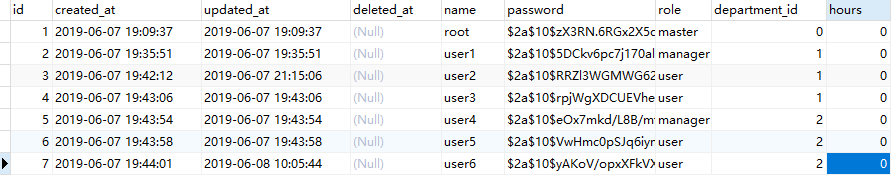
# 本机进行后端测试用例

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：

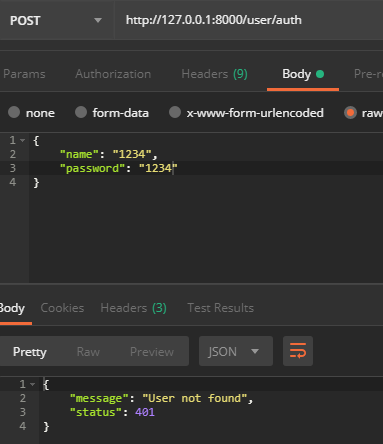
{

"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：

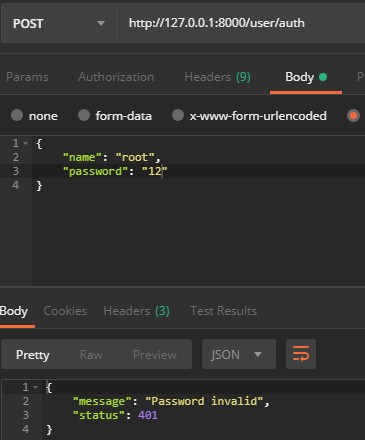
{

"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：

{

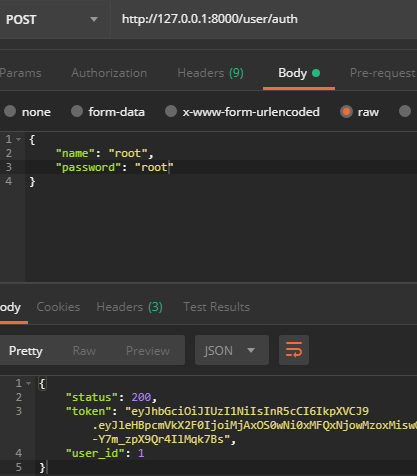
"status": 200,

"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：

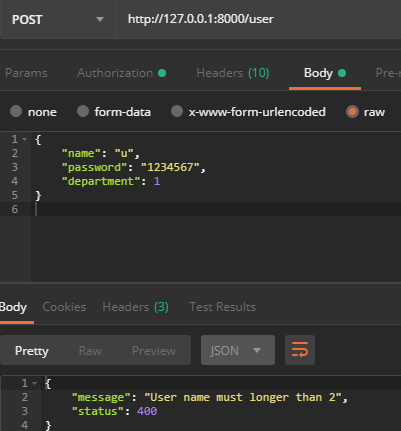
{

"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：

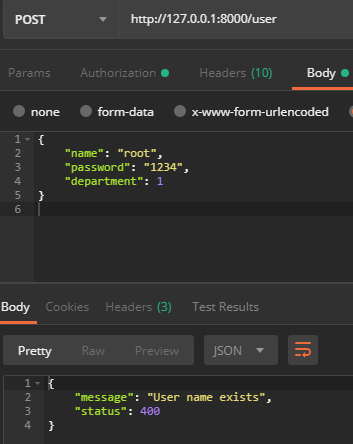
{

"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：

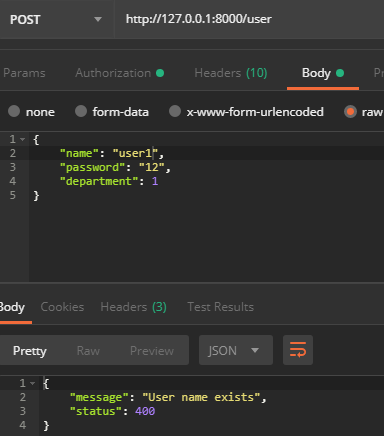
{

"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：

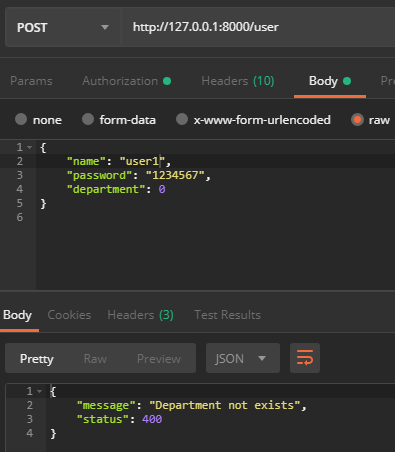
{

