出“账号密码不正确”提示框；  2.输入框内的账号、密码清空；  3.留在“登陆页面”。 |  |
| 1.6 | 正确的规范账号和规范密码 | 在“登陆页面” | 1.在“账号输入框”，键入长度大于二位的账号；  2.在“密码输入框”，键入长度大于四位的密码；  （所输入的账号密码为正确员工所有）  3.点击“登陆”。 | 1.进入“主页页面”。 |  |

## 二、主页模块

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 2.1 | **主**  **页** | 登陆token验证（未过期） | 距离上次“正确登陆”还未过24小时 | 进入小程序 | 直接进入“主页页面”。 |  |
| 2.2 | 登陆token验证（已过期） | 距离上次“正确登陆”过了24小时 | 进入小程序 | 1.显示“token已过期”；  2.进入“登陆页面” |  |
| 2.3 | 用户名显示 | 在“主页页面” | 无 | 显示“某某某，您好”的“欢迎词” |  |
| 2.4 | 签到验证（未签到） | 1.在“主页页面”；  2.今日还未签到。 | 无 | 1.“签到情况”显示“未签到”；  2.出现可点击的“扫码”图标；  3.无可点击的“签退”按键。  4.无“班次信息显示框” |  |
| 2.5 | 签到验证（已签到） | 1.在“主页页面”；  2.今日已签到。 | 无 | 1.“签到情况”显示“已签到”；  2.无可点击的“扫码”图标。  3.出现可点击的“签退”按键。  4.有“班次信息显示框” |  |
| 2.6 | 班次信息显示 | 1.在“主页页面”；  2.已签到。 | 无 | “班次信息”显示用户当日的“班次信息” |  |
| 2.7 | 正确扫码 | 1.在“主页页面”  2.今日还未签到 | 1.单击扫码；  2.扫描正确的签到二维码图片。 | 1.“主页页面”刷新；  2.显示已签到。 |  |
| 2.8 | 非正确扫码 | 1.在“主页页面”  2.今日还未签到 | 1.点击扫码；  2.扫描非正确的签到二维码。 | 1.“主页页面”刷新；  2.显示未签到。 |  |
| 2.9 | 进入“查看班次安排” | 在“主页页面” | 单击“查看班次安排”按键 | 进入“查看班次安排页面” |  |
| 2.10 | 进入“申请/查看请假页面” | 在“主页页面” | 单击“申请/查看请假页面”按键 | 进入“申请/查看请假页面” |  |
| 2.11 | 进入“申请加班页面” | 1.在“主页页面”；  2.有可点击的“申请加班”按键。 | 单击“申请加班”按键 | 进入“申请加班页面” |  |
| 2.12 | 进入“查看个人信息页面” | 在“主页页面” | 单击“查看个人信息”按键 | 进入“查看个人信息按键” |  |
| 2.13 | 进入“系统发送信息” | 在“主页页面” | 单击“欢迎词”按键 | 进入“系统发送信息页面” |  |
| 2.14 | 下班前签退 | 1.在“主页页面”；  2.有可点击的“签退”按键。  3.当前时间在下班前。 | 单击“签退”按键 | 1.弹出“成功签退”的提示框；  2.无可点击的“申请加班”按键。 |  |
| 2.15 | 下班后签退 | 1.在“主页页面”；  2.今日已签到；  3.当前时间在下班后。 | 单击“签退”按键 | 1.弹出“成功签退”的提示框；  2.出现可点击的“申请加班”按键。 |  |
| 2.16 | “定时查询系统”弹出提醒 | 1.加载过“主页页面”；  2.有发送给当前用户且状态为“unread”的消息；  3.在任一页面。 | 无 | 弹出“有未读消息”的提示框 |  |
| 2.17 | “未读消息提示框”对“定时查询系统”控制（仅关闭） | 1.在任一页面”；  2.弹出“有未读消息”的提示框；  3.未“处理”未读消息。 | 单击提示框的“确认”按键 | 1.提示框消失；  2.只要未读消息未被“处理”，系统便不再弹出“有未读消息”的提示框，。 |  |
| 2.18 | “主页页面的显示”对“定时查询系统”的控制（开启） | 1.通过其他页面返回“主页页面”；  2.（在“未读消息”页面）“处理”完所有未读消息 | 无 | “定时查询系统”开启（由系统控制台可观测到），此时之后若有新消息，将弹出提示框 |  |
| 2.19 | “主页页面的显示”对“定时查询系统”的控制（关闭） | 1.通过其他页面返回“主页页面”；  2.（在“未读消息”页面）未“处理”完所有未读消息 | 无 | “定时查询系统”未开启（由系统控制台可观测到），此时之后若有新消息，不会弹出提示框 |  |
| 2.20 | “主页页面”关闭对“定时查询系统”的控制（关闭） | 在任一页面 | 关闭“主页页面”（退出小程序） | “定时查询系统”关闭（由系统控制台可观测到） |  |

## 三、查看班次安排模块

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3.1 | **查**  **看**  **班**  **次**  **安**  **排** | 查看显示的当前月份的某一日的班次安排 | 在“查看班次安排页面” | 单击日历上的某一日 | 1.弹出对应该日的正确的“班次安排信息框”；  2.“红色标记”聚焦在该日。 |  |
| 3.2 | “班次安排信息框”的关闭 | 1.在“查看班次安排页面；  2.已弹出在“查看班次安排信息框”。 | 单击框外的任一位置 | “查看班次安排信息框”消失 |  |
| 3.3 | 改变月份（增和减） | 1.在“查看班次安排页面；  2.无“查看班次安排信息框”。 | 单击“月份栏”的“向左（右）”按键 | 月份相应减（加）一 |  |
| 3.4 | 查看改变月份后的某一日班次安排（该日安排班次） | 1.在“查看班次安排页面；  2.无“查看班次安排信息框”；  3.当前显示月份不是初始月份。 | 单击日历上的某一日且该日安排班次 | 1.弹出对应该日的正确的“班次安排信息框”；  2.“红色标记”聚焦在该日。 |  |
| 3.5 | 查看改变月份后的某一日班次安排（该日未安排班次） | 1.在“查看班次安排页面；  2.无“查看班次安排信息框”；  3.当前显示月份不是初始月份。 | 单击日历上的某一日且该日未安排班次 | 1.弹出“该日未安排工作”的提示框；  2.“红色标记”聚焦在该日。 |  |
| 3.6 | 从任一月份回到当月当日 | 1.在“查看班次安排页面；  2.无“查看班次安排信息框”；  3.当前显示月份不是初始月份。 | 单击“回到今天”按键 | 1.显示日历回到为当月；  2.“红色标记”聚焦在当日。 |  |
| 3.7 | 回到主页 | 在“查看班次安排页面” | 单击左上角的“返回”按键 | 回到“主页页面” |  |

## 四、申请/查看请假模块

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 4.1 | **申**  **请**  **和**  **查**  **看**  **请**  **假** | 请假起始日期的选择 | 在“申请/查看请假页面” | 1.单击“请假起始日期时间选择”按键；  2.选择“年月日时分秒”；  3.单击“确定（取消）”按键。 | 1.从屏幕下方弹出可选择“年月日时分秒”的“选择框”；  2.可以滚动“选择框”中的相应选项；  3.单击“确认（取消）”后，  选择框消失且“请假起始日期时间显示框”（不）改变为选择的时间 |  |
| 4.2 | 请假结束日期的选择 | 在“申请/查看请假页面” | 1.单击“请假结束日期时间选择”按键；  2.选择“年月日时分秒”；  3.单击“确定（取消）”按键。 | 1.从屏幕下方弹出可选择“年月日时分秒”的“选择框”；  2.可以滚动“选择框”中的相应选项；  3.单击“确认（取消）”后，选择框消失且“请假结束日期时间显示框”（不）改变为选择的时间 |  |
| 4.3 | 不规范起始、结束时间和不规范申请加班理由 | 在“申请/查看请假页面” | 1.选择的起始日期晚于结束日期；  2.填写字数小于3或大于1000；  3.单击“提交请假。申请”按键 | 弹出“结束日期应大于开始日期”的提示框 |  |
| 4.4 | 不规范起始、结束时间和规范申请加班理由 | 在“申请/查看请假页面” | 1.选择的起始日期晚于结束日期；  2.填写字数大于等于3或小于等于1000；  3.单击“提交请假申请”按键。 | 弹出“结束日期应大于开始日期”的提示框 |  |
| 4.5 | 规范起始、结束时间和不规范申请加班理由 | 在“申请/查看请假页面” | 1.选择的起始日期早于结束日期；  2.单击文本输入框；  3.填写字数小于3或大于1000；  4.单击“提交请假申请”按键。 | 1.单击文本输入框后弹出软键盘；  2.弹出“理由应在3-1000个字”的提示框 |  |
| 4.6 | 规范起始、结束时间和规范申请加班理由 | 在“申请/查看请假页面” | 1.选择的起始日期早于结束日期；  2.单击文本输入框；  3.填写字数大于等于3或小于等于1000；  4.单击“提交请假申请”按键。 | 1.单击文本输入框后弹出软键盘；  2.弹出“提交成功”的提示框 |  |
| 4.7 | 回到主页 | 在“申请/查看请假页面” | 单击左上角的“返回”按键 | 回到“主页页面” |  |
| 4.8 | 查看已申请的请假 | 在“申请/查看请假页面” | 单击“查看已申请的请假”按键 | 进入“进行中的请假申请页面” |  |
| 4.9 | 选择请假项目 | 在“进行中的请假申请页面” | 1.单击“当前选择”信息框；  2.滑动选定某一请假项目；  3.单击“确定（取消）”按键。 | 1.弹出当前用户进行中的请假申请的“项目”选择框；  2.可以滚动选择框中的选项；  3.单击“确定（取消）”后，选择框消失且“当前选择”信息框和请假相关信息框（不）改变为所选择的请假项目。 |  |
| 4.10 | 显示请假相关信息 | 在“进行中的请假申请页面” | 无 | 请假相关信息框为“当前选择”信息框中对应请假项目的相关信息 |  |
| 4.11 | 销假（允许） | 1.在“进行中的请假申请页面”；  2.当前选择的请假项目的状态为“通过”。 | 单击“销假”按键 | 1.“销假”按键显示状态为“可点击”；  2.单击“销假”按键后弹出“销假成功”的提示框；  3.请假相关信息框中的状态标识更新为“销假”。 |  |
| 4.12 | 销假（不允许） | 1.在“进行中的请假申请页面”；  2.当前选择的请假项目的状态为“不通过”。 | 无 | 1.“销假”按键显示状态为“不可点击”；  2.点击“销假”按键，无反应。 |  |
| 4.13 | 回到申请请假 | 在“进行中的请假申请页面” | 单击左上角的“返回”按键 | 回到“申请/查看请假页面” |  |

## 五、申请加班模块

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 5.1 | **申**  **请**  **加**  **班** | 不规范申请加班理由 | 在“申请加班页面” | 1.单击文本输入框；  2.填写字数小于3或大于1000；  3.单击“Submit”按键 | 1.单击文本输入框后弹出软键盘；  2.弹出“理由应在3-1000个字”的提示框。 |  |
| 5.2 | 规范申请加班理由 | 在“申请加班页面” | 1.单击文本输入框；  2.填写字数大于等于3或小于等于1000；  3.单击“Submit”。 | 1.单击文本输入框后弹出软键盘；  2.弹出“提交成功”的提示框。 |  |
| 5.3 | 重置文本框 | 1.在“申请加班页面”；  2.文本框内有文字。 | 单击“Reset”按键 | 文本框内的文字消失 |  |
| 5.4 | 回到主页 | 在“申请加班页面” | 单击左上角的“返回”按键 | 回到“主页页面” |  |

## 六、查看个人信息模块

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 6.1 | **查**  **看**  **个**  **人**  **信**  **息** | 显示个人信息 | 在“查看个人信息页面” | 无 | “个人信息显示框”正确显示相关信息 |  |
| 6.2 | 选择图片 | 在“查看个人信息页面” | 1.单击“选择照片”按键；  2.选择某一照片并确认。 | 1.弹出相册可选择照片；  2.选择照片时只能选择一张；  3.选择新照片并确认后“个人信息显示框”中的“图片”区域更新为新照片。 |  |
| 6.3 | 上传图片（有新图片） | 1.在“查看个人信息页面”；  2.已选择新照片。 | 单击“确认上传”按键 | 弹出“图片上传成功”的提示框 |  |
| 6.4 | 上传图片（无新图片） | 1.在“查看个人信息页面”；  2.未选择新照片。 | 单击“确认上传”按键 | 弹出“您还未选择照片”的提示框 |  |
| 6.5 | 回到主页 | 在“查看个人信息页面” | 单击左上角的“返回”按键 | 回到“主页页面” |  |
| 6.6 | 修改密码 | 在“查看个人信息页面” | 单击“修改密码”按键 | 弹出“修改密码”框 |  |
| 6.7 | 规范的“原密码”和规范的“新密码”和规范的“新密码确认”且“新密码”与“新密码确认”一致 | 1.在“查看个人信息页面”；  2.已弹出“修改密码”框。 | 1.在“原密码输入框”输入位数不小于4的密码；  2.在“新密码输入框”输入位不小于4的密码；  3.在“新密码确认输入框”输入位数不小于4且与“新密码输入框”一致的密码；  4.单击提交。 | 弹出“密码修改成功”的提示框 |  |
| 6.8 | 不规范的“原密码”和规范的“新密码”和规范的“新密码确认”且“新密码”与“新密码确认”一致 | 1.在“查看个人信息页面”；  2.已弹出“修改密码”框。 | 1.在“原密码输入框”输入位数小于4的密码；  2.在“新密码输入框”输入位不小于4的密码；  3.在“新密码确认输入框”输入位数不小于4且与“新密码输入框”一致的密码；  4.单击提交。 | 弹出“密码最短为4位”的提示框 |  |
| 6.9 | 规范的“原密码”和不规范的“新密码”和规范的“新密码确认”且“新密码”与“新密码确认”一致 | 1.在“查看个人信息页面”；  2.已弹出“修改密码”框。 | 1.在“原密码输入框”输入位数不小于4的密码；  2.在“新密码输入框”输入位小于4的密码；  3.在“新密码确认输入框”输入位数不小于4且与“新密码输入框”一致的密码；  4.单击提交。 | 弹出“密码最短为4位”的提示框 |  |
| 6.10 | 规范的“原密码”和规范的“新密码”和不规范的“新密码确认”且“新密码”与“新密码确认”一致 | 1.在“查看个人信息页面”；  2.已弹出“修改密码”框。 | 1.在“原密码输入框”输入位数不小于4的密码；  2.在“新密码输入框”输入位不小于4的密码；  3.在“新密码确认输入框”输入位数小于4且与“新密码输入框”一致的密码；  4.单击提交。 | 弹出“密码最短为4位”的提示框 |  |
| 6.11 | 规范的“原密码”和不规范的“新密码”和不规范的“新密码确认”且“新密码”与“新密码确认”一致 | 1.在“查看个人信息页面”；  2.已弹出“修改密码”框。 | 1.在“原密码输入框”输入位数不小于4的密码；  2.在“新密码输入框”输入位小于4的密码；  3.在“新密码确认输入框”输入位数小于4且与“新密码输入框”一致的密码；  4.单击提交。 | 弹出“密码最短为4位”的提示框 |  |
| 6.12 | 不规范的“原密码”和规范的“新密码”和不规范的“新密码确认”且“新密码”与“新密码确认”一致 | 1.在“查看个人信息页面”；  2.已弹出“修改密码”框。 | 1.在“原密码输入框”输入位数小于4的密码；  2.在“新密码输入框”输入位不小于4的密码；  3.在“新密码确认输入框”输入位数小于4且与“新密码输入框”一致的密码；  4.单击提交。 | 弹出“密码最短为4位”的提示框 |  |
| 6.13 | 不规范的“原密码”和不规范的“新密码”和规范的“新密码确认”且“新密码”与“新密码确认”一致 | 1.在“查看个人信息页面”；  2.已弹出“修改密码”框。 | 1.在“原密码输入框”输入位数小于4的密码；  2.在“新密码输入框”输入位小于4的密码；  3.在“新密码确认输入框”输入位数不小于4且与“新密码输入框”一致的密码；  4.单击提交。 | 弹出“密码最短为4位”的提示框 |  |
| 6.14 | 不规范的“原密码”和不规范的“新密码”和不规范的“新密码确认”且“新密码”与“新密码确认”一致 | 1.在“查看个人信息页面”；  2.已弹出“修改密码”框。 | 1.在“原密码输入框”输入位数小于4的密码；  2.在“新密码输入框”输入位小于4的密码；  3.在“新密码确认输入框”输入位数小于4且与“新密码输入框”一致的密码；  4.单击提交。 | 弹出“密码最短为4位”的提示框 |  |
| 6.15 | 规范的“原密码”和规范的“新密码”和规范的“新密码确认”且“新密码”与“新密码确认”不一致 | 1.在“查看个人信息页面”；  2.已弹出“修改密码”框。 | 1.在“原密码输入框”输入位数不小于4的密码；  2.在“新密码输入框”输入位不小于4的密码；  3.在“新密码确认输入框”输入位数不小于4且与“新密码输入框”不一致的密码；  4.单击提交。 | 弹出“两次输入的新密码不一致”的提示框 |  |
| 6.16 | 不规范的“原密码”和规范的“新密码”和规范的“新密码确认”且“新密码”与“新密码确认”不一致 | 1.在“查看个人信息页面”；  2.已弹出“修改密码”框。 | 1.在“原密码输入框”输入位数小于4的密码；  2.在“新密码输入框”输入位不小于4的密码；  3.在“新密码确认输入框”输入位数不小于4且与“新密码输入框”不一致的密码；  4.单击提交。 | 弹出“密码最短为4位”的提示框 |  |
| 6.17 | 规范的“原密码”和不规范的“新密码”和规范的“新密码确认”且“新密码”与“新密码确认”不一致 | 1.在“查看个人信息页面”；  2.已弹出“修改密码”框。 | 1.在“原密码输入框”输入位数不小于4的密码；  2.在“新密码输入框”输入位小于4的密码；  3.在“新密码确认输入框”输入位数不小于4且与“新密码输入框”不一致的密码；  4.单击提交。 | 弹出“密码最短为4位”的提示框 |  |
| 6.18 | 规范的“原密码”和规范的“新密码”和不规范的“新密码确认”且“新密码”与“新密码确认”不一致 | 1.在“查看个人信息页面”；  2.已弹出“修改密码”框。 | 1.在“原密码输入框”输入位数不小于4的密码；  2.在“新密码输入框”输入位不小于4的密码；  3.在“新密码确认输入框”输入位数小于4且与“新密码输入框”不一致的密码；  4.单击提交。 | 弹出“密码最短为4位”的提示框 |  |
| 6.19 | 规范的“原密码”和不规范的“新密码”和不规范的“新密码确认”且“新密码”与“新密码确认”不一致 | 1.在“查看个人信息页面”；  2.已弹出“修改密码”框。 | 1.在“原密码输入框”输入位数不小于4的密码；  2.在“新密码输入框”输入位小于4的密码；  3.在“新密码确认输入框”输入位数小于4且与“新密码输入框”不一致的密码；  4.单击提交。 | 弹出“密码最短为4位”的提示框 |  |
| 6.20 | 不规范的“原密码”和规范的“新密码”和不规范的“新密码确认”且“新密码”与“新密码确认”不一致 | 1.在“查看个人信息页面”；  2.已弹出“修改密码”框。 | 1.在“原密码输入框”输入位数小于4的密码；  2.在“新密码输入框”输入位小于4的密码；  3.在“新密码确认输入框”输入位数不小于4且与“新密码输入框”不一致的密码；  4.单击提交。 | 弹出“密码最短为4位”的提示框 |  |
| 6.21 | 不规范的“原密码”和不规范的“新密码”和规范的“新密码确认”且“新密码”与“新密码确认”不一致 | 1.在“查看个人信息页面”；  2.已弹出“修改密码”框。 | 1.在“原密码输入框”输入位数小于4的密码；  2.在“新密码输入框”输入位小于4的密码；  3.在“新密码确认输入框”输入位数不小于4且与“新密码输入框”不一致的密码；  4.单击提交。 | 弹出“密码最短为4位”的提示框 |  |
| 6.22 | 不规范的“原密码”和不规范的“新密码”和不规范的“新密码确认”且“新密码”与“新密码确认”不一致 | 1.在“查看个人信息页面”；  2.已弹出“修改密码”框。 | 1.在“原密码输入框”输入位数小于4的密码；  2.在“新密码输入框”输入位小于4的密码；  3.在“新密码确认输入框”输入位数小于4且与“新密码输入框”不一致的密码；  4.单击提交。 | 弹出“密码最短为4位”的提示框 |  |
| 6.23 | 直接重置 | 1.在“查看个人信息页面”；  2.已弹出“修改密码”框；  3.已在任一输入框输入密码。 | 单击“重置”按键 | 输入框的输入内容清空 |  |
| 6.24 | 间接重置 | 1.在“查看个人信息页面”；  2.已弹出“修改密码”框；  3.已在任一输入框输入密码。 | 1.关闭“修改密码框”；  2.单击“修改密码”按键。 | 输入框的输入内容为空 |  |
| 6.25 | 密码显示与隐藏 | 1.在“查看个人信息页面”；  2.已弹出“修改密码”框。 | 1.在“密码输入框”键入大于零位的密码；  2.点击“输入框”右侧的“显示”图标。 | 1.在点击“显示”图标之前，密码为黑色圆点（不可见）；  2.点击“显示”图标之后，密码为输入密码（可见）。 |  |