"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：

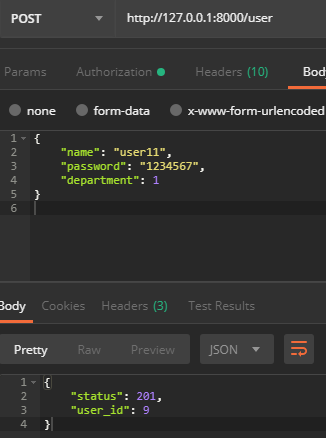
{

"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：

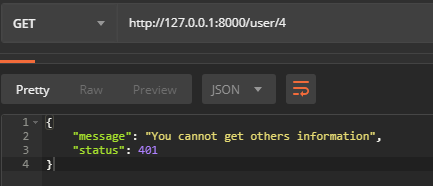
{

"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,

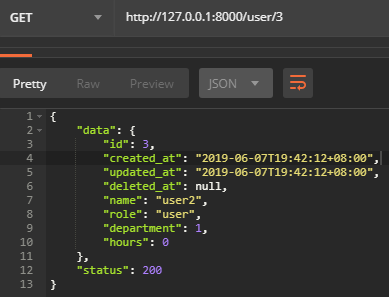
"hours": 0

},

"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,

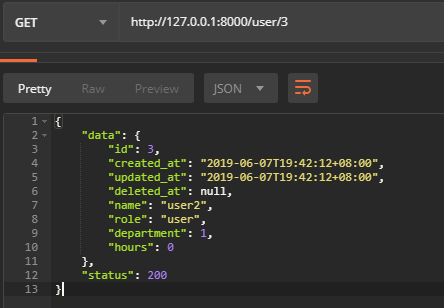
"hours": 0

},

"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：

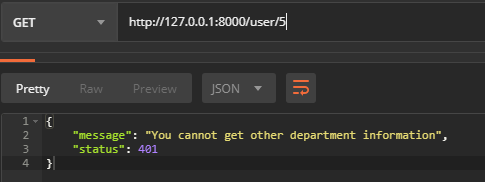
{

"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,

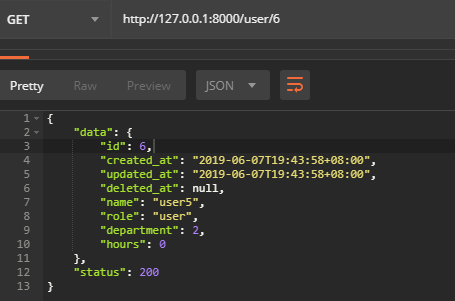
"hours": 0

},

"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：

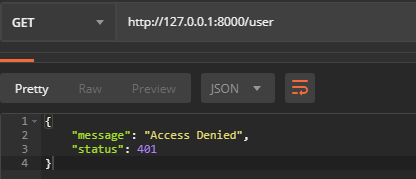
{

"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

},

......

],

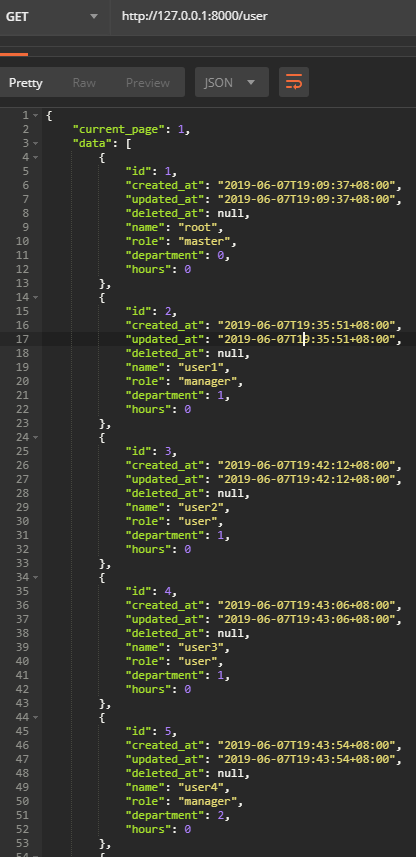
"per\_page": 20,

"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：

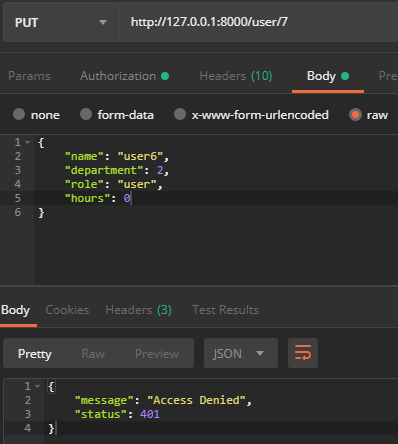
{

"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：

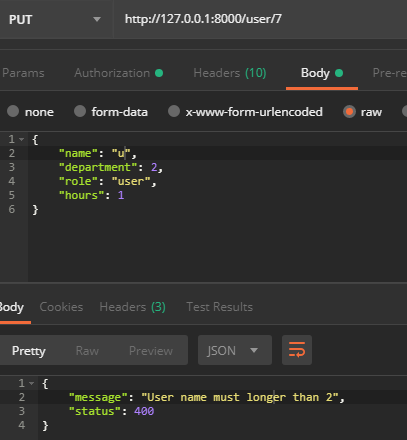
{

"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：

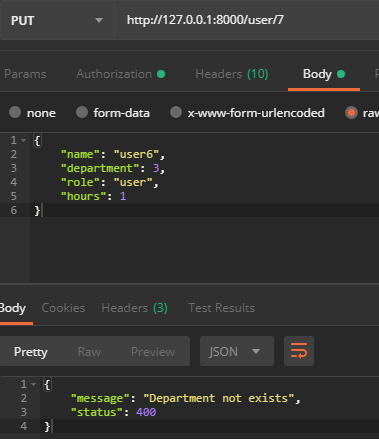
{

"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：

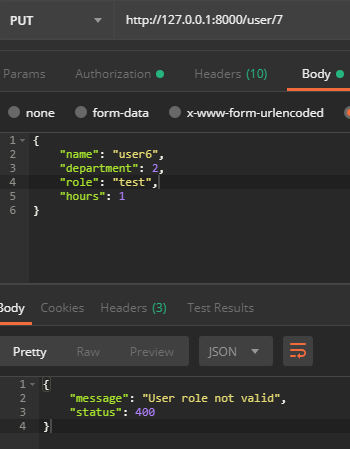
{

"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：

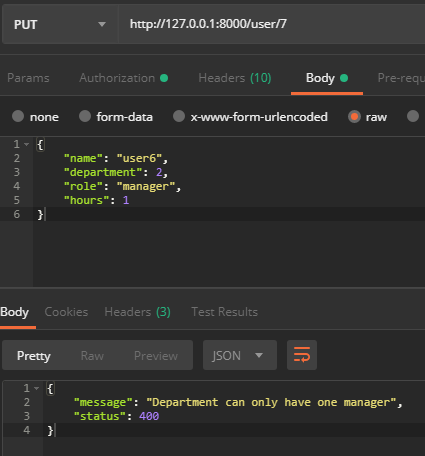
{

"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：

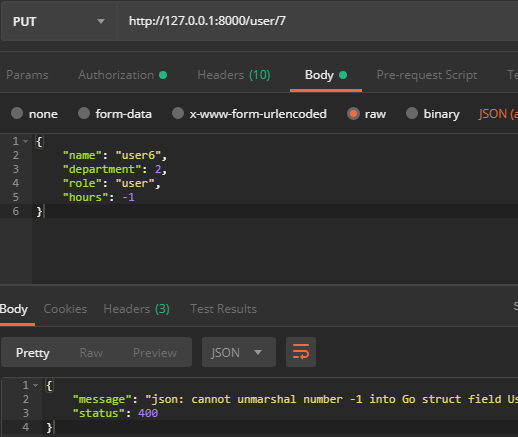
{

"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

}

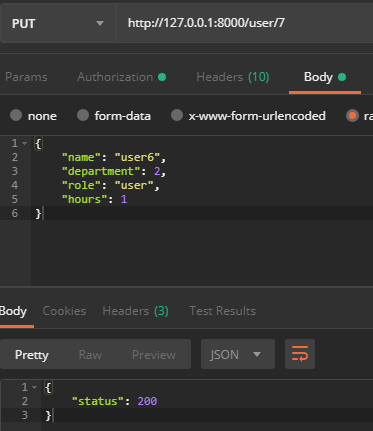
##### 预期输出：

{

"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：

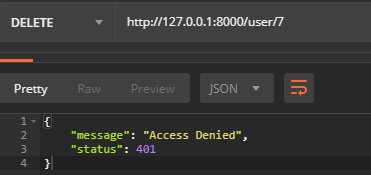
{

"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：

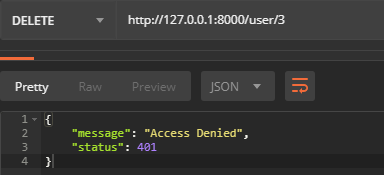
{

"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：

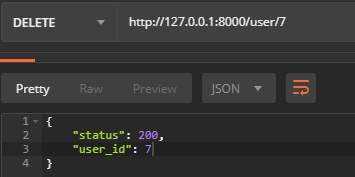