## 七、查看系统消息模块

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 7.1 | **查**  **看**  **系**  **统**  **消**  **息** | 显示未读消息 | 在“查看系统发送消息（未读）页面” | 无 | 显示系统发送且状态为“unread”的消息及其相关信息 |  |
| 7.2 | 查看某条未读消息（不止一条消息） | 1.在“查看系统发送消息（未读）页面”；  2.有不止一条的未读消息。 | 单击某条消息的“查看”按键 | 1.弹出该消息的具体内容框；  2.该内容被标记为“read”且从页面中消失； |  |
| 7.3 | 查看某条未读消息（只有一条消息） | 1.在“查看系统发送消息（未读）页面”；  2.只有一条未读消息。 | 单击该条消息的“查看”按键 | 1.弹出该消息的具体内容框；  2.该内容被标记为“read”且从页面中消失；  3.将控制“主页页面”中的“定时查询系统”在缓存中的变量设置为“true”（由缓存信息观测到）。 |  |
| 7.4 | 清除所有消息（有消息） | 1在“查看系统发送消息（未读）页面”；  2.有未读消息。 | 1.单击“清除所有消息”按键；  2.在弹出的“是否确认清除”提示框中单击“确认（取消）”按键。 | 1.弹出“是否确认清除”提示框；  2.单击“确认（取消）”按键后，所有消息（不）被标记为“read”且（不）从页面中消失；  3.将控制“主页页面”中的“定时查询系统”在缓存中的变量设置为“true”（由缓存信息观测到）。 |  |
| 7.5 | 清除所有消息（无消息） | 1在“查看系统发送消息（未读）页面”；  2.无未读消息。 | 单击“清除所有消息”按键； | 弹出“消息已为空”提示框 |  |
| 7.6 | 未读消息下拉刷新 | 在“查看系统发送消息（未读）页面” | 下拉页面 | 页面将刷新，若此时有新的未读消息，将会显示在页面中 |  |
| 7.7 | 回到主页 | 在“查看系统发送消息（未读）页面” | 单击左上角的“返回”按键 | 回到“主页页面” |  |
| 7.8 | 查看已读消息 | 在“查看系统发送消息（未读）页面” | 单击“查看已读消息”按键 | 进入“查看系统发送消息（已读）页面” |  |
| 7.9 | 显示已读消息 | 在“查看系统发送消息（已读）页面 | 无 | 显示系统发送且状态为“read”的消息及其相关信息 |  |
| 7.10 | 查看某条已读消息 | 1.在“查看系统发送消息（已读）页面；  2.有已读消息。 | 单击某消息的“查看”按键 | 弹出该消息的具体内容框； |  |
| 7.11 | 回到未读消息 | 在“查看系统发送消息（已读）页面 | 单击左上角的“返回”按键 | 回到“查看系统发送消息（未读）页面” |  |
| 7.12 | 已读消息下拉刷新 | 在“查看系统发送消息（已读）页面” | 下拉页面 | 页面将刷新，若此时有新的已读消息，将会显示在页面中 |  |

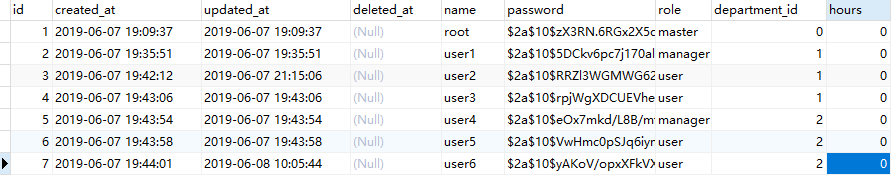
# 后端测试

说明：初始数据库为空，本机运行程序后，数据库自动迁移构造各个数据表，同时初始root用户。

为了方便进行测试，会在测试过程中使用上一步测试留下的结果

## 用户相关

参考结构：



### 功能点1：用户认证

#### 001：输入用户名错误

##### 输入：

URL：

POST: http://127.0.0.1:8000/user/auth

JSON：

{

"name": "1234",

"password": "1234"

}

##### 预期输出：

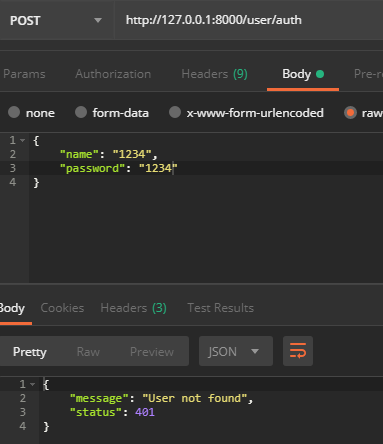
{

"message": "User not found",

"status": 401

}

##### 测试结果：



#### 002：输入用户名正确，密码错误

##### 输入：

URL：

POST: http://127.0.0.1:8000/user/auth

JSON：

{

"name": "root",

"password": "12"

}

##### 预期输出：

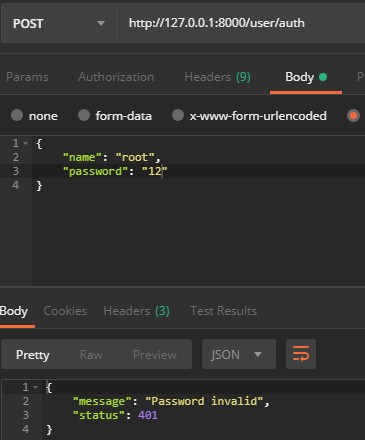
{

"message": "Password invalid",

"status": 401

}

##### 测试结果：



#### 003：输入用户名和密码都正确

##### 输入：

URL：

POST: http://127.0.0.1:8000/user/auth

JSON：

{

"name": "root",

"password": "root"

}

##### 预期输出：

{

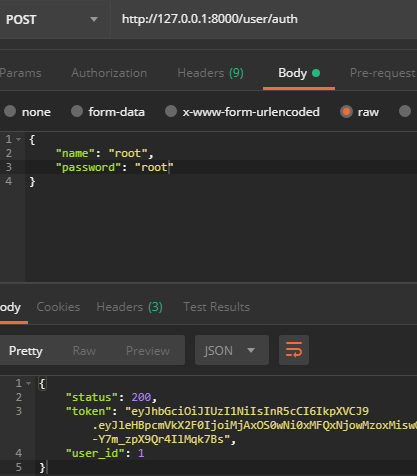
"status": 200,

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHBpcmVkX2F0IjoiMjAxOS0wNi0xMFQxNjowMzoxMiswODowMCIsImlkIjoxLCJyb2xlIjoibWFzdGVyIn0.MhvJ0DWrF8OLrCnaWQpB-Vm-Y7m\_zpX9Qr4IlMqk7Bs",

"user\_id": 1

}

测试结果：



### 功能点2：添加用户

#### 004：输入名字小于最小长度

##### 输入：

URL：

POST http://127.0.0.1:8000/user

JSON：

{

"name": "u",

"password": "1234567",

"department": 1

}

##### 预期输出：

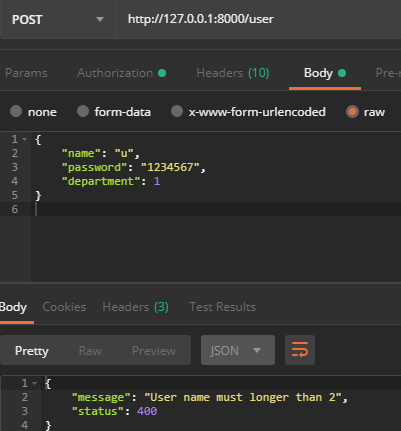
{

"message": "User name must longer than 2",

"status": 400

}

##### 测试结果：



#### 005：输入名字重复

##### 输入：

URL：

POST http://127.0.0.1:8000/user

JSON：

{

"name": "root",

"password": "1234",

"department": 1

}

##### 预期输出：

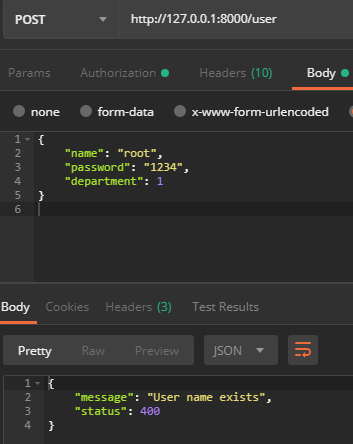
{

"message": "User name exists",

"status": 400

}

##### 测试结果：



#### 006：输入密码小于最小长度

##### 输入：

URL：

POST http://127.0.0.1:8000/user

JSON：

{

"name": "user1",

"password": "12",

"department": 1

}

##### 预期输出：

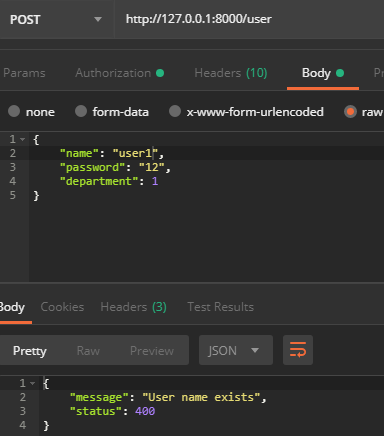
{

"message": "Password must longer than 7",

"status": 400

}

##### 测试结果：



#### 007：输入用户部门不存在

##### 输入：

URL：

POST http://127.0.0.1:8000/user

JSON：

{

"name": "user1",

"password": "1234567",

"department": 0

}

##### 预期输出：

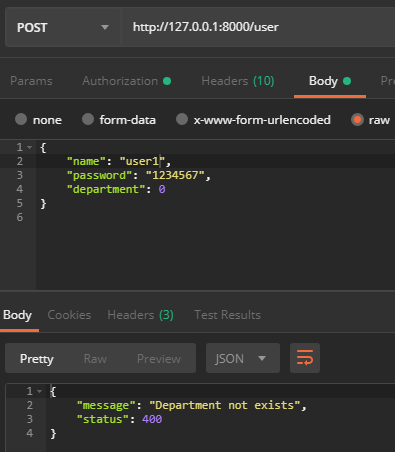
{

"message": "Department not exists",

"status": 400

}

##### 测试结果：



#### 008：正确创建

##### 输入：

URL：

POST http://127.0.0.1:8000/user

JSON：

{

"name": "user11",

"password": "1234567",

"department": 1

}

##### 预期输出：

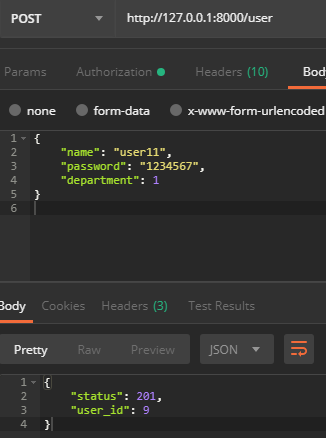
{

"status": 201,

"user\_id": 9

}

##### 测试结果：



### 功能点3：获取指定用户信息

#### 009：用户获取他人的信息

##### 输入：

登录员工user2

URL：

GET http://127.0.0.1:8000/user/3

##### 预期输出：

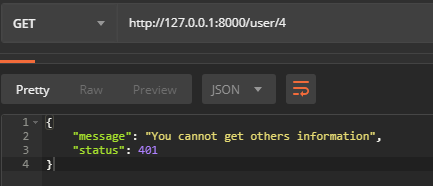
{

"message": "You cannot get others information",

"status": 401

}

##### 测试结果：



#### 010：用户获取自己的信息

##### 输入：

登录员工user2

URL：

GET http://127.0.0.1:8000/user/2

##### 预期输出：

{

"data": {

"id": 3,

"created\_at": "2019-06-07T19:42:12+08:00",

"updated\_at": "2019-06-07T19:42:12+08:00",

"deleted\_at": null,

"name": "user2",

"role": "user",

"department": 1,

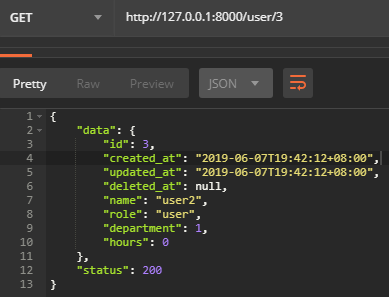
"hours": 0

},

"status": 200

}

##### 输出结果：



#### 011：部门主管获取本部门员工信息

##### 输入：

登录部门经理user1

URL：

GET http://127.0.0.1:8000/user/3

##### 预期输出：

{

"data": {

"id": 3,

"created\_at": "2019-06-07T19:42:12+08:00",

"updated\_at": "2019-06-07T19:42:12+08:00",

"deleted\_at": null,

"name": "user2",

"role": "user",

"department": 1,

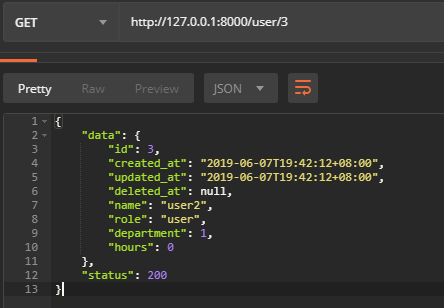
"hours": 0

},

"status": 200

}

##### 测试结果：



#### 012：部门主管获取其他部门员工信息

##### 输入：

登录部门经理user1

URL：

GET http://127.0.0.1:8000/user/5

##### 预期输出：

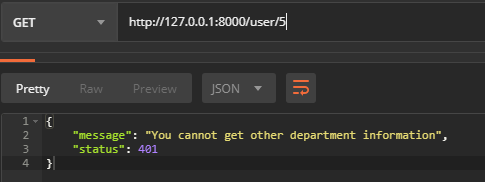
{

"message": "You cannot get other department information",

"status": 401

}

##### 测试结果：



#### 013：经理获取任何人信息

##### 输入：

登录经理root

URL：GET http://127.0.0.1:8000/user/6

##### 预期输出：

{

"data": {

"id": 6,

"created\_at": "2019-06-07T19:43:58+08:00",

"updated\_at": "2019-06-07T19:43:58+08:00",

"deleted\_at": null,

"name": "user5",

"role": "user",

"department": 2,

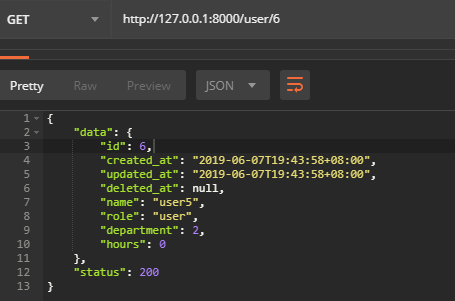
"hours": 0

},

"status": 200

}

##### 测试结果：



### 功能点3：用户列表

#### 014：用户查询用户列表

##### 输入：

登录用户user2

URL：GET http://127.0.0.1:8000/user

##### 预期输出：

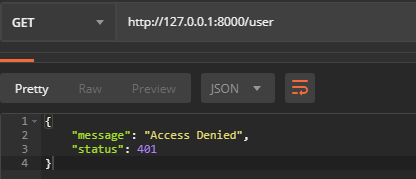
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 015：部门主管查看用户列表

##### 输入：

登录部门主管user1

URL：GET http://127.0.0.1:8000/user

##### 预期输出：

data中的数据为部门主管所在部门用户

{

"current\_page": 1,

"data": [

{

"id": 2,

"created\_at": "2019-06-07T19:35:51+08:00",

"updated\_at": "2019-06-07T19:35:51+08:00",

"deleted\_at": null,

"name": "user1",

"role": "manager",

"department": 1,

"hours": 0

},

......

],

"per\_page": 20,

"status": 200,

"total": 3

}

##### 测试结果：



#### 016：经理查看用户列表：

##### 输入：

登录经理root

URL：GET http://127.0.0.1:8000/user

##### 预期输出：

输出所有的用户

{

"current\_page": 1,

"data": [

{

"id": 1,

"created\_at": "2019-06-07T19:09:37+08:00",

"updated\_at": "2019-06-07T19:09:37+08:00",

"deleted\_at": null,

"name": "root",

"role": "master",

"department": 0,

"hours": 0

},

{

"id": 2,

"created\_at": "2019-06-07T19:35:51+08:00",

"updated\_at": "2019-06-07T19:35:51+08:00",

"deleted\_at": null,

"name": "user1",

"role": "manager",

"department": 1,

"hours": 0

},

......

],

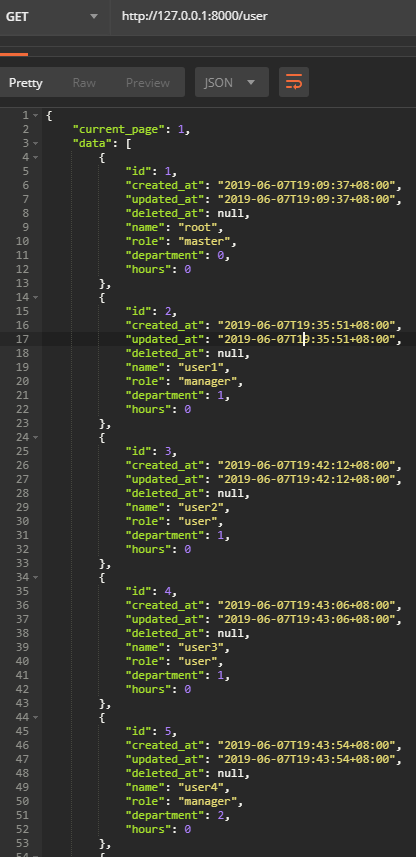
"per\_page": 20,

"status": 200,

"total": 7

}

##### 测试结果：



### 功能点4：修改用户

#### 017：非经理用户修改用户

##### 输入：

登录用户user2

URL：POST <http://127.0.0.1:8000/user/7>

JSON：

{

"name": "user6",

"department": 2,

"role": "user",

"hours": 0

}

##### 预期输出：

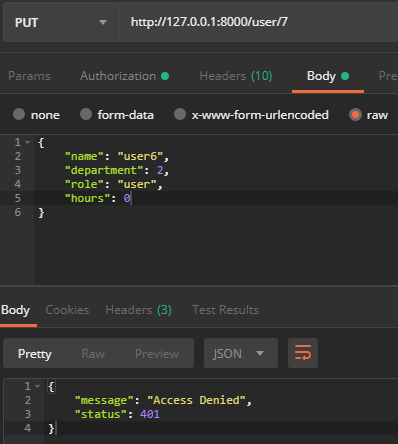
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 018：经理修改用户，用户名过短

##### 输入：

登录经理root

URL：POST http://127.0.0.1:8000/user/7

JSON：

{

"name": "u",

"department": 2,

"role": "user",

"hours": 1

}

##### 预期输出：

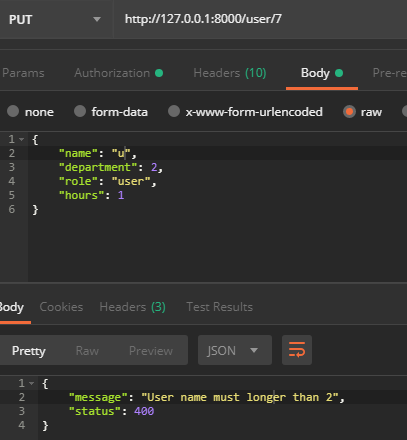
{

"message": "User name must longer than 2",

"status": 400

}

##### 测试结果：



#### 019：经理修改用户，部门不存在

##### 输入：

登录经理root

URL：POST http://127.0.0.1:8000/user/7

JSON：

{

"name": "user6",

"department": 3,

"role": "user",

"hours": 1

}

##### 预期输出：

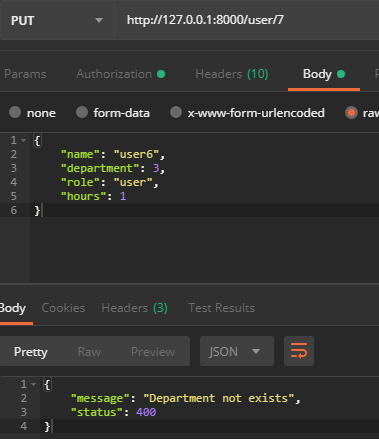
{

"message": "Department not exists",

"status": 400

}

##### 测试结果：



#### 020：经理修改用户，输入角色错误

##### 输入：

登录经理root

URL：POST http://127.0.0.1:8000/user/7

JSON：

{

"name": "user6",

"department": 2,

"role": "test",

"hours": 1

}

##### 预期输出：

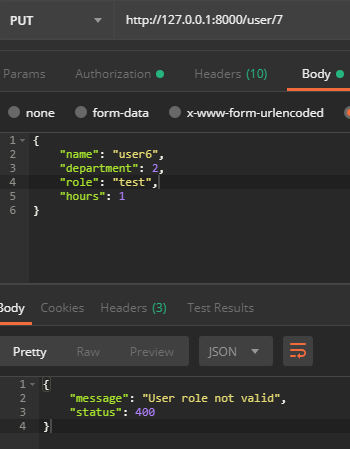
{

"message": "User role not valid",

"status": 400

}

##### 测试结果：



#### 021：经理修改用户，设置部门主管冲突（所在部门已经存在主管）

##### 输入：

登录经理root

（user4已经被设置为manager了）

URL：POST http://127.0.0.1:8000/user/7

JSON：

{

"name": "user6",

"department": 2,

"role": "manager",

"hours": 1

}

##### 预期输出：

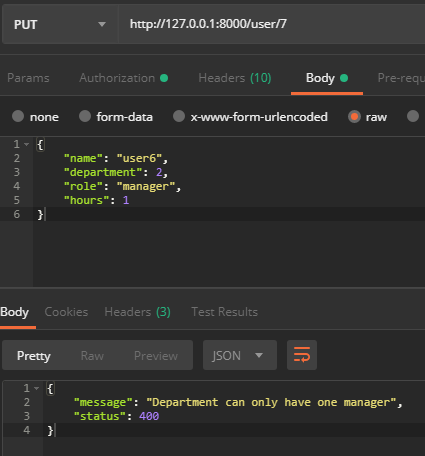
{

"message": "Department can only have one manager",

"status": 400

}

##### 测试结果：



#### 022：经理修改用户，输入工时小于0

##### 输入：

登录经理root

URL：POST http://127.0.0.1:8000/user/7

JSON：

{

"name": "user6",

"department": 2,

"role": "user",

"hours": -1

}

##### 预期输出：

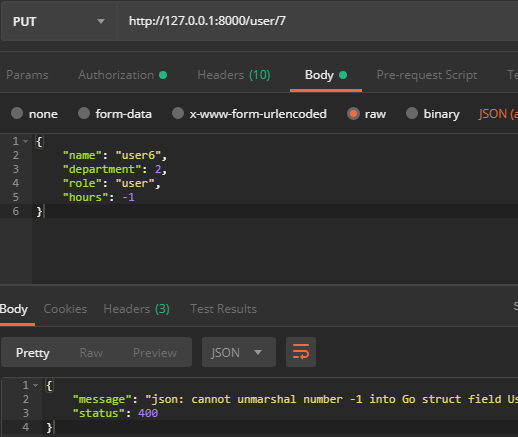
{

"message": "json: cannot unmarshal number -1 into Go struct field UserUpdateRequest.Hours of type uint",

"status": 400

}

##### 测试结果：



#### 023：经理成功修改用户资料

##### 输入：

登录经理root

URL：POST http://127.0.0.1:8000/user/7

JSON：

{

"name": "user6",

"department": 2,

"role": "user",

"hours": 1

}

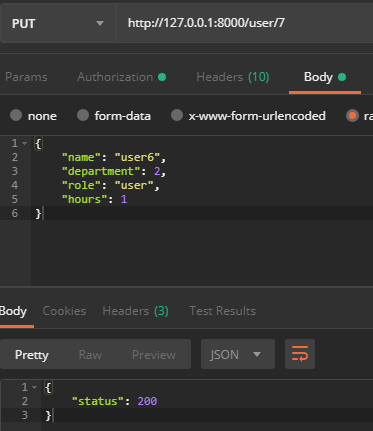
##### 预期输出：

{

"status": 200

}

##### 测试结果：



### 功能点5：删除用户

#### 024：用户删除用户

##### 输入：

登录用户user2

URL：DELET http://127.0.0.1:8000/user/7

##### 预期输出：

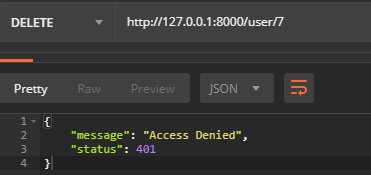
{

"message": "Access Denied",

"status": 401

}

##### 测试结果



#### 025：部门经理删除用户

##### 输入：

登录部门主管user1

URL：DELET http://127.0.0.1:8000/user/2

##### 预期输出：

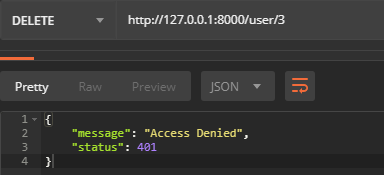
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 027：经理删除用户

##### 输入：

登录经理root

URL：DELET http://127.0.0.1:8000/user/7

##### 预期输出：

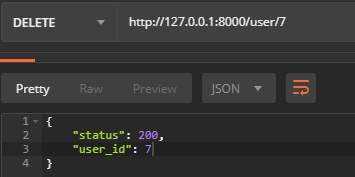
{

"status": 200,

"user\_id": 7

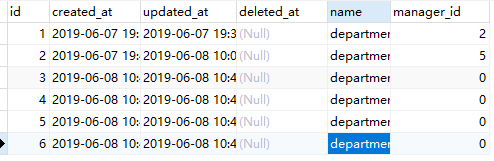
}

##### 测试结果：



## 部门相关

参考数据库：



### 功能点6：添加部门

#### 028：用户添加部门

##### 输入：

登录用户user2

URL：PUT <http://127.0.0.1:8000/department>

JSON：

{

"name": "department3"

}

##### 预期输出：

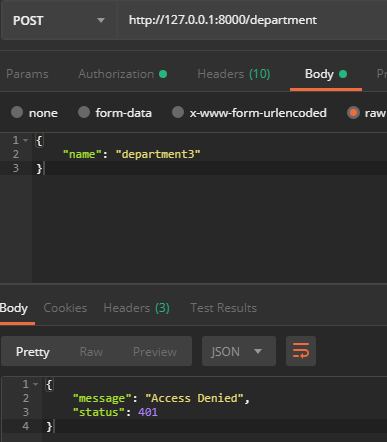
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 029：部门主管添加部门

##### 输入：

登录部门主管user1

URL：PUT <http://127.0.0.1:8000/department>

JSON：

{

"name": "department3"

}

##### 预期输出：

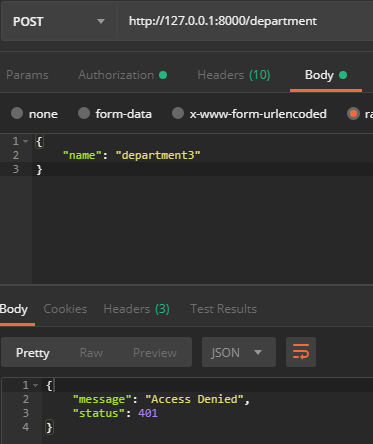
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 030：经理添加部门名字过短

##### 输入：

登录经理root

URL：POST：http://127.0.0.1:8000/department

JSON：

{

"name": "d"

}

##### 预期输入

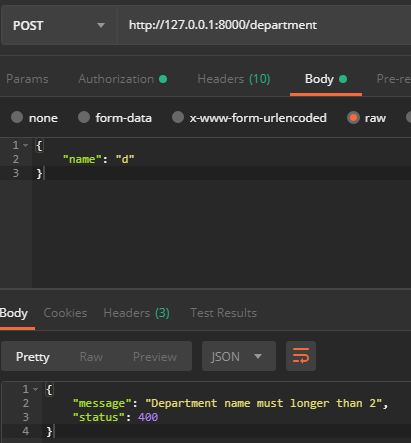
{

"message": "Department name must longer than 2",

"status": 400

}

##### 测试结果：



#### 032：经理添加部门名字重复

##### 输入：

登录经理root

URL：POST：http://127.0.0.1:8000/department

JSON：

{

"name": "department1"

}

##### 预期输出：

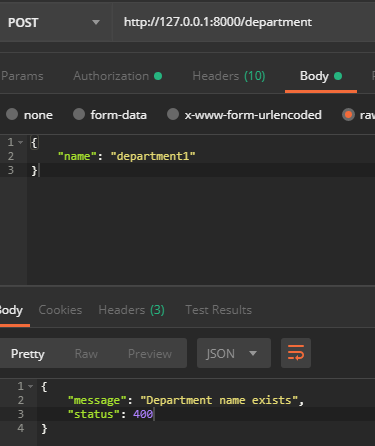
{

"message": "Department name exists",

"status": 400

}

##### 测试结果：



#### 033：经理成功添加部门

##### 输入：

登录经理root

URL：POST：http://127.0.0.1:8000/department

JSON：

{

"name": "department3"

}

##### 预期输出：

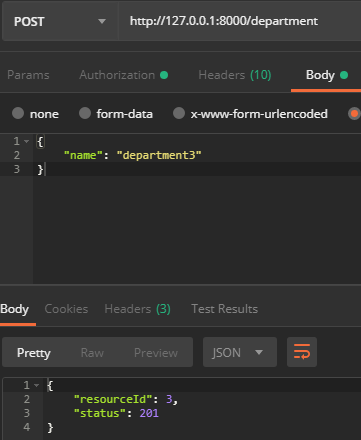
{

"resourceId": 3,

"status": 201

}

##### 测试结果：



### 功能点7：修改部门

#### 034：用户修改部门

##### 输入：

登录用户user2

URL：PUT <http://127.0.0.1:8000/department/3>

JSON：

{

"name": "department3"

}

##### 预期输出：

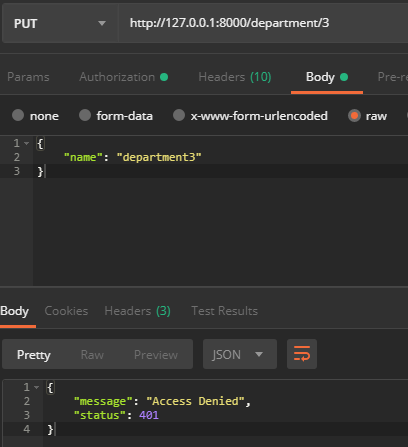
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 035：部门主管修改部门

##### 输入：

登录部门主管user1

URL：PUT <http://127.0.0.1:8000/department/3>

JSON：

{

"name": "department3"

}

##### 预期输出：

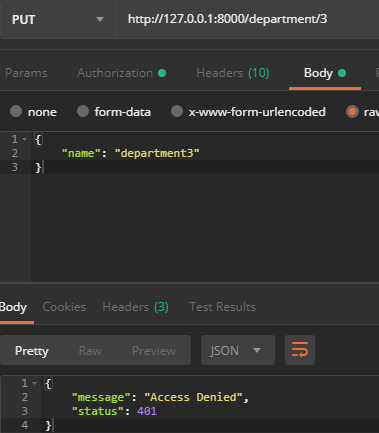
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 036：经理修改部门，名字过短

##### 输入：

登录经理

URL：PUT http://127.0.0.1:8000/department/6

JSON：

{

"name": "d"

}

##### 预期输出：

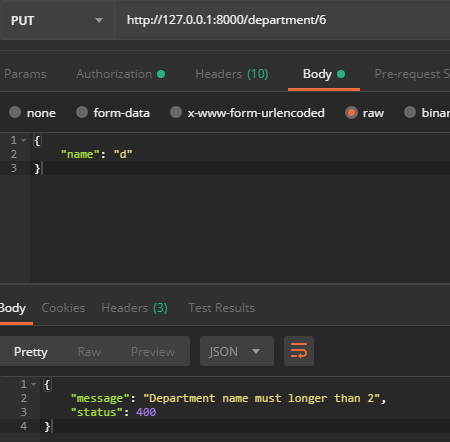
{

"message": "Department name must longer than 2",

"status": 400

}

##### 测试结果：



#### 037：经理修改部门，名字已存在

##### 输入：

登录经理

URL：PUT http://127.0.0.1:8000/department/6

JSON：

{

"name": "department6"

}

##### 预期输出：

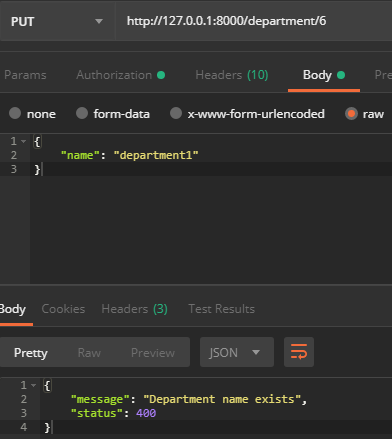
{

"message": "Department name exists",

"status": 400

}

##### 测试结果：



#### 038：经理成功修改部门

##### 输入：

URL：PUT http://127.0.0.1:8000/department/6

JSON：

{

"name": "dp6"

}

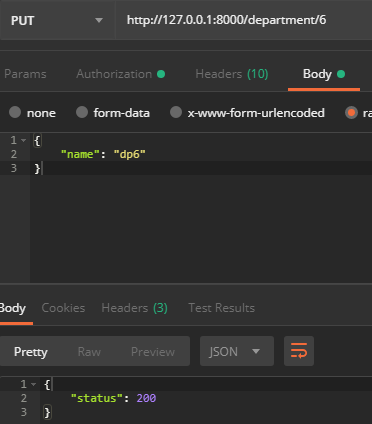
##### 预期输出：

{

"status": 200

}

##### 测试结果：



### 功能点8：获取指定部门信息

#### 039：用户获取其他部门信息

##### 输入：

登录用户user2

URL：GET http://127.0.0.1:8000/department/2

##### 预期输出：

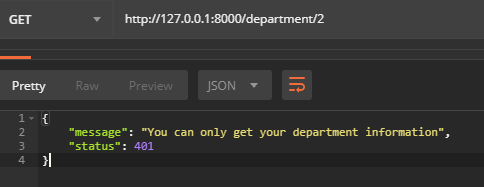
{

"message": "You can only get your department information",

"status": 401

}

##### 测试结果：



#### 040：用户获取本部门信息

##### 输入：

登录用户user2

URL：GET http://127.0.0.1:8000/department/1

##### 预期输出：

{

"data": {

"id": 1,

"created\_at": "2019-06-07T19:35:19+08:00",

"updated\_at": "2019-06-07T19:35:23+08:00",

"deleted\_at": null,

"name": "department1",

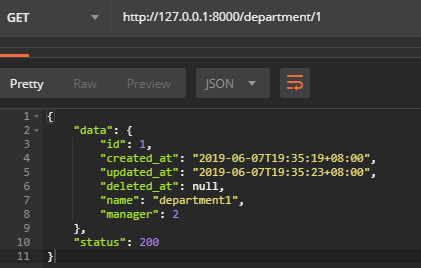
"manager": 2

},

"status": 200

}

##### 测试结果：



#### 041：部门主管获取其他部门信息

##### 输入：

登录部门主管user2

URL：GET http://127.0.0.1:8000/department/2

##### 预期输出：

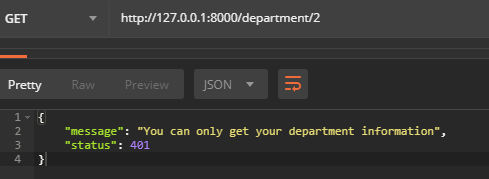
{

"message": "You can only get your department information",

"status": 401

}

##### 测试结果：



#### 042：部门主管获取本部门信息

##### 输入：

登录部门主管user2

URL：GET http://127.0.0.1:8000/department/1

##### 预期输出：

{

"data": {

"id": 1,

"created\_at": "2019-06-07T19:35:19+08:00",

"updated\_at": "2019-06-07T19:35:23+08:00",

"deleted\_at": null,

"name": "department1",

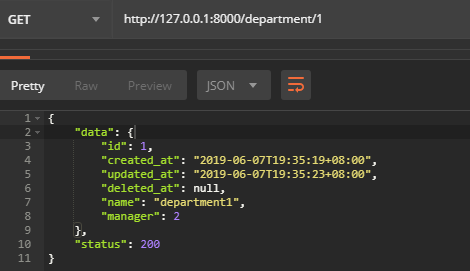
"manager": 2

},

"status": 200

}

##### 测试结果：



#### 043：经理获取部门不存在

##### 输入：

登录经理root

URL：GET http://127.0.0.1:8000/department/10

##### 预期输出：

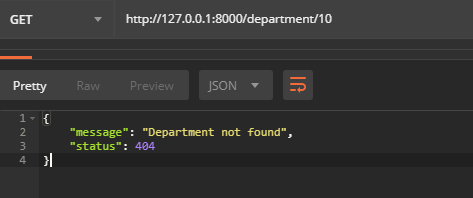
{

"message": "Department not found",

"status": 404

}

##### 测试结果：



#### 044：经理成功获取部门信息

##### 输入：

登录经理root

URL：GET http://127.0.0.1:8000/department/2

##### 预期输出：

{

"data": {

"id": 2,

"created\_at": "2019-06-07T19:43:14+08:00",

"updated\_at": "2019-06-08T10:05:44+08:00",

"deleted\_at": null,

"name": "department2",

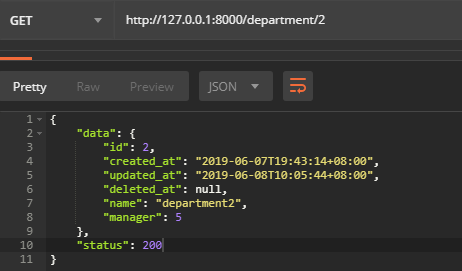
"manager": 5

},

"status": 200

}

##### 测试结果：



### 功能点9：获取部门列表

#### 045：用户获取部门列表

##### 输入：

登录用户user2

URL：GET http://127.0.0.1:8000/department

##### 预期输出：

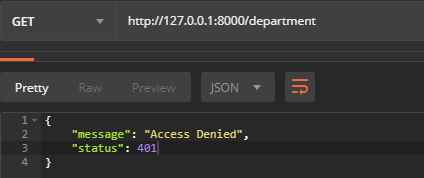
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 046：部门主管获取部门列表