{

"status": 200,

"user\_id": 7

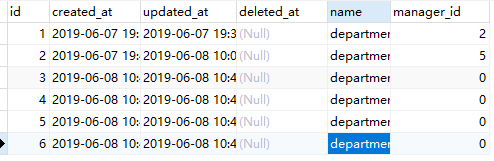
}

##### 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：

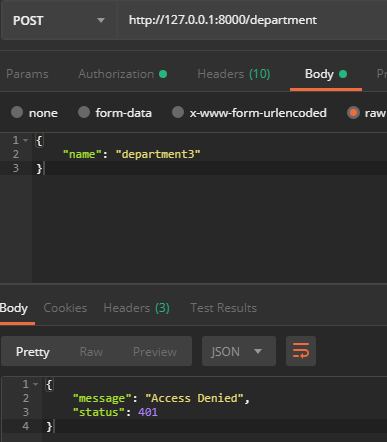
{

"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：

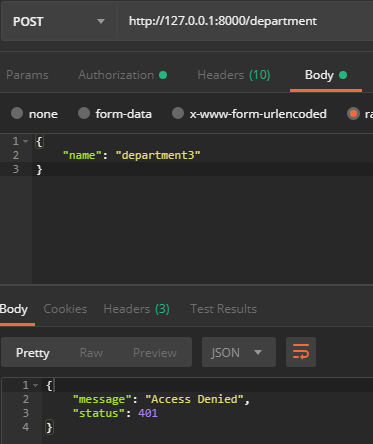
{

"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

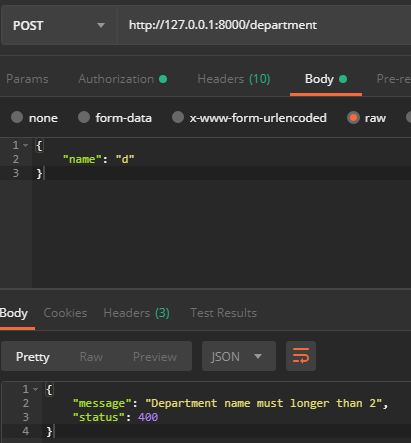
{

"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：

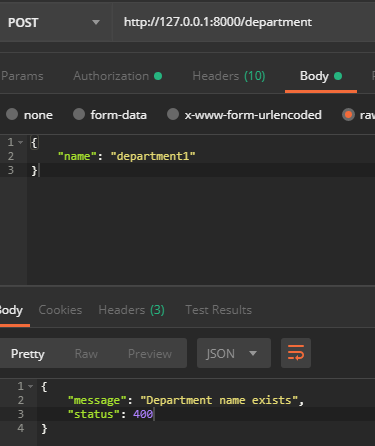
{

"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：

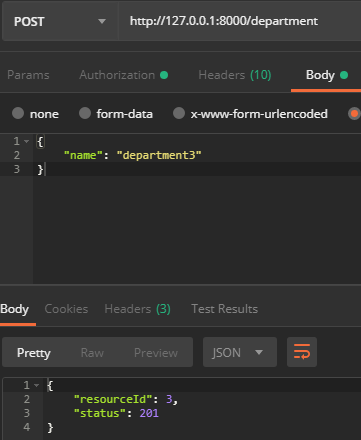
{

"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：

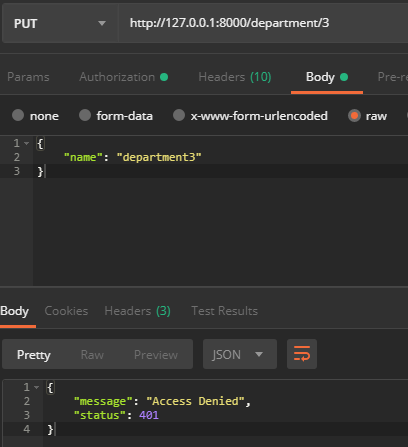
{

"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：

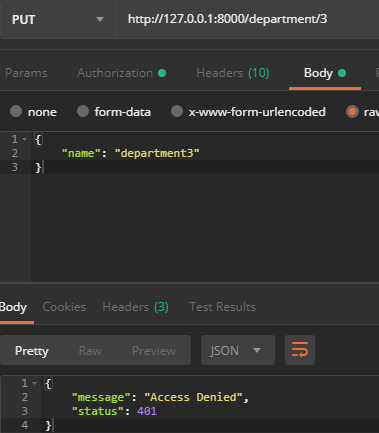
{

"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：

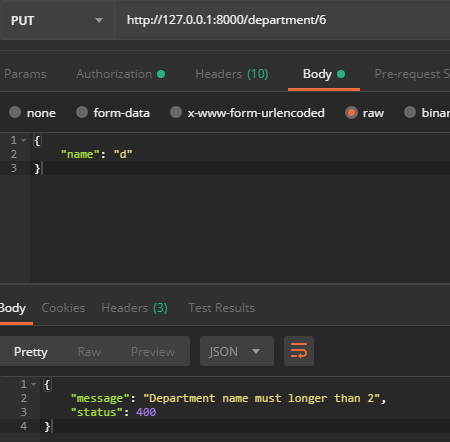
{

"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：

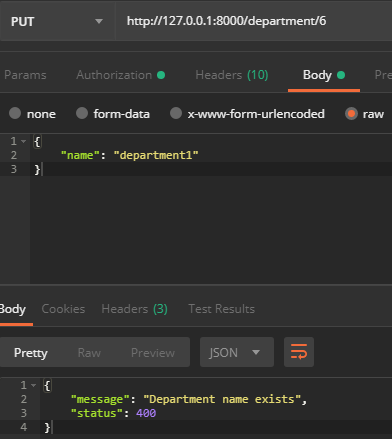
{

"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"