##### 输入：

登录部门主管user1

URL：GET http://127.0.0.1:8000/department

##### 预期输出：

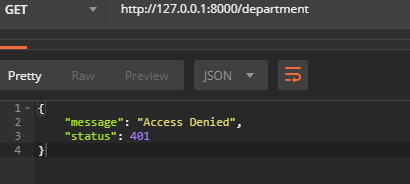
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 047：经理获取部门列表

##### 输入：

登录经理root

URL：GET http://127.0.0.1:8000/department

##### 预期输出：

data中为所有部门信息

{

"current\_page": 1,

"data": [

{

"id": 1,

"created\_at": "2019-06-07T19:35:19+08:00",

"updated\_at": "2019-06-07T19:35:23+08:00",

"deleted\_at": null,

"name": "department1",

"manager": 2

},

......

],

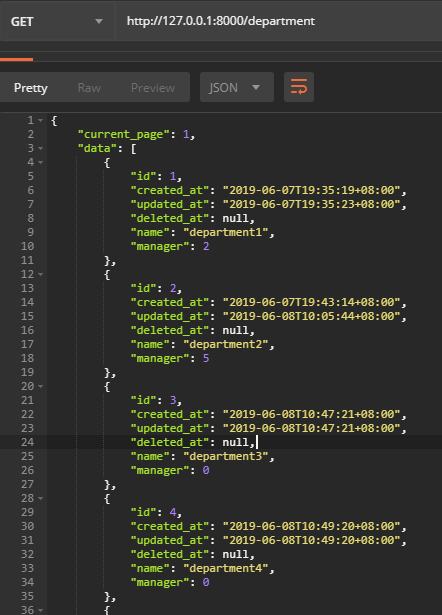
"per\_page": 20,

"status": 200,

"total": 6

}

##### 测试结果：



### 功能点10：删除部门

#### 048：用户删除部门

##### 输入：

登录用户user2:

URL：DELET http://127.0.0.1:8000/department/6

##### 预期输出：

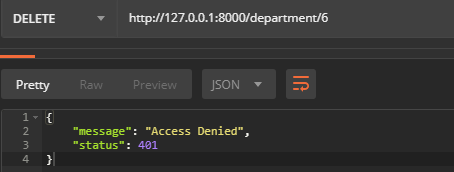
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 049：部门主管删除部门

##### 输入：

登录部门主管user1:

URL：DELET http://127.0.0.1:8000/department/1

##### 预期输出：

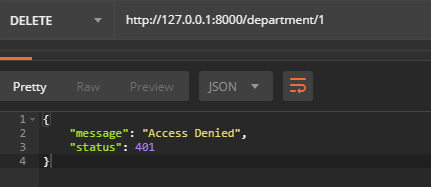
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 050：经理删除部门

##### 输入：

登录经理root:

URL：DELET http://127.0.0.1:8000/department/6

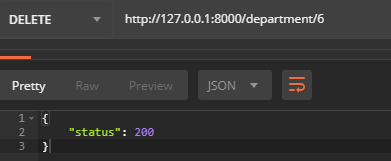
##### 预期输出：

{

"status": 200

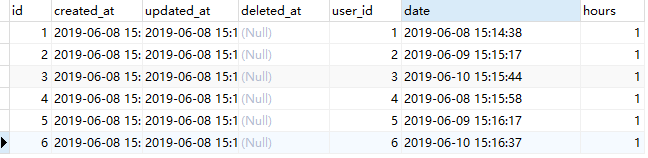
}

##### 测试结果：



## 工时相关

工时数据库参考：



### 功能点11：获取工时记录

#### 051：用户查询他人工时记录

##### 输入：

登录用户user2,查询user1的工时

URL：GET http://127.0.0.1:8000/hours?user\_id=2

##### 预期输出：

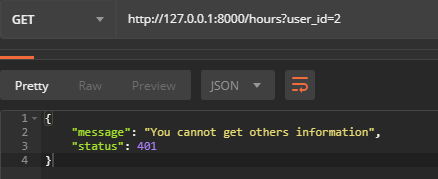
{

"message": "You cannot get others information",

"status": 401

}

##### 测试结果：



#### 052：用户查询自己的工时记录

##### 输入：

登录用户user2,查询自己的工时

URL：GET http://127.0.0.1:8000/hours?user\_id=3

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"id": 3,

"user\_id": 3,

"user\_name": "",

"date": "2019-06-10T15:15:44+08:00",

"hours": 1

}

],

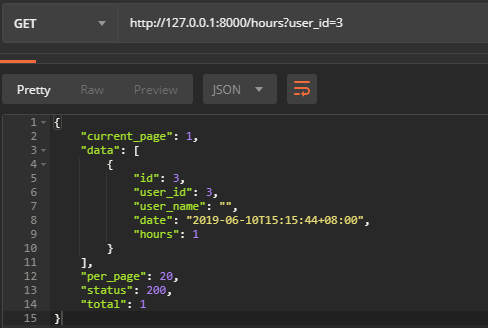
"per\_page": 20,

"status": 200,

"total": 1

}

##### 测试结果：



#### 053：部门主管查询部门内员工工时记录

##### 输入：

登录部门主管user1查询用户user2的工时

URL：GET http://127.0.0.1:8000/hours?user\_id=3

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"id": 3,

"user\_id": 3,

"user\_name": "",

"date": "2019-06-10T15:15:44+08:00",

"hours": 1

}

],

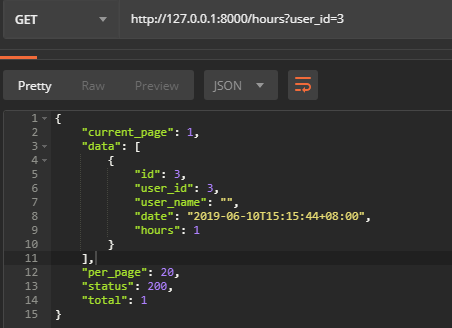
"per\_page": 20,

"status": 200,

"total": 1

}

##### 测试结果：



#### 055：部门主管查询自己的工时记录

##### 输入：

登录部门主管user1查询自己的工时

URL：GET http://127.0.0.1:8000/hours?user\_id=2

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"id": 2,

"user\_id": 2,

"user\_name": "",

"date": "2019-06-09T15:15:17+08:00",

"hours": 1

}

],

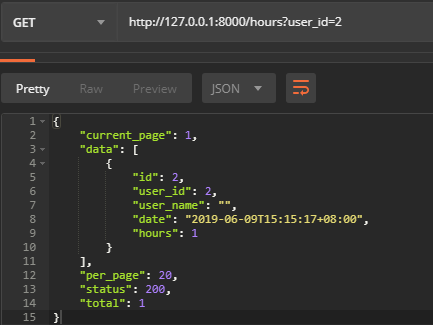
"per\_page": 20,

"status": 200,

"total": 1

}

##### 测试结果：



#### 056：部门主管查询部门内员工工时记录

##### 输入：

登录部门主管user1

URL：GET http://127.0.0.1:8000/hours

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"id": 2,

"user\_id": 2,

"user\_name": "",

"date": "2019-06-09T15:15:17+08:00",

"hours": 1

},

{

"id": 3,

"user\_id": 3,

"user\_name": "",

"date": "2019-06-10T15:15:44+08:00",

"hours": 1

},

{

"id": 4,

"user\_id": 4,

"user\_name": "",

"date": "2019-06-08T15:15:58+08:00",

"hours": 1

}

],

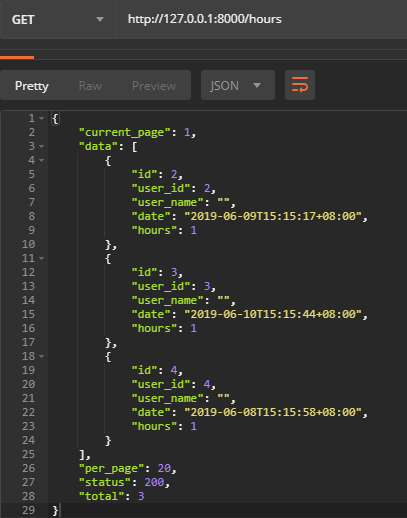
"per\_page": 20,

"status": 200,

"total": 3

}

##### 测试结果：



#### 053：经理查询工时记录

##### 输入：

登录经理root

URL：GET http://127.0.0.1:8000/hours

##### 预期输出：

会输出所有的工时记录

{

"current\_page": 1,

"data": [

{

"id": 1,

"user\_id": 1,

"user\_name": "",

"date": "2019-06-08T15:14:38+08:00",

"hours": 1

},

......

],

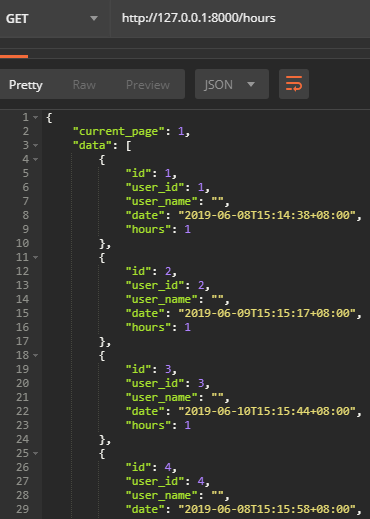
"per\_page": 20,

"status": 200,

"total": 6

}

##### 测试结果：



#### 054：经理限制开始和结束时间查询工时

##### 输入：

登录经理root

URL：GET http://127.0.0.1:8000/hours?start\_at=2019-06-08 00:00:00&end\_at=2019-06-10 00:00:00

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"id": 1,

"user\_id": 1,

"user\_name": "",

"date": "2019-06-08T15:14:38+08:00",

"hours": 1

},

{

"id": 2,

"user\_id": 2,

"user\_name": "",

"date": "2019-06-09T15:15:17+08:00",

"hours": 1

},

{

"id": 4,

"user\_id": 4,

"user\_name": "",

"date": "2019-06-08T15:15:58+08:00",

"hours": 1

},

{

"id": 5,

"user\_id": 5,

"user\_name": "",

"date": "2019-06-09T15:16:17+08:00",

"hours": 1

}

],

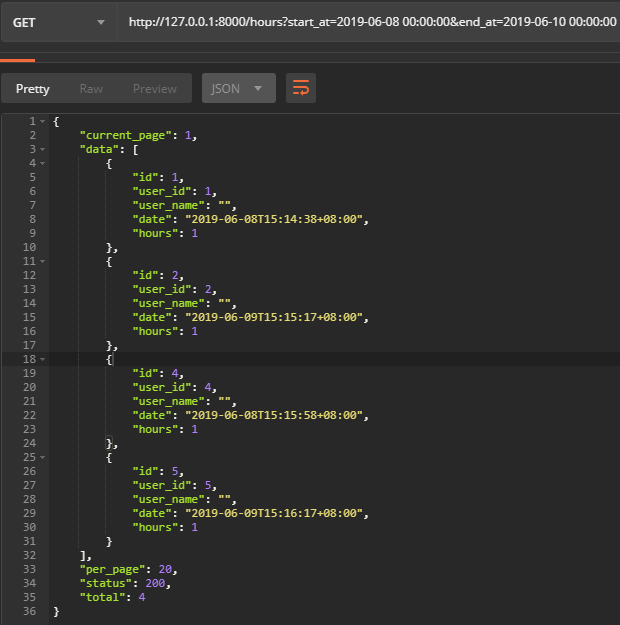
"per\_page": 20,

"status": 200,

"total": 4

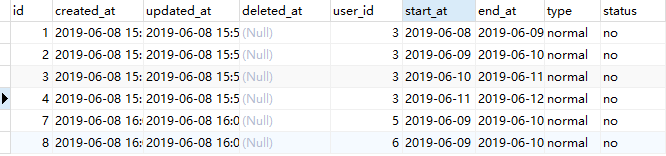
}

##### 测试结果：



## 排班相关

参考数据库：



### 功能点12：添加排班

#### 051：用户添加排班

##### 输入：

登录用户user2

URL：POST <http://127.0.0.1:8000/shift/user/3>

JSON：

{

"start\_at": "2019-06-10 09:00:00"，

"end\_at": "2019-06-10 17:00:00"，

"type": "normal"

}

##### 预期输出：

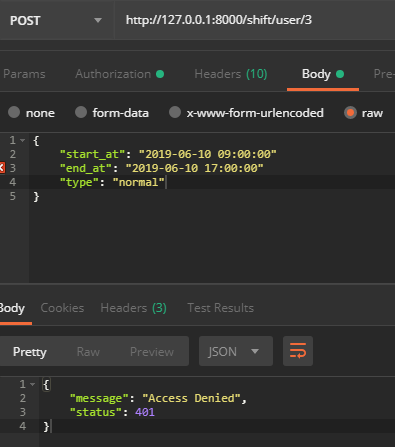
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 052：部门主管添加排班，输入时间错误

##### 输入：

登录部门主管user1

URL：POST <http://127.0.0.1:8000/shift/user/3>

JSON：

{

"start\_at": "2019-06-10 09:00:00"，

"end\_at": "2019-06-09 17:00:00"，

"type": "normal"

}

##### 预期输出：

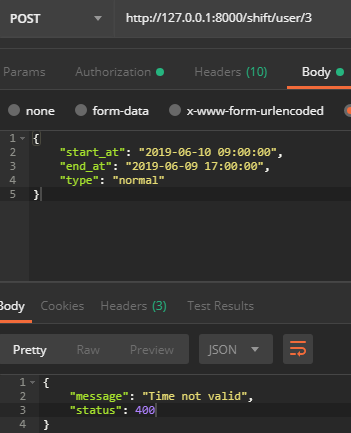
{

"message": "Time not valid",

"status": 400

}

##### 测试结果：



#### 053：部门主管添加排班，和员工排班冲突

##### 输入：

登录部门主管user1

URL：POST <http://127.0.0.1:8000/shift/user/3>

JSON：

{

"start\_at": "2019-06-09 09:00:00",

"end\_at": "2019-06-09 17:00:00",

"type": "normal"

}

##### 预期输出：

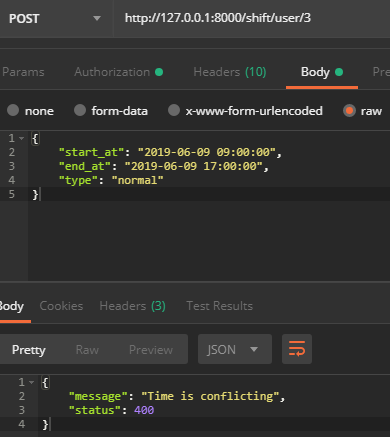
{

"message": "Time is conflicting",

"status": 400

}

##### 测试结果：



#### 054：部门主管添加排班，正确添加

##### 输入：

登录部门主管user1

URL：POST <http://127.0.0.1:8000/shift/user/3>

JSON：

{

"start\_at": "2019-06-11 09:00:00",

"end\_at": "2019-06-11 17:00:00",

"type": "normal"

}

##### 预期输出：

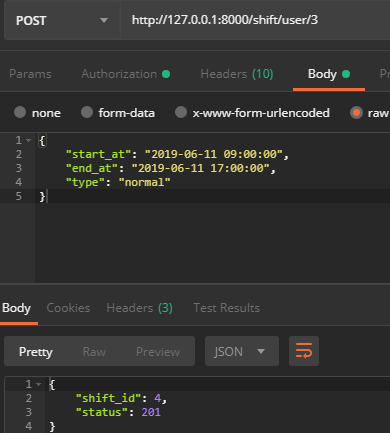
{

"shift\_id": 4,

"status": 201

}

##### 测试结果：



### 功能点13：部门排班

#### 055：部门主管给本部门排班

##### 输入：

登录部门主管user4

URL：POST http://127.0.0.1:8000/shift/department/2

JSON：

{

"start\_at": "2019-06-07 09:00:00"

"end\_at": "2019-06-07 17:00:00"

"type": "normal"

}

##### 预期输出：

{

"shift\_ids": [

5,

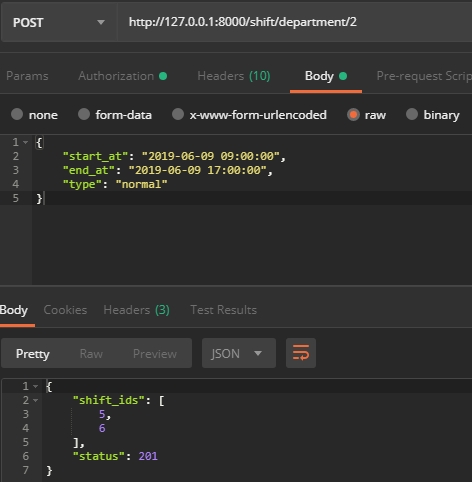
6

],

"status": 201

}

##### 测试结果：



#### 056：部门主管给其他部门排班

##### 输入：

登录部门主管user4

URL：POST http://127.0.0.1:8000/shift/department/1

JSON：

{

"start\_at": "2019-06-07 09:00:00"

"end\_at": "2019-06-07 17:00:00"

"type": "normal"

}

##### 预期输出：

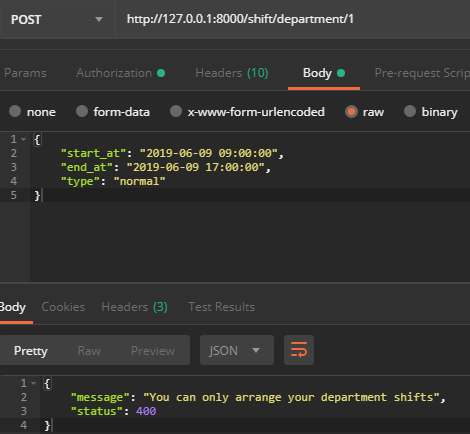
{

"message": "You can only arrange your department shifts",

"status": 400

}

##### 测试结果：



#### 057：用户给全部门排班

##### 输入：

登录用户user2

URL：POST http://127.0.0.1:8000/shift/department/1

JSON：

{

"start\_at": "2019-06-07 09:00:00"

"end\_at": "2019-06-07 17:00:00"

"type": "normal"

}

##### 预期输出：

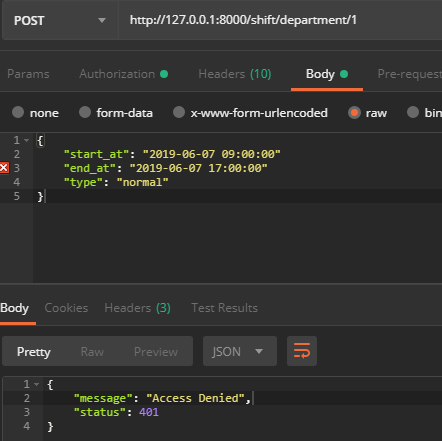
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



### 功能点14：获取排班列表

#### 058：用户查看自己的排班

##### 输入：

登录用户user2，查看自己工时

URL：GET http://127.0.0.1:8000/shift?user\_id=3

##### 预期输出：

所有user2的排班

{

"current\_page": 1,

"data": [

{

"created\_at": "2019-06-08T15:54:34+08:00",

"end\_at": "2019-06-09T01:00:00+08:00",

"id": 1,

"start\_at": "2019-06-08T17:00:00+08:00",

"status": "no",

"type": "normal",

"updated\_at": "2019-06-08T15:54:34+08:00",

"user\_id": 3,

"user\_name": ""

},

......