}

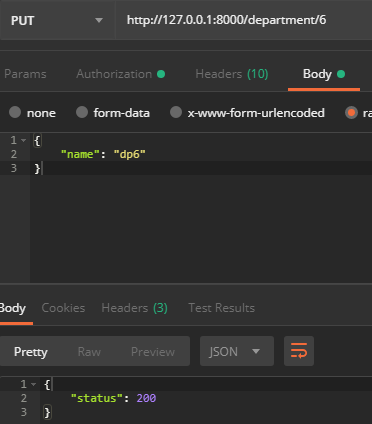
##### 预期输出：

{

"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：

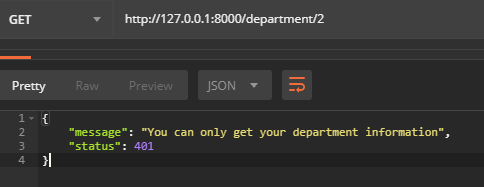
{

"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",

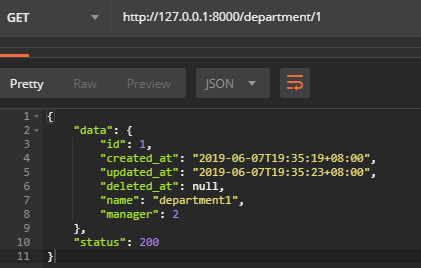
"manager": 2

},

"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：

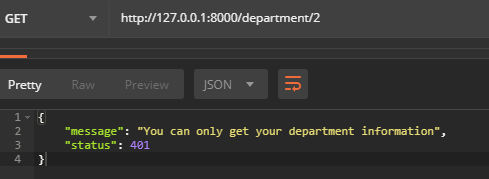
{

"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",

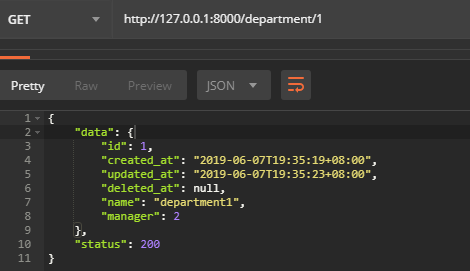
"manager": 2

},

"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：

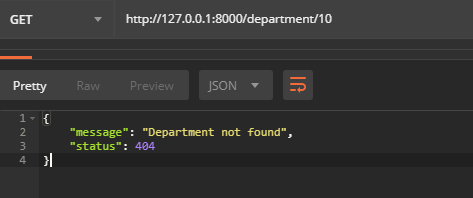
{

"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",

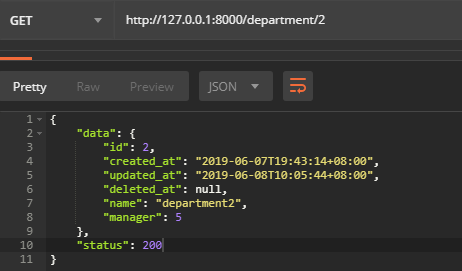
"manager": 5

},

"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：

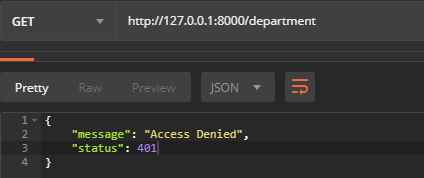
{

"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：

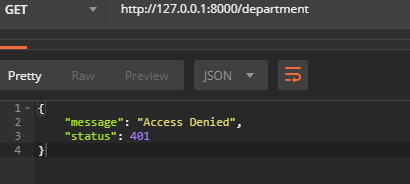
{

"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

},

......

],

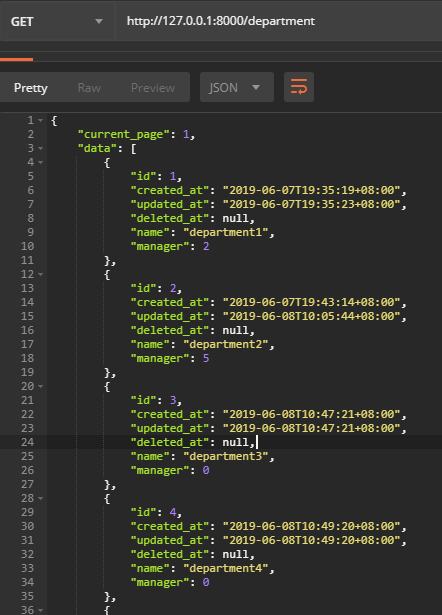
"per\_page": 20,

"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：

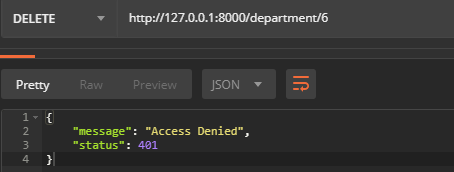
{

"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：

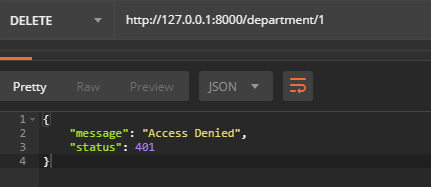
{

"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

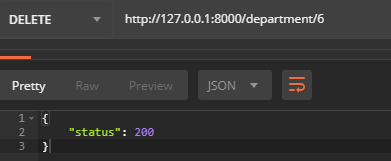
##### 预期输出：

{

"status": 200

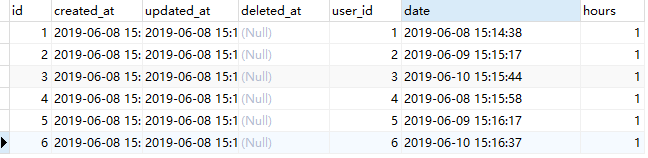
}

##### 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：

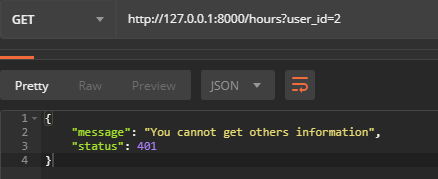
{

"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

}

],

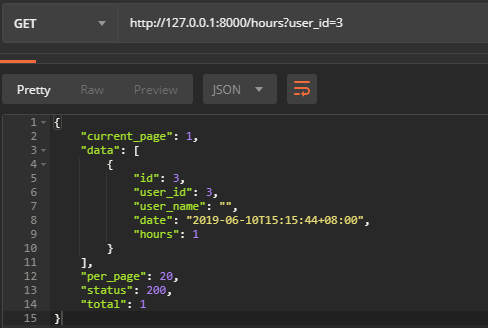
"per\_page": 20,

"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

}

],

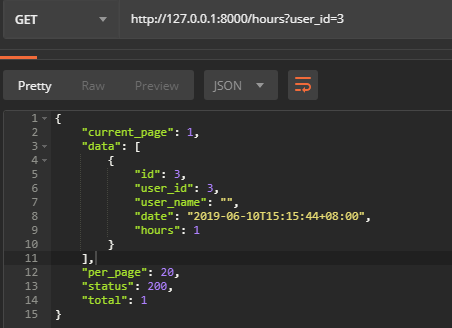
"per\_page": 20,

"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

}

],

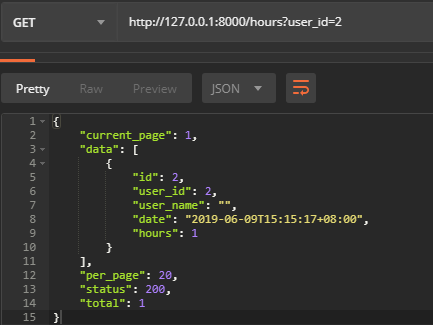
"per\_page": 20,

"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

}

],

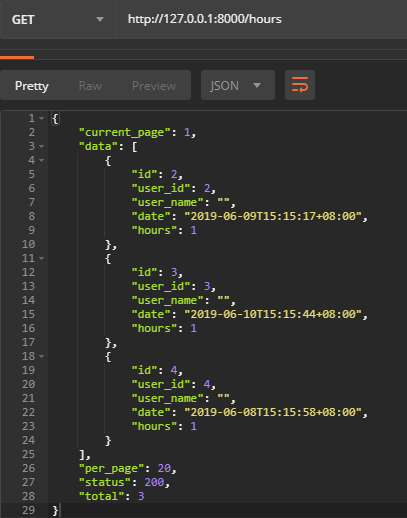
"per\_page": 20,

"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

},

......