],

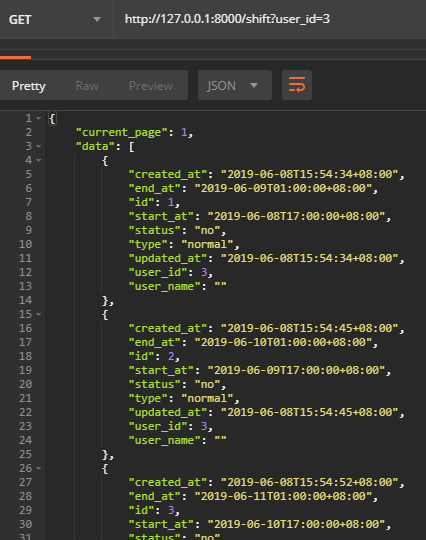
"per\_page": 20,

"status": 200,

"total": 4

}

##### 测试结果：



#### 059：用户查看他人排班

##### 输入：

登录用户user2，查看user3工时

URL：GET http://127.0.0.1:8000/shift?user\_id=4

##### 预期输出：

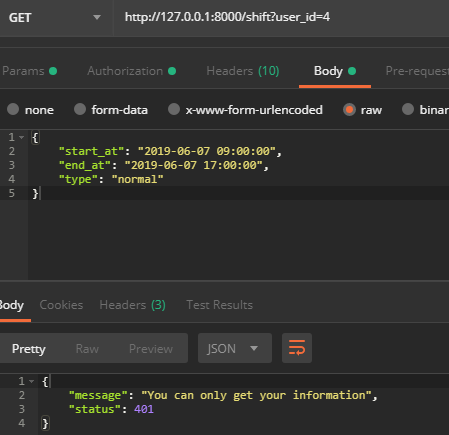
{

"message": "You can only get your information",

"status": 401

}

##### 测试结果：



#### 060：部门主管查看本部门排班

##### 输入：

登录部门主管user4

URL：GET http://127.0.0.1:8000/shift?department\_id=2

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"created\_at": "2019-06-08T16:08:23+08:00",

"end\_at": "2019-06-10T01:00:00+08:00",

"id": 7,

"start\_at": "2019-06-09T17:00:00+08:00",

"status": "no",

"type": "normal",

"updated\_at": "2019-06-08T16:08:23+08:00",

"user\_id": 5,

"user\_name": ""

},

{

"created\_at": "2019-06-08T16:08:23+08:00",

"end\_at": "2019-06-10T01:00:00+08:00",

"id": 8,

"start\_at": "2019-06-09T17:00:00+08:00",

"status": "no",

"type": "normal",

"updated\_at": "2019-06-08T16:08:23+08:00",

"user\_id": 6,

"user\_name": ""

}

],

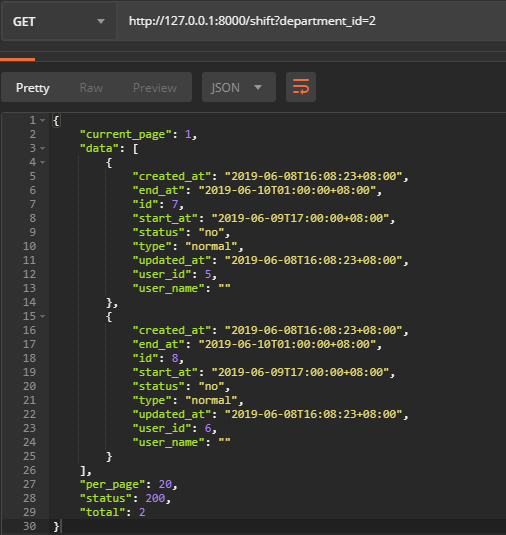
"per\_page": 20,

"status": 200,

"total": 2

}

##### 测试结果：



#### 061：部门主管查看其它部门排班

##### 输入：

登录部门主管user4

URL：GET http://127.0.0.1:8000/shift?department\_id=1

##### 预期输出：

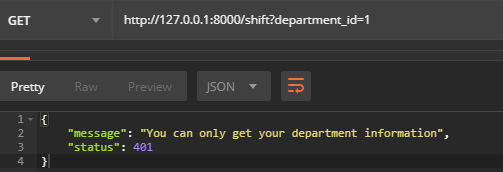
{

"message": "You can only get your department information",

"status": 401

}

##### 测试结果：



#### 062：经理查看所有排班

##### 输入：

登录经理root

URL：GET http://127.0.0.1:8000/shift

##### 预期输出：

可以输出所有排班情况

{

"current\_page": 1,

"data": [

{

"created\_at": "2019-06-08T15:54:34+08:00",

"end\_at": "2019-06-09T01:00:00+08:00",

"id": 1,

"start\_at": "2019-06-08T17:00:00+08:00",

"status": "no",

"type": "normal",

"updated\_at": "2019-06-08T15:54:34+08:00",

"user\_id": 3,

"user\_name": ""

},

......

],

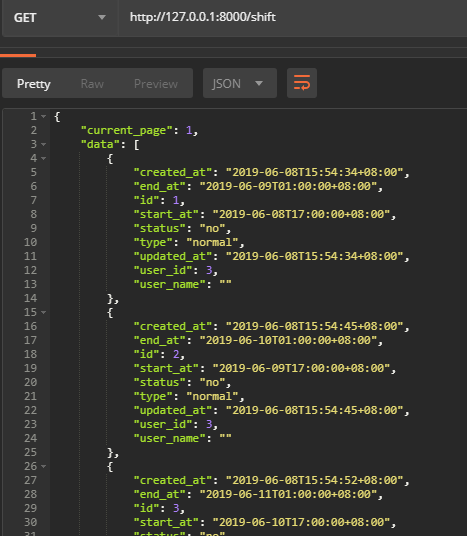
"per\_page": 20,

"status": 200,

"total": 6

}

##### 测试结果：



#### 063：经理筛选时间查看排班

##### 输入：

登录经理root

URL：GET http://127.0.0.1:8000/shift?start\_at=2019-06-09 00:00:00&end\_at=2019-06-11 00:00:00

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"created\_at": "2019-06-08T15:54:45+08:00",

"end\_at": "2019-06-10T01:00:00+08:00",

"id": 2,

"start\_at": "2019-06-09T17:00:00+08:00",

"status": "no",

"type": "normal",

"updated\_at": "2019-06-08T15:54:45+08:00",

"user\_id": 3,

"user\_name": ""

},

{

"created\_at": "2019-06-08T16:08:23+08:00",

"end\_at": "2019-06-10T01:00:00+08:00",

"id": 7,

"start\_at": "2019-06-09T17:00:00+08:00",

"status": "no",

"type": "normal",

"updated\_at": "2019-06-08T16:08:23+08:00",

"user\_id": 5,

"user\_name": ""

},

{

"created\_at": "2019-06-08T16:08:23+08:00",

"end\_at": "2019-06-10T01:00:00+08:00",

"id": 8,

"start\_at": "2019-06-09T17:00:00+08:00",

"status": "no",

"type": "normal",

"updated\_at": "2019-06-08T16:08:23+08:00",

"user\_id": 6,

"user\_name": ""

}

],

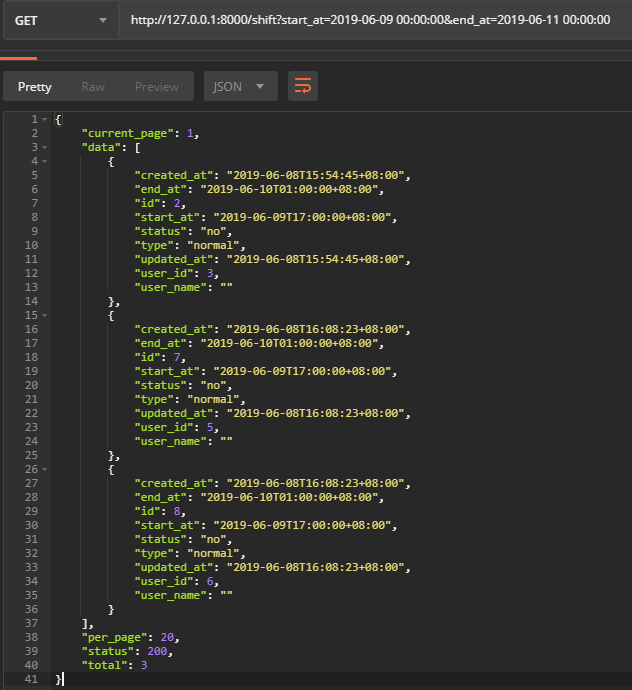
"per\_page": 20,

"status": 200,

"total": 3

}

##### 测试结果：



### 功能点15：删除排班

#### 064：用户删除排班

##### 输入：

登录用户user2

URL：DELET <http://127.0.0.1:8000/shift/4>

##### 预期输出：

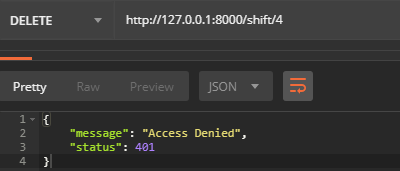
{

"message": "Access Denied",

"status": 401

}

##### 测试结果



#### 065：部门主管删除本部门排班

##### 输入：

登录用户user1

URL：DELET <http://127.0.0.1:8000/shift/4>

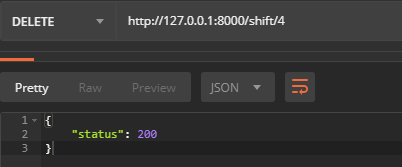
##### 预期输出：

{

"status": 200

}

##### 测试结果



#### 066：部门主管删除其他部门排班

##### 输入：

登录部门主管user1

URL：DELET [http://127.0.0.1:8000/shift/](http://127.0.0.1:8000/shift/4)8

##### 预期输出：

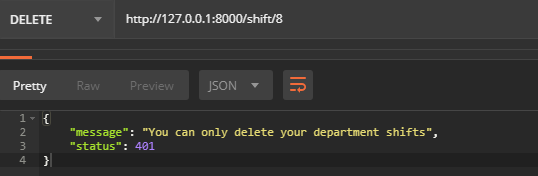
{

"message": "You can only delete your department shifts",

"status": 401

}

##### 测试结果



#### 067：经理删除排班

##### 输入：

登录经理root

URL：DELET [http://127.0.0.1:8000/shift/](http://127.0.0.1:8000/shift/4)8

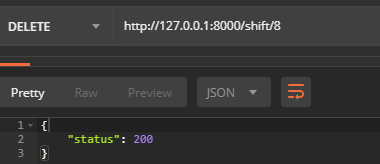
##### 预期输出：

{

"status": 200

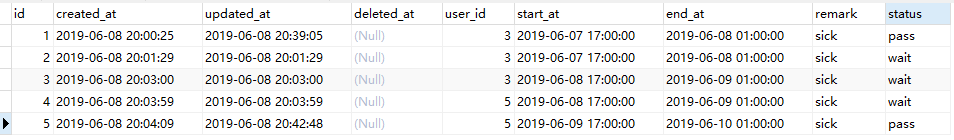
}

##### 测试结果：



## 请假相关

数据库参考：



### 功能点16：申请请假

#### 068：填写开始和结束时间错误

##### 输入：

登录用户user2

URL：POST http://127.0.0.1:8000/leave/user/1

JSON：

{

"start\_at": "2019-06-07 09:00:00",

"end\_at": "2019-06-06 17:00:00",

"remark": "sick"

}

##### 预期输出：

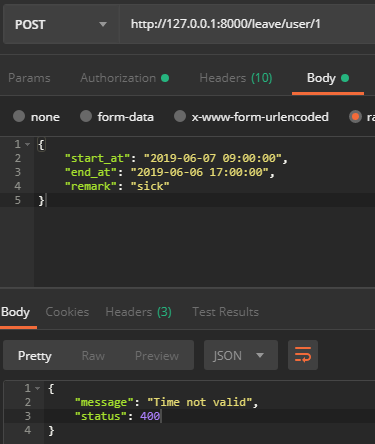
{

"message": "Time not valid",

"status": 400

}

##### 测试



#### 069：用户正确提交请假

##### 输入：

登录用户user2

URL：POST http://127.0.0.1:8000/leave/user/3

JSON：

{

"start\_at": "2019-06-07 09:00:00",

"end\_at": "2019-06-07 17:00:00",

"remark": "sick"

}

##### 预期输出：

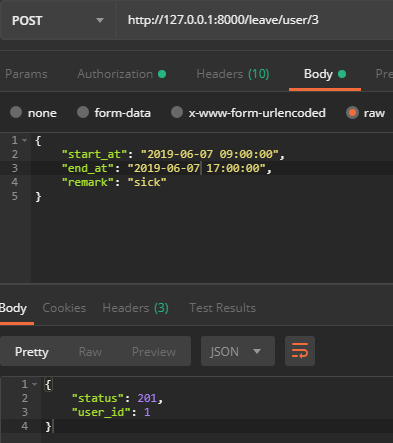
{

"status": 201,

"user\_id": 1

}

##### 测试结果：



#### 070：用户给别人请假

##### 输入：

登录用户user2

URL：POST http://127.0.0.1:8000/leave/user/4

JSON：

{

"start\_at": "2019-06-07 09:00:00",

"end\_at": "2019-06-07 17:00:00",

"remark": "sick"

}

##### 预期输出：

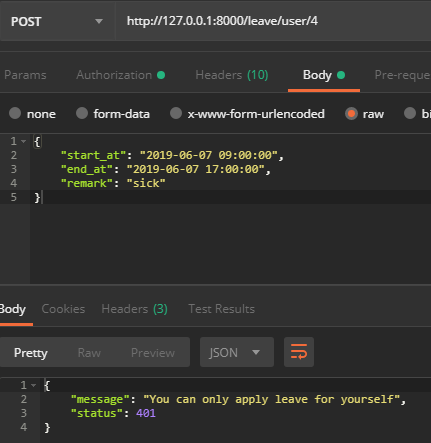
{

"message": "You can only apply leave for yourself",

"status": 401

}

##### 测试结果：



### 功能点17：获取指定用户请假

#### 071：用户获取别人请假

##### 输入：

登录用户user2

URL：GET [http://127.0.0.1:8000/leave/user/](http://127.0.0.1:8000/leave/user/3)2

##### 预期输出：

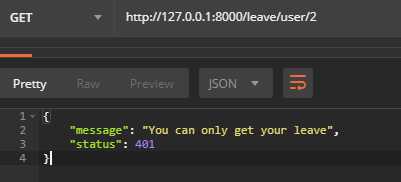
{

"message": "You can only get your leave",

"status": 401

}

##### 测试结果：



#### 072：用户获取自己请假

##### 输入：

登录用户user2

URL：GET <http://127.0.0.1:8000/leave/user/3>

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"id": 1,

"created\_at": "2019-06-08T20:00:25+08:00",

"updated\_at": "2019-06-08T20:00:25+08:00",

"deleted\_at": null,

"user\_id": 3,

"start\_at": "2019-06-07T17:00:00+08:00",

"end\_at": "2019-06-08T01:00:00+08:00",

"remark": "sick",

"status": "wait"

},

{

"id": 2,

"created\_at": "2019-06-08T20:01:29+08:00",

"updated\_at": "2019-06-08T20:01:29+08:00",

"deleted\_at": null,

"user\_id": 3,

"start\_at": "2019-06-07T17:00:00+08:00",

"end\_at": "2019-06-08T01:00:00+08:00",

"remark": "sick",

"status": "wait"

},

{

"id": 3,

"created\_at": "2019-06-08T20:03:00+08:00",

"updated\_at": "2019-06-08T20:03:00+08:00",

"deleted\_at": null,

"user\_id": 3,

"start\_at": "2019-06-08T17:00:00+08:00",

"end\_at": "2019-06-09T01:00:00+08:00",

"remark": "sick",

"status": "wait"

}

],

"per\_page": 20,

"status": 200,

"total": 3

}

##### 测试结果：



#### 073：部门主管获取部门员工请假

##### 输入：

登录部门主管user1

URL：GET <http://127.0.0.1:8000/leave/user/3>

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"id": 1,

"created\_at": "2019-06-08T20:00:25+08:00",

"updated\_at": "2019-06-08T20:00:25+08:00",

"deleted\_at": null,

"user\_id": 3,

"start\_at": "2019-06-07T17:00:00+08:00",

"end\_at": "2019-06-08T01:00:00+08:00",

"remark": "sick",

"status": "wait"

},

{

"id": 2,

"created\_at": "2019-06-08T20:01:29+08:00",

"updated\_at": "2019-06-08T20:01:29+08:00",

"deleted\_at": null,

"user\_id": 3,

"start\_at": "2019-06-07T17:00:00+08:00",

"end\_at": "2019-06-08T01:00:00+08:00",

"remark": "sick",

"status": "wait"

},

{

"id": 3,

"created\_at": "2019-06-08T20:03:00+08:00",

"updated\_at": "2019-06-08T20:03:00+08:00",

"deleted\_at": null,

"user\_id": 3,

"start\_at": "2019-06-08T17:00:00+08:00",

"end\_at": "2019-06-09T01:00:00+08:00",

"remark": "sick",

"status": "wait"

}

],

"per\_page": 20,

"status": 200,

"total": 3

}

##### 测试结果：



#### 074：部门主管获取其他部门员工请假

##### 输入：

登录部门主管user1

URL：GET [http://127.0.0.1:8000/leave/user/](http://127.0.0.1:8000/leave/user/3)5

##### 预期输出：

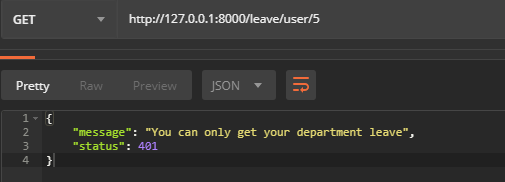
{

"message": "You can only get your department leave",

"status": 401

}

##### 测试结果：



#### 075：经理获取员工请假

##### 输入：

登录经理root

URL：GET [http://127.0.0.1:8000/leave/user/](http://127.0.0.1:8000/leave/user/3)5

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"id": 4,

"created\_at": "2019-06-08T20:03:59+08:00",

"updated\_at": "2019-06-08T20:03:59+08:00",

"deleted\_at": null,

"user\_id": 5,

"start\_at": "2019-06-08T17:00:00+08:00",

"end\_at": "2019-06-09T01:00:00+08:00",

"remark": "sick",

"status": "wait"

},

{

"id": 5,

"created\_at": "2019-06-08T20:04:09+08:00",

"updated\_at": "2019-06-08T20:04:09+08:00",

"deleted\_at": null,

"user\_id": 5,

"start\_at": "2019-06-09T17:00:00+08:00",

"end\_at": "2019-06-10T01:00:00+08:00",

"remark": "sick",

"status": "wait"

}

],

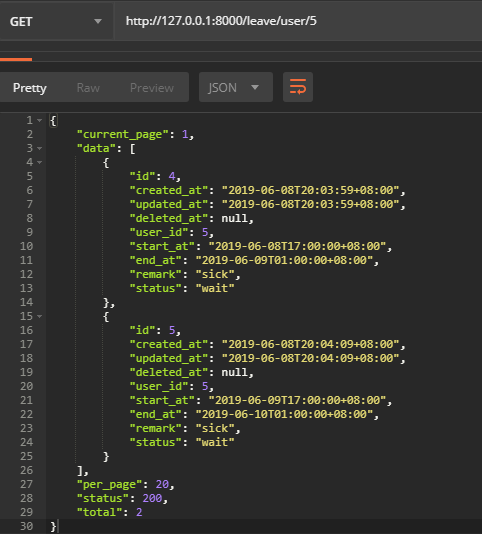
"per\_page": 20,

"status": 200,

"total": 2

}

##### 测试结果：



### 功能点18：获取请假列表

#### 076：用户获取请假列表

##### 输入：

登录用户user2

URL：GET http://127.0.0.1:8000/leave

##### 预期输出：

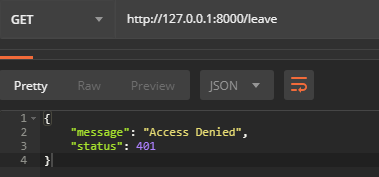
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 077：部门主管获取本部门请假列表

##### 输入：

登录用户user1

URL：GET http://127.0.0.1:8000/leave

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"id": 1,

"created\_at": "2019-06-08T20:00:25+08:00",

"updated\_at": "2019-06-08T20:00:25+08:00",

"deleted\_at": null,

"user\_id": 3,

"start\_at": "2019-06-07T17:00:00+08:00",

"end\_at": "2019-06-08T01:00:00+08:00",

"remark": "sick",

"status": "wait"

},

{

"id": 2,

"created\_at": "2019-06-08T20:01:29+08:00",

"updated\_at": "2019-06-08T20:01:29+08:00",

"deleted\_at": null,

"user\_id": 3,

"start\_at": "2019-06-07T17:00:00+08:00",

"end\_at": "2019-06-08T01:00:00+08:00",

"remark": "sick",

"status": "wait"

},

{

"id": 3,

"created\_at": "2019-06-08T20:03:00+08:00",

"updated\_at": "2019-06-08T20:03:00+08:00",

"deleted\_at": null,

"user\_id": 3,

"start\_at": "2019-06-08T17:00:00+08:00",

"end\_at": "2019-06-09T01:00:00+08:00",

"remark": "sick",

"status": "wait"

}

],

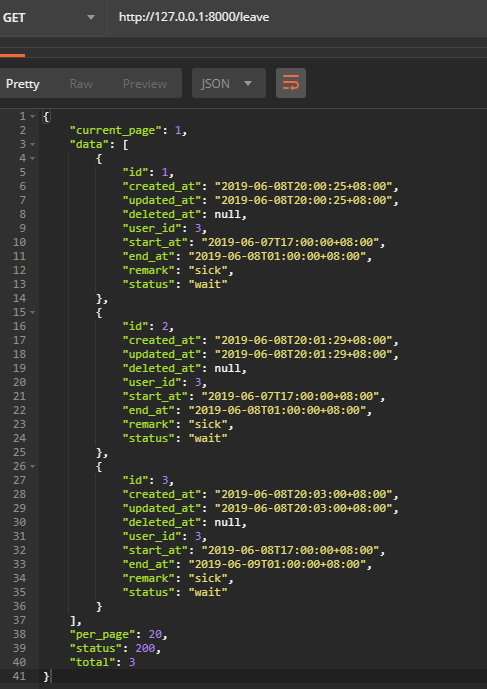
"per\_page": 20,

"status": 200,

"total": 3

}

##### 测试结果：



#### 078：经理获取请假列表

##### 输入：

登录用户root

URL：GET http://127.0.0.1:8000/leave

##### 预期输出：

可获得所有请假记录

{

"current\_page": 1,

"data": [

{

"id": 1,

"created\_at": "2019-06-08T20:00:25+08:00",

"updated\_at": "2019-06-08T20:00:25+08:00",

"deleted\_at": null,

"user\_id": 3,

"start\_at": "2019-06-07T17:00:00+08:00",

"end\_at": "2019-06-08T01:00:00+08:00",

"remark": "sick",

"status": "wait"

},

......

],

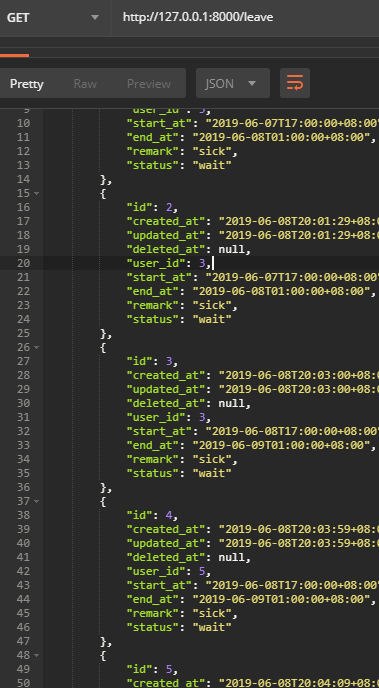
"per\_page": 20,

"status": 200,

"total": 5

}

##### 测试结果：



### 功能点19：审批请假

#### 079：用户审批请假

##### 输入：

登录用户user2

URL：PUT <http://127.0.0.1:8000/leave/1>

JSON：

{

"status": "pass"

}

##### 预期输出：

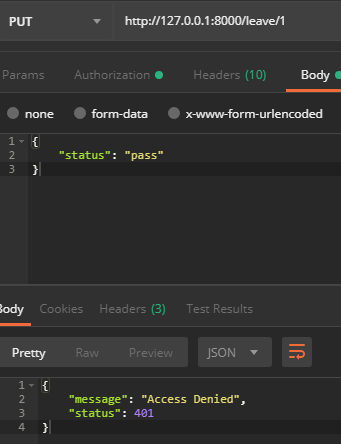
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 080：部门主管审批本部门请假

##### 输入：

登录部门主管user1

URL：PUT <http://127.0.0.1:8000/leave/1>

JSON：

{

"status": "pass"

}

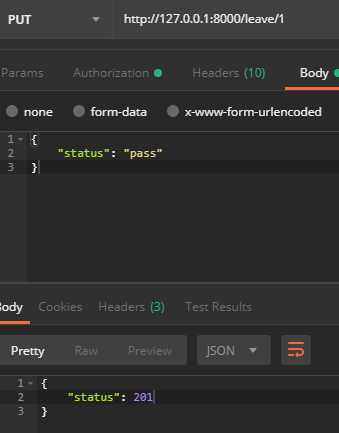
##### 预期输出：

{

"status": 201

}

##### 测试结果：



#### 081：部门主管审批其他部门请假

##### 输入：

登录部门主管user1

URL：PUT [http://127.0.0.1:8000/leave/](http://127.0.0.1:8000/leave/1)5

JSON：

{

"status": "pass"

}

##### 预期输出：

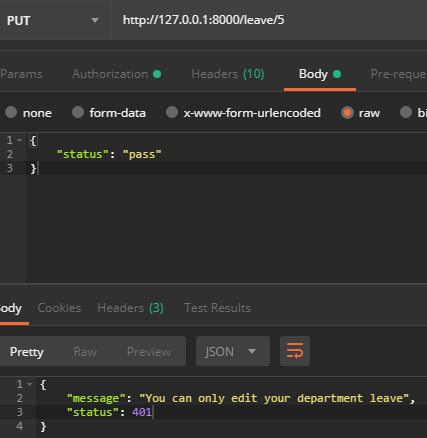
{

"message": "You can only edit your department leave",

"status": 401

}

##### 测试结果：



#### 082：经理审批请假

##### 输入：

登录经理root

URL：PUT [http://127.0.0.1:8000/leave/](http://127.0.0.1:8000/leave/1)5

JSON：

{

"status": "pass"

}

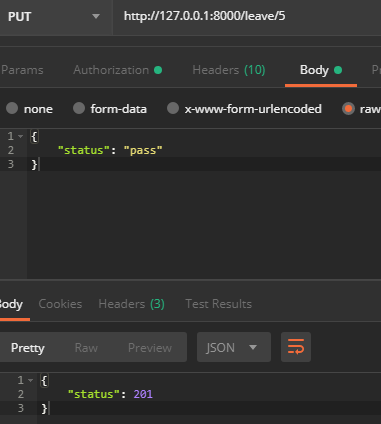
##### 预期输出：

{

"status": 201

}

##### 测试结果：



### 功能点20：销假

#### 083：用户给自己销假（请假已通过）

##### 输入：

登录用户user2

URL：DELET http://127.0.0.1:8000/leave/1

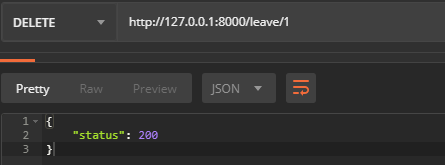
##### 预期输出：

{

"status": 200

}

##### 测试结果：



#### 084：用户给自己销假（请假未通过）

##### 输入：

登录用户user2

URL：DELET http://127.0.0.1:8000/leave/2

##### 预期输出：

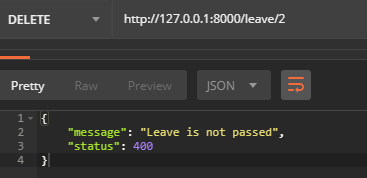
{

"message": "Leave is not passed",

"status": 400

}

##### 测试结果：



#### 085：用户给其他用户销假

##### 输入：

登录用户user2

URL：DELET http://127.0.0.1:8000/leave/5

##### 预期输出：

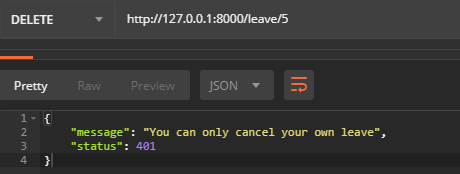
{

"message": "You can only cancel your own leave",

"status": 401

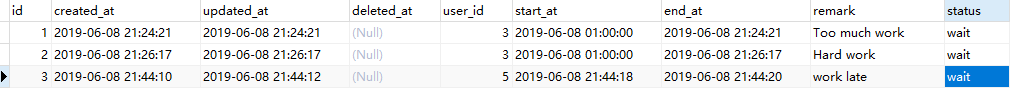
}

##### 测试结果：



## 加班相关

参考数据库：



### 功能点21：申请加班

#### 086：用户申请加班（无排班）

##### 输入：

登录用户user2

当天没有已经签退off的排班

URL：POST <http://127.0.0.1:8000/overtime/user/3>

JSON：

{

"remark": "Too much work"

}

##### 预期输出：

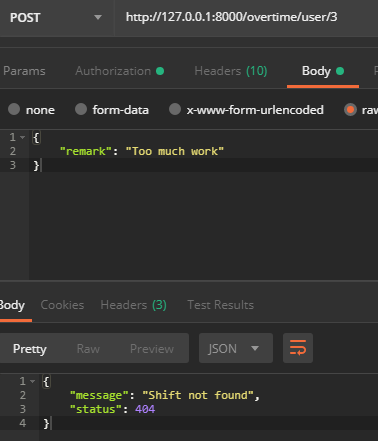
{

"message": "Shift not found",

"status": 404

}

##### 测试结果：



#### 087：用户申请加班（有排班）

##### 输入：

登录用户user2

当天有已经签退off的排班

URL：POST <http://127.0.0.1:8000/overtime/user/3>

JSON：

{

"remark": "Too much work"

}

##### 预期输出：

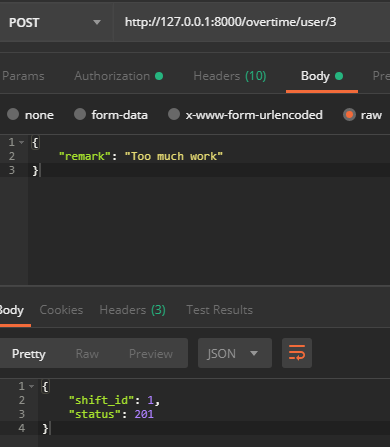
{

"shift\_id": 1,

"status": 201

}

##### 测试结果：



### 功能点22：获取指定用户加班

#### 088：用户获取本人加班情况

##### 输入：

登录用户user2:

URL：http://127.0.0.1:8000/overtime/user/3

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"id": 1,

"created\_at": "2019-06-08T21:24:21+08:00",

"updated\_at": "2019-06-08T21:24:21+08:00",

"deleted\_at": null,

"user\_id": 3,

"start\_at": "2019-06-08T01:00:00+08:00",

"end\_at": "2019-06-08T21:24:21+08:00",

"remark": "Too much work",

"status": "wait"

},

{

"id": 2,

"created\_at": "2019-06-08T21:26:17+08:00",

"updated\_at": "2019-06-08T21:26:17+08:00",

"deleted\_at": null,

"user\_id": 3,

"start\_at": "2019-06-08T01:00:00+08:00",

"end\_at": "2019-06-08T21:26:17+08:00",

"remark": "Hard work",

"status": "wait"

}

],

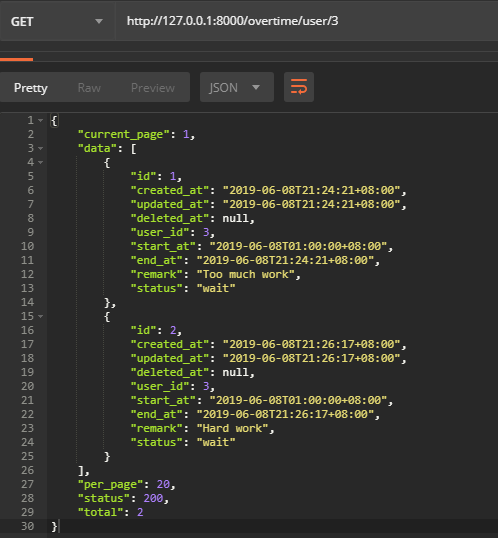
"per\_page": 20,

"status": 200,

"total": 2

}

##### 测试结果：



#### 089：用户获取其他用户加班情况

##### 输入：

登录员工user2

URL：http://127.0.0.1:8000/overtime/user/4

##### 预期输出：

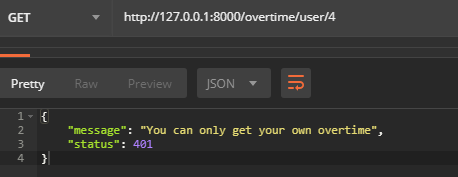
{

"message": "You can only get your own overtime",

"status": 401

}

##### 测试结果：



#### 090：部门主管获取本部门员工加班情况

##### 输入：

登录部门主管user1

URL：http://127.0.0.1:8000/overtime/user/3

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"id": 1,

"created\_at": "2019-06-08T21:24:21+08:00",

"updated\_at": "2019-06-08T21:24:21+08:00",

"deleted\_at": null,

"user\_id": 3,

"start\_at": "2019-06-08T01:00:00+08:00",

"end\_at": "2019-06-08T21:24:21+08:00",

"remark": "Too much work",

"status": "wait"

},

{

"id": 2,

"created\_at": "2019-06-08T21:26:17+08:00",

"updated\_at": "2019-06-08T21:26:17+08:00",

"deleted\_at": null,

"user\_id": 3,

"start\_at": "2019-06-08T01:00:00+08:00",

"end\_at": "2019-06-08T21:26:17+08:00",

"remark": "Hard work",

"status": "wait"

}

],

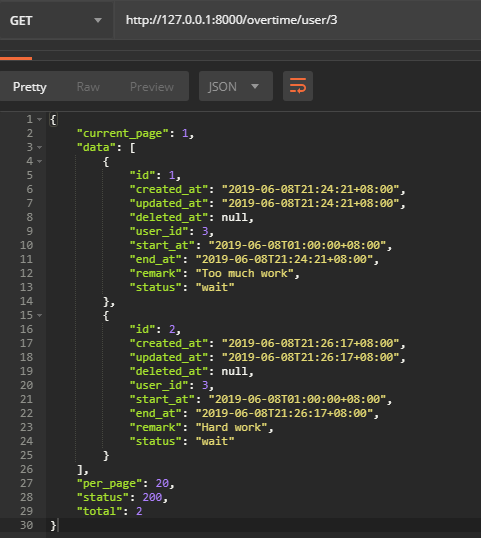
"per\_page": 20,

"status": 200,

"total": 2

}

##### 测试结果：



#### 091：部门主管获取其他部门员工加班情况

##### 输入：

登录部门主管user1

URL：http://127.0.0.1:8000/overtime/user/5

##### 预期输出：

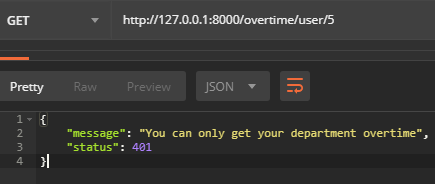
{

"message": "You can only get your department overtime",

"status": 401

}

##### 测试结果：



#### 092：经理获得任意员工加班情况

##### 输入：

登录经理root

URL：http://127.0.0.1:8000/overtime/user/5

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"id": 3,

"created\_at": "2019-06-08T21:44:10+08:00",

"updated\_at": "2019-06-08T21:44:12+08:00",

"deleted\_at": null,

"user\_id": 5,

"start\_at": "2019-06-08T21:44:18+08:00",

"end\_at": "2019-06-08T21:44:20+08:00",

"remark": "work late",

"status": "wait"