],

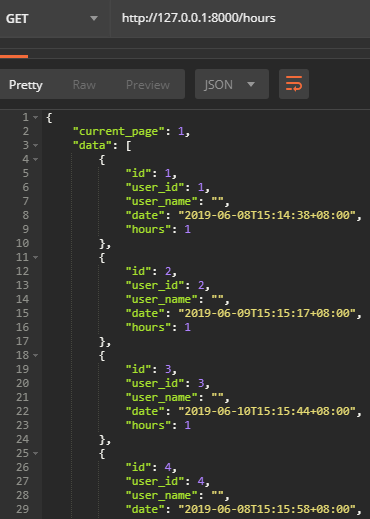
"per\_page": 20,

"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

}

],

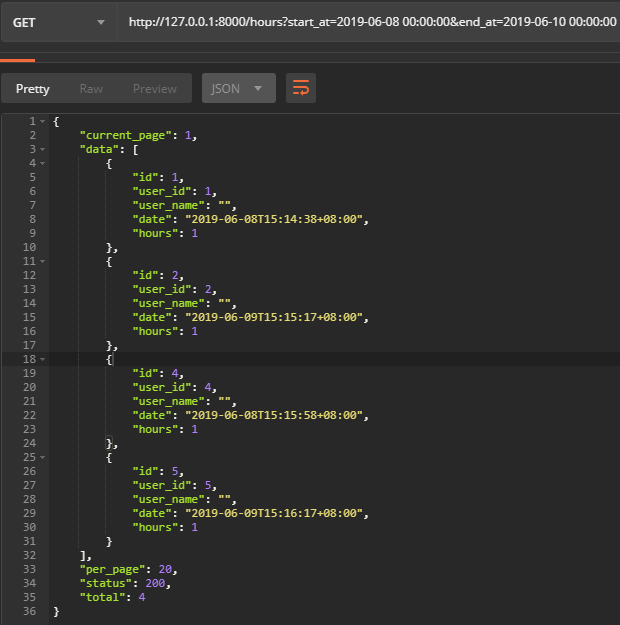
"per\_page": 20,

"status": 200,

"total": 4

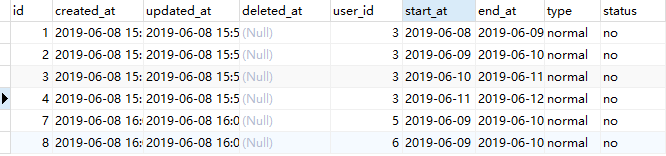
}

##### 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：

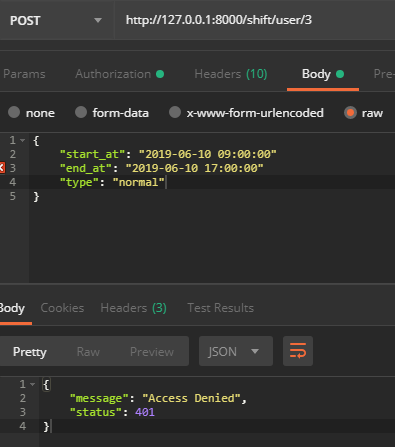
{

"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：

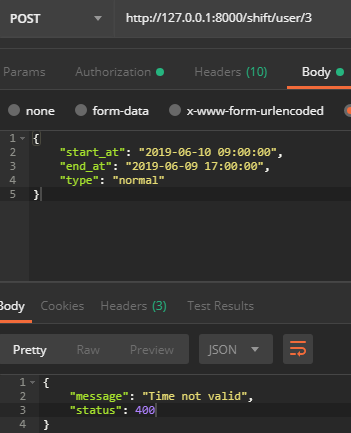
{

"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：

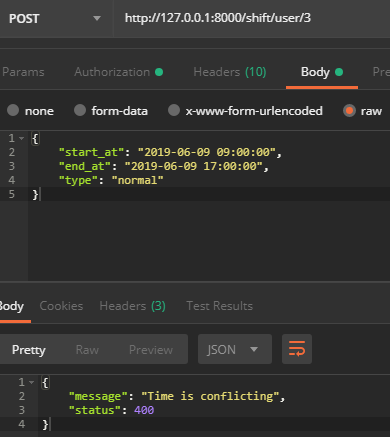
{

"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：

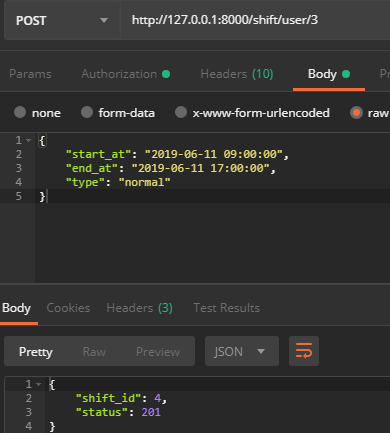
{

"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,

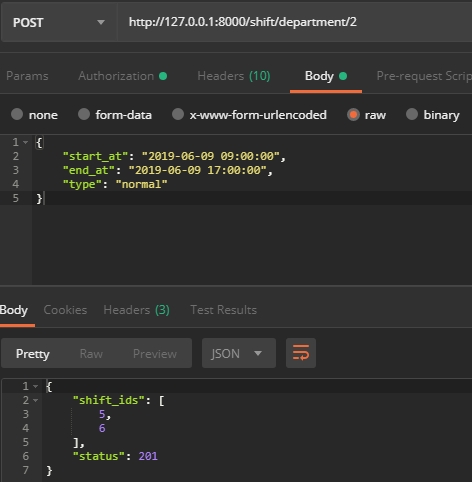
6

],

"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：

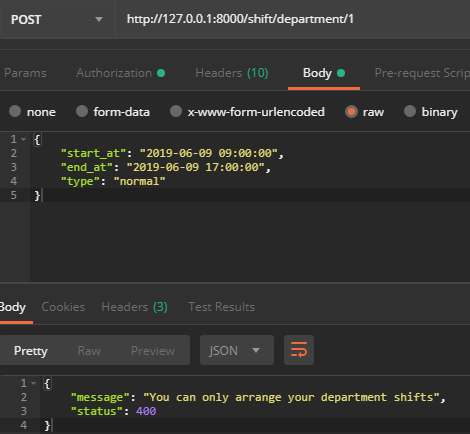
{

"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：

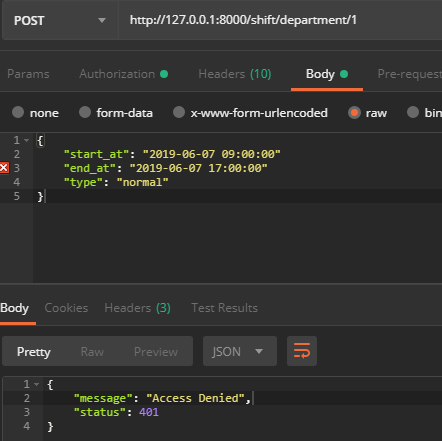
{

"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": ""

},

......