}

],

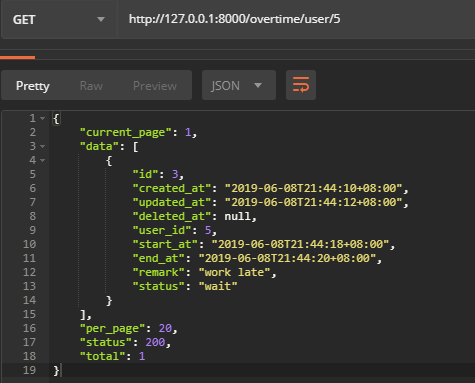
"per\_page": 20,

"status": 200,

"total": 1

}

##### 测试结果：



### 功能点23：获取加班申请列表

093：用户获取加班列表

##### 输入：

登录用户user2

URL：GET http://127.0.0.1:8000/overtime

##### 预期输出：

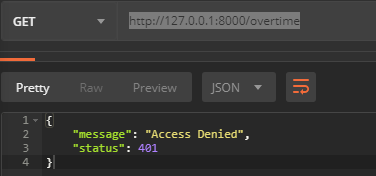
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 094：部门主管获取本部门加班申请列表

##### 输入：

登录用户user1

URL：GET http://127.0.0.1:8000/overtime

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"id": 2,

"created\_at": "2019-06-08T21:26:17+08:00",

"updated\_at": "2019-06-08T21:26:17+08:00",

"deleted\_at": null,

"user\_id": 3,

"start\_at": "2019-06-08T01:00:00+08:00",

"end\_at": "2019-06-08T21:26:17+08:00",

"remark": "Hard work",

"status": "wait"

},

{

"id": 1,

"created\_at": "2019-06-08T21:24:21+08:00",

"updated\_at": "2019-06-08T21:24:21+08:00",

"deleted\_at": null,

"user\_id": 3,

"start\_at": "2019-06-08T01:00:00+08:00",

"end\_at": "2019-06-08T21:24:21+08:00",

"remark": "Too much work",

"status": "wait"

}

],

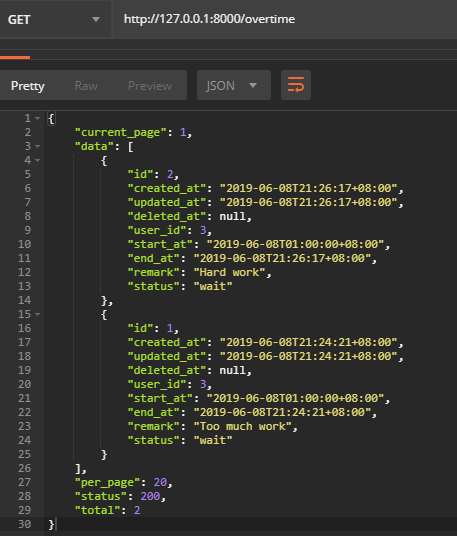
"per\_page": 20,

"status": 200,

"total": 2

}

##### 测试结果：



#### 095：经理获取加班申请列表

##### 输入：

登录经理root

URL：GET http://127.0.0.1:8000/overtime

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"id": 3,

"created\_at": "2019-06-08T21:44:10+08:00",

"updated\_at": "2019-06-08T21:44:12+08:00",

"deleted\_at": null,

"user\_id": 5,

"start\_at": "2019-06-08T21:44:18+08:00",

"end\_at": "2019-06-08T21:44:20+08:00",

"remark": "work late",

"status": "wait"

},

{

"id": 2,

"created\_at": "2019-06-08T21:26:17+08:00",

"updated\_at": "2019-06-08T21:26:17+08:00",

"deleted\_at": null,

"user\_id": 3,

"start\_at": "2019-06-08T01:00:00+08:00",

"end\_at": "2019-06-08T21:26:17+08:00",

"remark": "Hard work",

"status": "wait"

},

{

"id": 1,

"created\_at": "2019-06-08T21:24:21+08:00",

"updated\_at": "2019-06-08T21:24:21+08:00",

"deleted\_at": null,

"user\_id": 3,

"start\_at": "2019-06-08T01:00:00+08:00",

"end\_at": "2019-06-08T21:24:21+08:00",

"remark": "Too much work",

"status": "wait"

}

],

"per\_page": 20,

"status": 200,

"total": 3

}

##### 测试结果



### 功能点24：审批加班

#### 096：用户审批加班

##### 输入：

登录用户user2

URL：PUT <http://127.0.0.1:8000/overtime/1>

JSON：

{

"status": "pass"

}

##### 预期输出：

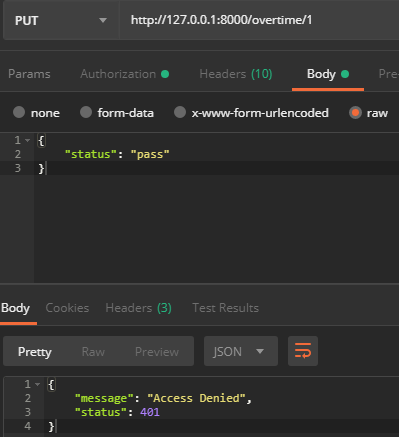
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 097：部门主管审批本部门用户加班

##### 输入：

登录主管user1

URL：PUT <http://127.0.0.1:8000/overtime/1>

JSON：

{

"status": "pass"

}

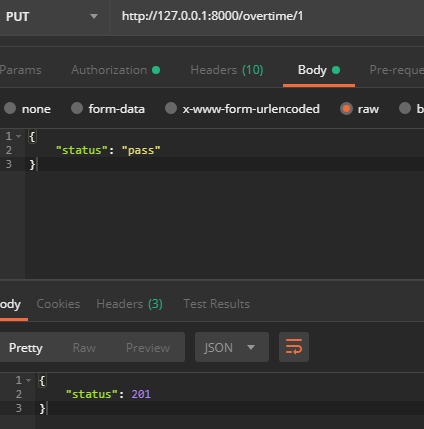
##### 预期输出：

{

"status": 201

}

##### 测试结果：



#### 098：部门主管审批其他部门用户加班

##### 输入：

登录主管user1

URL：PUT [http://127.0.0.1:8000/overtime/](http://127.0.0.1:8000/overtime/1)3

JSON：

{

"status": "pass"

}

##### 预期输出：

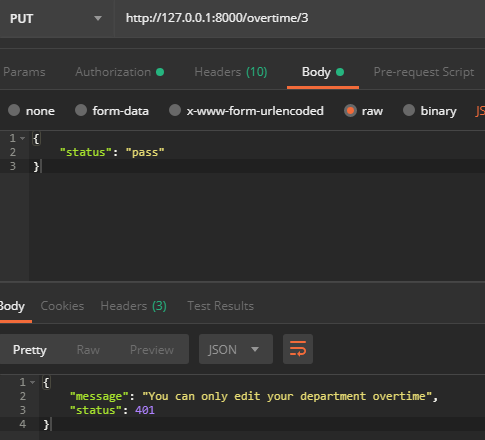
{

"message": "You can only edit your department overtime",

"status": 401

}

##### 测试结果：



099：部门主管审批加班状态输入错误

输入：

登录主管user1

URL：PUT [http://127.0.0.1:8000/overtime/](http://127.0.0.1:8000/overtime/1)1

JSON：

{

"status": "test"

}

**预期输出：**

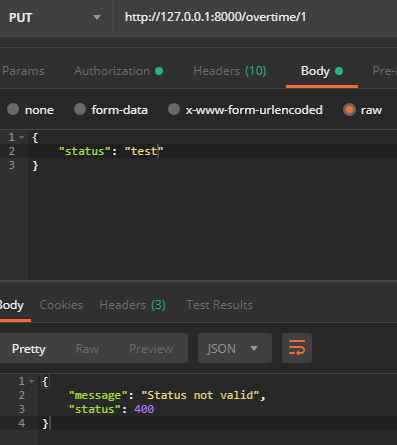
{

"message": "Status not valid",

"status": 400

}

测试结果：



## 人脸相关

### 功能点25：更新指定用户人脸信息

#### 099：用户正确添加人脸

##### 输入：

登录用户user2给自己添加人脸信息

URL：http://127.0.0.1:8000/face/user/3

{

"info": "data:image/jpeg;base64,/9j/4AAQSk......ZPUY9PU0vlQep/WmI//2Q=="

}

##### 预期输出：

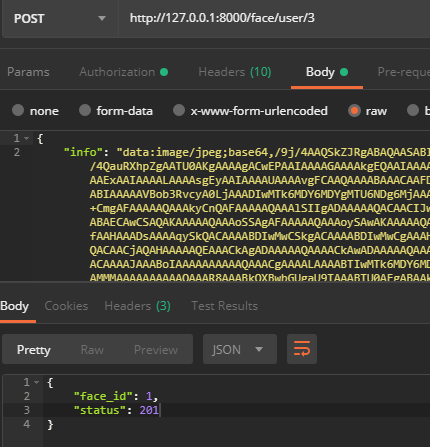
{

"face\_id": 1,

"status": 201

}

##### 测试结果：



#### 100：用户给他人添加人脸

##### 输入：

登录用户user2给user3添加人脸

URL：http://127.0.0.1:8000/face/user/4

{

"info": "data:image/jpeg;base64,/9j/4AAQSk......ZPUY9PU0vlQep/WmI//2Q=="

}

##### 预期输出：

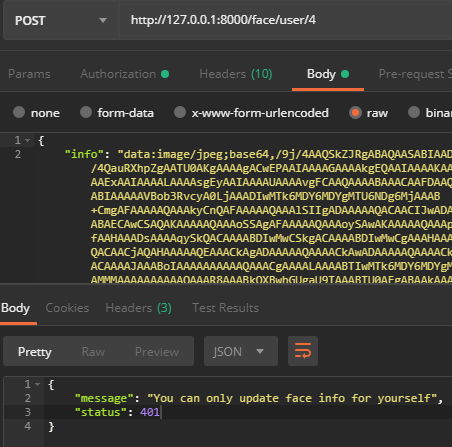
{

"message": "You can only update face info for yourself",

"status": 401

}

##### 测试结果：



#### 101：图片格式不正确

##### 输入：

登录用户user2给自己添加人脸

URL：POST http://127.0.0.1:8000/face/user/3

{

"info": "data:image/jpeg;base64,/9j/4AAQSk......ZPUY9PU0vlQep/WmI//2Q=="

}

这里将info中的数据任意删除中间一部分

##### 预期输出：

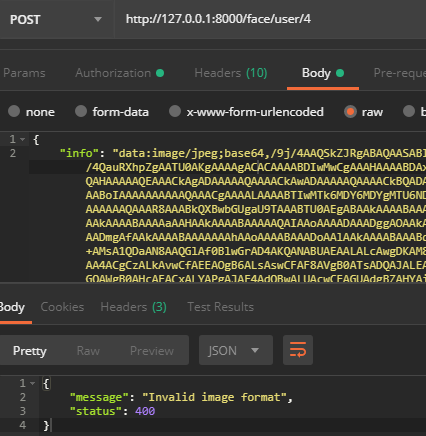
{

"message": "Invalid image format",

"status": 400

}

##### 测试结果：



### 功能点26：获取指定用户可用的人脸信息

#### 102：用户获取自己的人脸信息

##### 输入：

登录user2获取自己的人脸信息

URL：GET http://127.0.0.1:8000/face/user/3

##### 预期输出：

{

"data": {

"id": 1,

"created\_at": "2019-06-09T09:31:16+08:00",

"updated\_at": "2019-06-09T09:31:16+08:00",

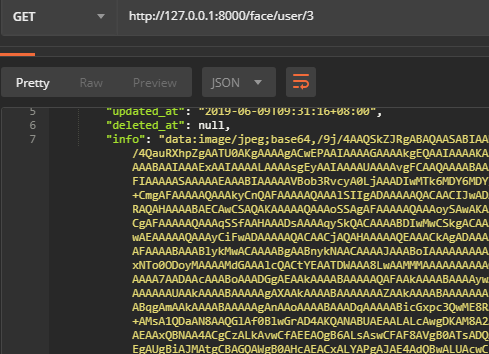
"deleted\_at": null,

"info":"data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAASABIAACwE...”

}

}

##### 测试结果：



#### 103：用户获取他人的人脸信息

##### 输入：

登录user2获取自己的人脸信息

URL：GET http://127.0.0.1:8000/face/user/4

##### 预期输出：

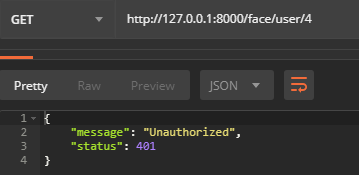
{

"message": "Unauthorized",

"status": 401

}

##### 测试结果：



#### 104：部门主管获取部门成员人脸信息

##### 输入：

登录主管user1获取user2的人脸信息（这里要求部门）

URL：GET http://127.0.0.1:8000/face/user/3

##### 预期输出：

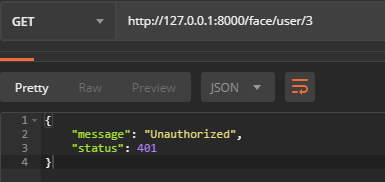
{

"message": "Unauthorized",

"status": 401

}

##### 测试结果：



#### 105：经理获取成员的人脸信息

##### 输入：

登录经理root获取user2的人脸信息（这里要求部门）

URL：GET http://127.0.0.1:8000/face/user/3

##### 预期输出：

{

"data": {

"id": 1,

"created\_at": "2019-06-09T09:31:16+08:00",

"updated\_at": "2019-06-09T09:31:16+08:00",

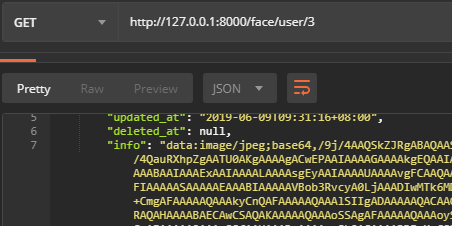
"deleted\_at": null,

"info":"data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAASABIAACwE...”

}

}

##### 测试结果：



### 功能点27：获取人脸列表

#### 106：用户获取人脸列表

##### 输入：

登录用户user2

URL：GET http://127.0.0.1:8000/face

##### 预期输出：

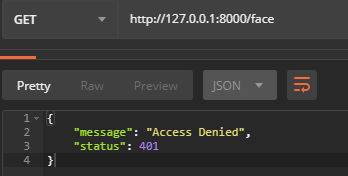
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 107：部门主管获取人脸列表

##### 输入：

登录部门主管user1

URL：GET http://127.0.0.1:8000/face

##### 预期输出：

部门主管没有权限获得用户的人脸列表

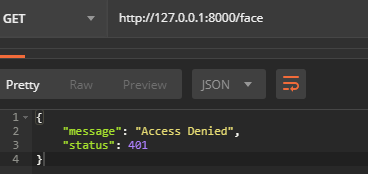
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 108：经理获取人脸列表

##### 输入：

登录经理root

URL：GET http://127.0.0.1:8000/face

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"id": 1,

"created\_at": "2019-06-09T09:31:16+08:00",

"updated\_at": "2019-06-09T09:31:16+08:00",

"deleted\_at": null,

"info": "data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAASABIAAD/4......

}

"status": "available",

"user": {

"id": 3,

"created\_at": "2019-06-07T19:42:12+08:00",

"updated\_at": "2019-06-08T22:20:33+08:00",

"deleted\_at": null,

"name": "user2",

"role": "user",

"department": 1,

"hours": 1

}

}

],

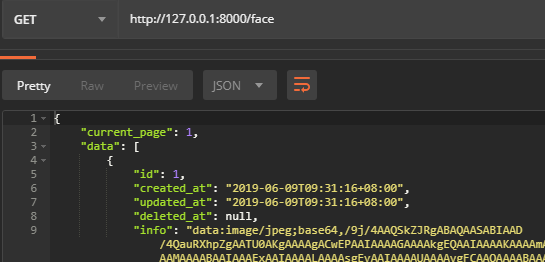
"per\_page": 20,

"status": 200,

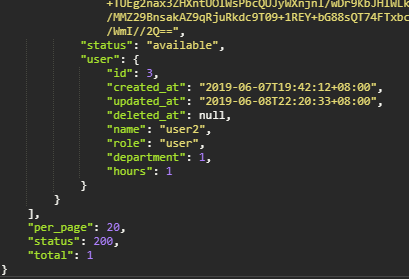
"total": 1

}

##### 测试结果：



......(此处省略info)



### 功能点28：审批人脸信息

#### 109：用户审批人脸信息

##### 输入：

登录用户user2

URL：PUT http://127.0.0.1:8000/face/1

##### 预期输出：

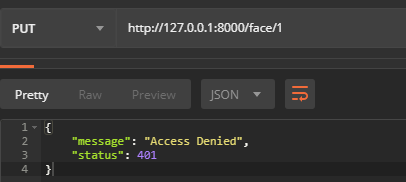
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 110：部门主管审批人脸

##### 输入：

登录部门主管user1

URL：PUT http://127.0.0.1:8000/face/1

##### 预期输出：

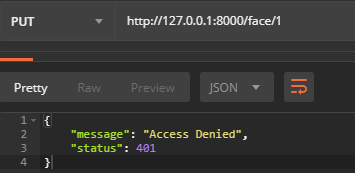
{

"message": "Access Denied",

"status": 401

}

##### 测试结果：



#### 111：经理审批人脸

##### 输入：

登录经理root

URL：PUT http://127.0.0.1:8000/face/1

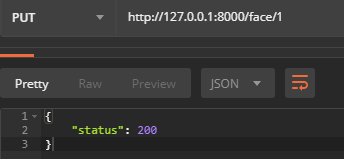
##### 预期输出：

{

"status": 200

}

##### 测试结果：



## 签到相关

### 功能点29：获取二维码

#### 112：部门主管获取二维码

##### 输入：

登录部门主管user1

URL：GET http://127.0.0.1:8000/sign/qrcode

##### 预期输出：

{

"data": {

"expired\_at": "2019-06-09T11:09:06.3216424+08:00",

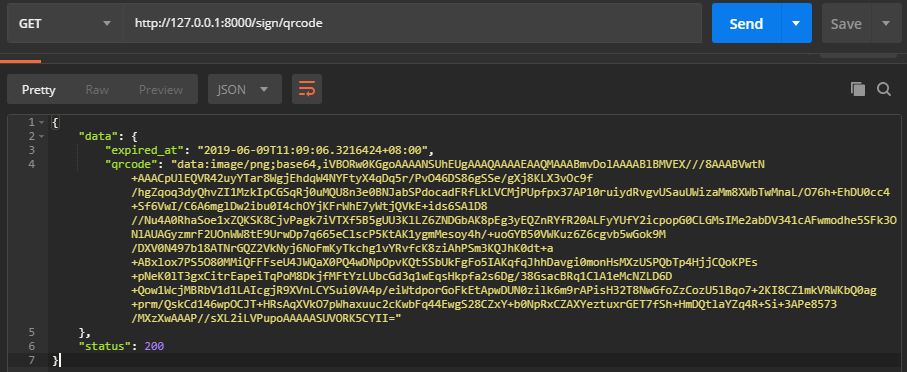
"qrcode": "data:image/png;base64,"

},

"status": 200

}

##### 测试结果：



#### 113：用户获取二维码

##### 输入：

登录用户user2

URL：GET <http://127.0.0.1:8000/sign/qrcode>

##### 预期输出：

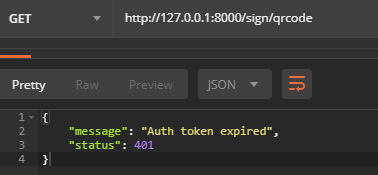
{

"message": "Auth token expired",

"status": 401

}

##### 测试结果：



### 功能点30：获取当前用户签到状态

#### 114：用户获取自己的签到状态（未签到）

##### 输入：

登录用户user2

没有进行签到

URL：GET <http://127.0.0.1:8000/>sign/user/3

##### 预期输出：

(内容为空)

##### 测试结果：

#### 115：用户获取自己签到状态（已经签到）

##### 输入：

登录用户user2

已经签到

URL：GET <http://127.0.0.1:8000/>sign/user/3

##### 预期输出：

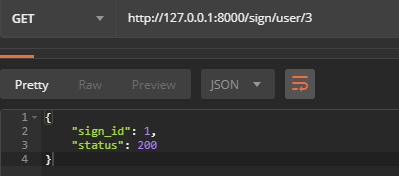
{

"sign\_id": 1,

"status": 200

}

##### 测试结果：



#### 116：用户获取他人签到状态

##### 输入：

登录用户user2

已经签到

URL：GET <http://127.0.0.1:8000/>sign/user/5

##### 预期输出：

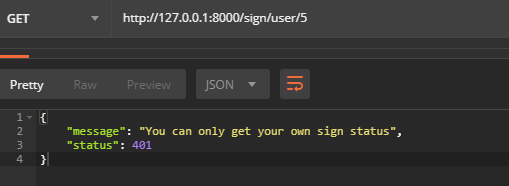
{

"message": "You can only get your own sign status",

"status": 401

}

##### 测试结果：



#### 117：部门主管获取本部门员工签到状态

##### 输入：

登录部门主管user1 查看本部门员工user2的签到状态

URL：GET <http://127.0.0.1:8000/>sign/user/3

##### 预期输出：

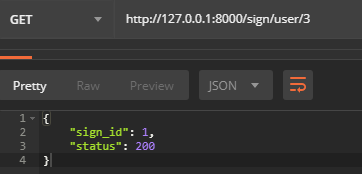
{

"sign\_id": 1,

"status": 200

}

##### 测试结果：



#### 118：部门主管获取其他部门员工签到状态

##### 输入：

登录部门主管user1 查看本部门员工user2的签到状态

URL：GET <http://127.0.0.1:8000/>sign/user/5

##### 预期输出：

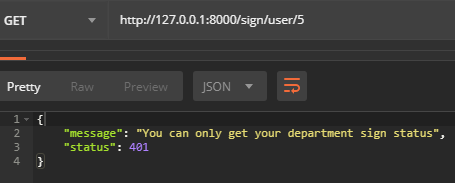
{

"message": "You can only get your department sign status",

"status": 401

}

##### 测试结果：



#### 119：经理获取他人签到状态

##### 输入：

登录经理root查看员工user2的签到状态

URL：GET <http://127.0.0.1:8000/>sign/user/3

##### 预期输出：

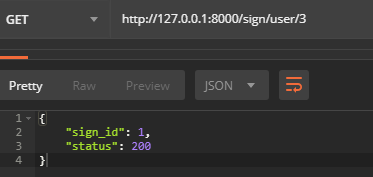
{

"sign\_id": 1,

"status": 200

}

##### 测试结果：



### 功能点31：二维码签到

#### 120：用户自己进行二维码签到

##### 输入：

登录user2:

URL：POST <http://127.0.0.1:8000/sign/qrcode/3>

JSON：

{

"token": "syCZczAFyzcRfNqgyWBmDbcBamVyYIijkBOhXxjjEoGcqNbQZPujjuusrtlrDYAfPsIlhOVQxHiEjXARCXvJkqHPpVIKYJYlfFMbvCxqscQJGPyAnulzCZBONtBbwAsn"

}

##### 预期输出：

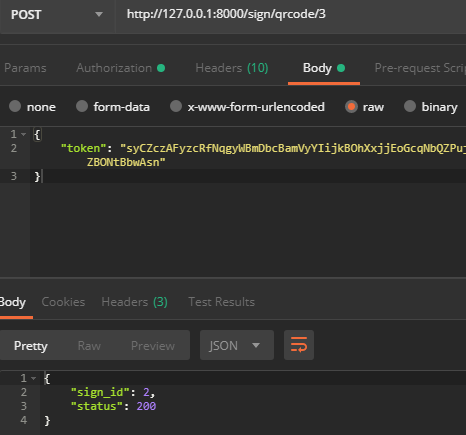
{

"sign\_id": 2,

"status": 200

}

##### 测试结果：



#### 121：用户给他人进行二维码签到

##### 输入：

登录user2:

URL：POST <http://127.0.0.1:8000/sign/qrcode/3>

JSON：

{

"token": "syCZczAFyzcRfNqgyWBmDbcBamVyYIijkBOhXxjjEoGcqNbQZPujjuusrtlrDYAfPsIlhOVQxHiEjXARCXvJkqHPpVIKYJYlfFMbvCxqscQJGPyAnulzCZBONtBbwAsn"

}

##### 预期输出：

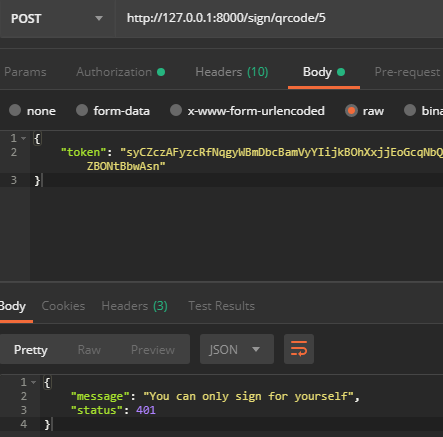
{

"message": "You can only sign for yourself",

"status": 401

}

##### 测试结果：



#### 122：用户签到token错误

##### 输入：

登录user2:

URL：POST <http://127.0.0.1:8000/sign/qrcode/3>

JSON：

{

"token": "syCZczAFyzcRfNqgyWBmDbcBamxjjEoGcqNbQZPujjuusrtlrDYAfPsIlhOVQxHiEjXARCXvJkqHPpVIKYJYlfFMbvCxqscQJGPyAnulzCZBONtBbwAsn"

}

##### 预期输出：

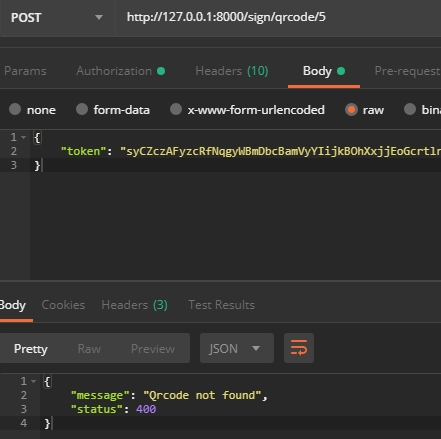
{

"message": "Qrcode not found",

"status": 400

}

##### 测试结果：



### 功能点32：人脸签到

这里不容易在后端单独测试，需要结合前端一起测试

## 信息相关

数据库参考：



### 功能点33：获取信息列表

#### 123：获取指定发送用户的信息

##### 输入：

登录经理root

URL：GET http://127.0.0.1:8000/message/?from\_user\_id=1

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"id": 7,

"created\_at": "2019-06-08T20:42:48+08:00",

"updated\_at": "2019-06-08T20:42:48+08:00",

"deleted\_at": null,

"title": "请假审批结果",

"content": "请假审批通过",

"status": "unread",

"from": {

"id": 1,

"created\_at": "2019-06-07T19:09:37+08:00",

"updated\_at": "2019-06-07T19:09:37+08:00",

"deleted\_at": null,

"name": "root",

"role": "master",

"department": 0,

"hours": 0

},

"to": {

"id": 5,

"created\_at": "2019-06-07T19:43:54+08:00",

"updated\_at": "2019-06-08T20:42:48+08:00",

"deleted\_at": null,

"name": "user4",

"role": "manager",

"department": 2,

"hours": 1

}

}

],

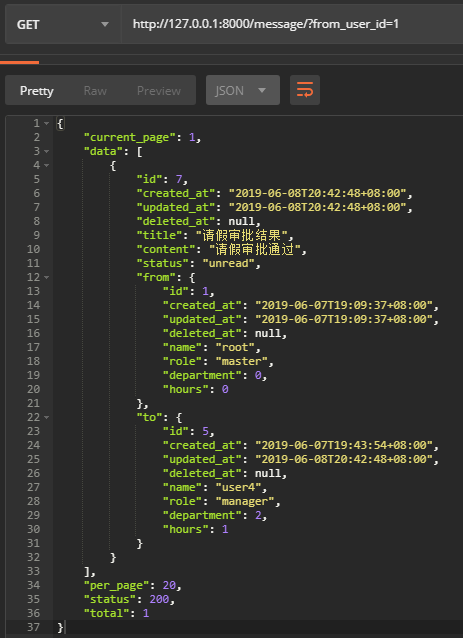
"per\_page": 20,

"status": 200,

"total": 1

}

##### 测试结果：



#### 124：获取指定接收用户的信息

##### 输入：

登录经理root

URL：GET http://127.0.0.1:8000/message/?to\_user\_id=5

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"id": 7,

"created\_at": "2019-06-08T20:42:48+08:00",

"updated\_at": "2019-06-08T20:42:48+08:00",

"deleted\_at": null,

"title": "请假审批结果",

"content": "请假审批通过",

"status": "unread",

"from": {

"id": 1,

"created\_at": "2019-06-07T19:09:37+08:00",

"updated\_at": "2019-06-07T19:09:37+08:00",

"deleted\_at": null,

"name": "root",

"role": "master",

"department": 0,

"hours": 0

},

"to": {

"id": 5,

"created\_at": "2019-06-07T19:43:54+08:00",

"updated\_at": "2019-06-08T20:42:48+08:00",

"deleted\_at": null,

"name": "user4",

"role": "manager",

"department": 2,

"hours": 1

}

}

],

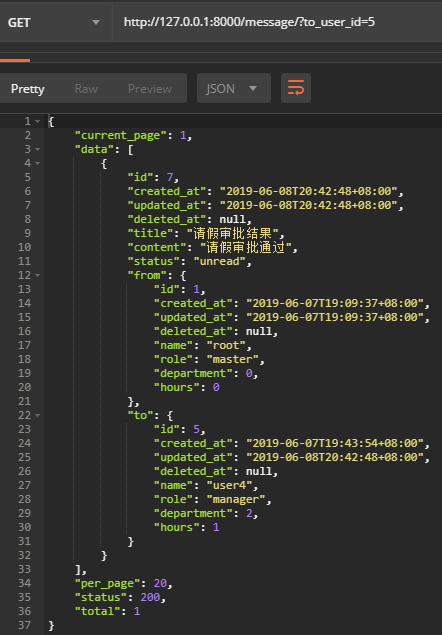
"per\_page": 20,

"status": 200,

"total": 1

}

##### 测试结果：



#### 125：获取指定状态的信息

##### 输入：

登录经理root

URL：GET http://127.0.0.1:8000/message/?status=unread

##### 预期输出：

{

"current\_page": 1,

"data": [

{

"id": 10,

"created\_at": "2019-06-08T22:20:33+08:00",

"updated\_at": "2019-06-08T22:20:33+08:00",

"deleted\_at": null,

"title": "加班申请结果",

"content": "加班申请通过",

"status": "unread",

"from": {

"id": 2,

"created\_at": "2019-06-07T19:35:51+08:00",

"updated\_at": "2019-06-07T19:35:51+08:00",

"deleted\_at": null,

"name": "user1",

"role": "manager",

"department": 1,

"hours": 1

},

"to": {

"id": 3,

"created\_at": "2019-06-07T19:42:12+08:00",

"updated\_at": "2019-06-08T22:20:33+08:00",

"deleted\_at": null,

"name": "user2",

"role": "user",

"department": 1,

"hours": 1

}

},

......(还有很多)

],

"per\_page": 20,

"status": 200,

"total": 10

}

##### 测试结果：



#### 126.用户读取列表（没有收发者为该用户的信息）

##### 输入：

登录用户user2

URL：GET http://127.0.0.1:8000/message/

##### 预期输出：

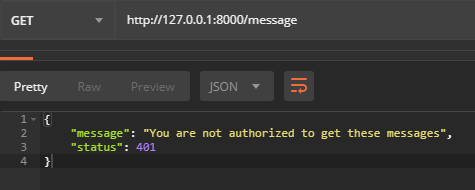
{

"message": "You are not authorized to get these messages",

"status": 401

}

##### 测试结果：



#### 127.用户读取列表（有收发者为该用户的信息）

##### 输入：

登录用户user2

URL：GET http://127.0.0.1:8000/message?to\_user\_id=3

##### 预期输出：

输出所有发送给user2的信息

{

"current\_page": 1,

"data": [

{

"id": 10,

"created\_at": "2019-06-08T22:20:33+08:00",

"updated\_at": "2019-06-08T22:20:33+08:00",

"deleted\_at": null,

"title": "加班申请结果",

"content": "加班申请通过",

"status": "unread",

"from": {

"id": 2,

"created\_at": "2019-06-07T19:35:51+08:00",

"updated\_at": "2019-06-07T19:35:51+08:00",

"deleted\_at": null,

"name": "user1",

"role": "manager",

"department": 1,

"hours": 1

},

"to": {

"id": 3,

"created\_at": "2019-06-07T19:42:12+08:00",

"updated\_at": "2019-06-08T22:20:33+08:00",

"deleted\_at": null,

"name": "user2",

"role": "user",

"department": 1,

"hours": 1

}

},

{

"id": 6,

"created\_at": "2019-06-08T20:39:05+08:00",

"updated\_at": "2019-06-08T20:39:05+08:00",

"deleted\_at": null,

"title": "请假审批结果",

"content": "请假审批通过",

"status": "unread",

"from": {

"id": 2,

"created\_at": "2019-06-07T19:35:51+08:00",

"updated\_at": "2019-06-07T19:35:51+08:00",

"deleted\_at": null,

"name": "user1",

"role": "manager",

"department": 1,

"hours": 1

},

"to": {

"id": 3,

"created\_at": "2019-06-07T19:42:12+08:00",

"updated\_at": "2019-06-08T22:20:33+08:00",

"deleted\_at": null,

"name": "user2",

"role": "user",

"department": 1,

"hours": 1

}

}

],

"per\_page": 20,

"status": 200,

"total": 2

}

##### 测试结果：



### 功能点34：获取指定信息

#### 128：无关人员获取指定信息

##### 输入：

登录user2

URL：GET <http://127.0.0.1:8000/message/7>

（7号message是root发送给user4的信息）

##### 预期输出：

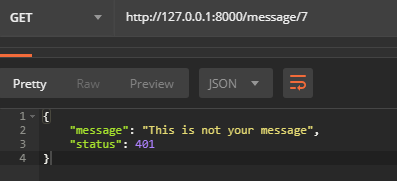
{

"message": "This is not your message",

"status": 401

}

##### 测试结果：



#### 129：发送者获取指定信息

##### 输入：

登录user2

URL：GET <http://127.0.0.1:8000/message/1>

（1号message是user2发送的信息）

##### 预期输出：

{

"data": {

"id": 1,

"created\_at": "2019-06-08T20:00:25+08:00",

"updated\_at": "2019-06-09T15:45:46.295077+08:00",

"title": "请假申请",

"content": "理由：sick",

"status": "unread",

"from": {

"id": 3,

"created\_at": "2019-06-07T19:42:12+08:00",

"updated\_at": "2019-06-09T15:45:46.2930827+08:00",

"name": "user2",

"role": "user",

"department": 1,

"hours": 1

},

"to": {

"id": 0,

"created\_at": "0001-01-01T00:00:00Z",

"updated\_at": "0001-01-01T00:00:00Z",

"name": "",

"role": "",

"department": 0,

"hours": 0

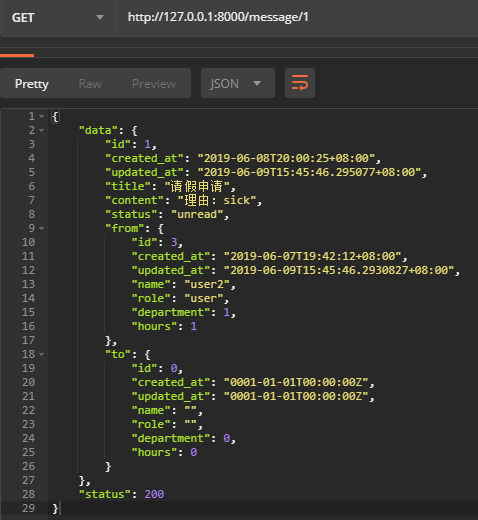
}

},

"status": 200

}

##### 测试结果：



数据库不更改状态：



#### 130：接受者获取指定信息

##### 输入：

登录user2

URL：GET [http://127.0.0.1:8000/message/](http://127.0.0.1:8000/message/1)6

（1号message是user1发送给user2的信息）

##### 预期输出：

{

"data": {

"id": 6,

"created\_at": "2019-06-08T20:39:05+08:00",

"updated\_at": "2019-06-09T15:49:00.714319+08:00",

"title": "请假审批结果",

"content": "请假审批通过",

"status": "unread",

"from": {

"id": 2,

"created\_at": "2019-06-07T19:35:51+08:00",

"updated\_at": "2019-06-09T15:49:00.7123249+08:00",

"name": "user1",

"role": "manager",

"department": 1,

"hours": 1

},

"to": {

"id": 3,

"created\_at": "2019-06-07T19:42:12+08:00",

"updated\_at": "2019-06-09T15:49:00.7133229+08:00",

"name": "user2",

"role": "user",

"department": 1,

"hours": 1

}

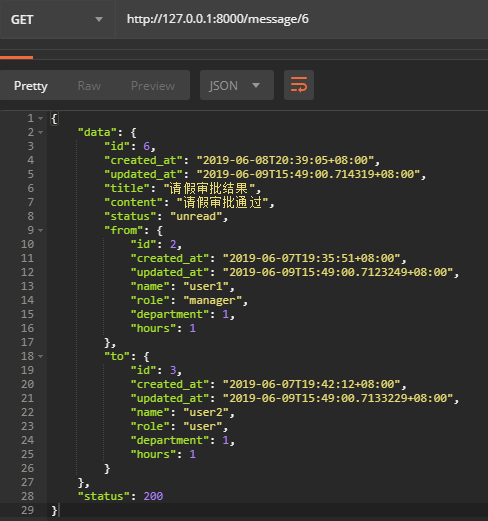
},

"status": 200

}

且数据库中6号message的状态改为read

##### 测试结果：



且数据库6号message的状态修改修改为read：