],

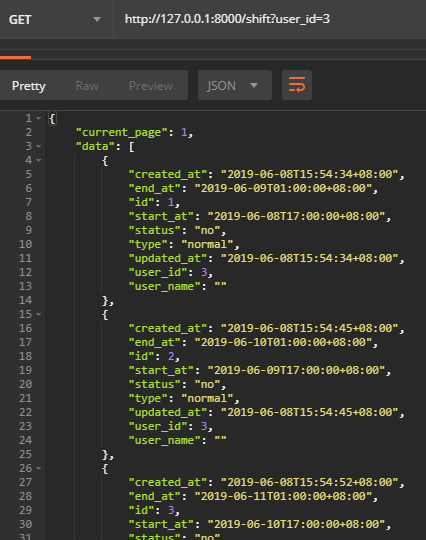
"per\_page": 20,

"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：

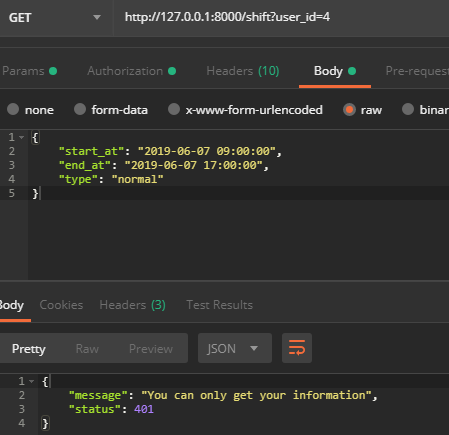
{

"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": ""

}

],

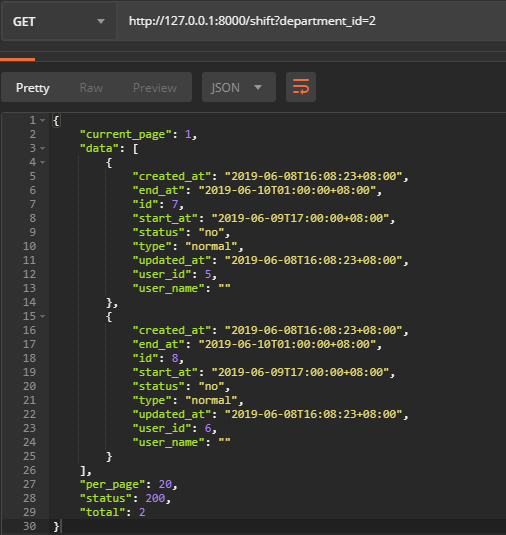
"per\_page": 20,

"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：

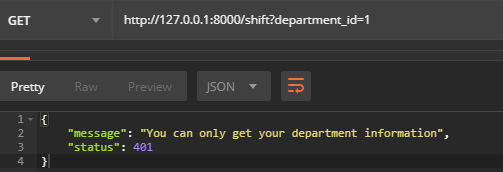
{

"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": ""

},

......

],

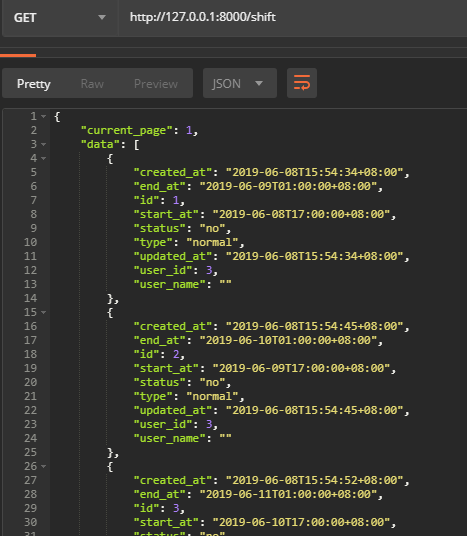
"per\_page": 20,

"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": ""

}

],

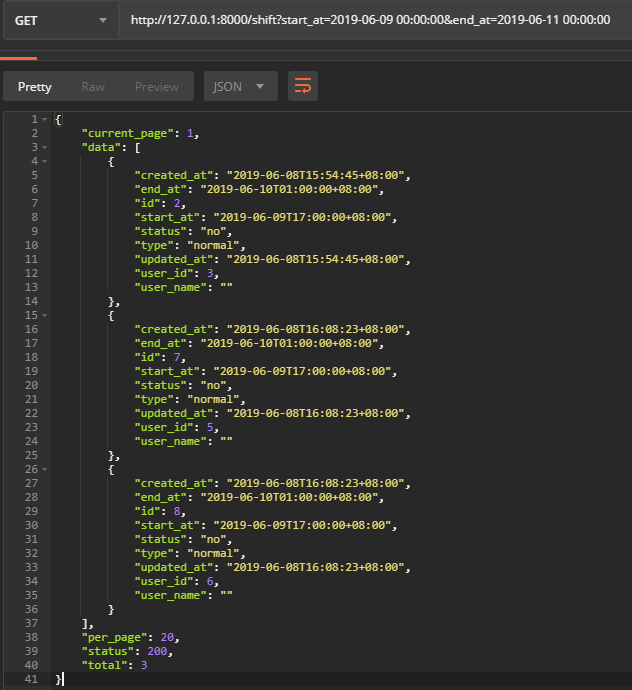
"per\_page": 20,

"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：

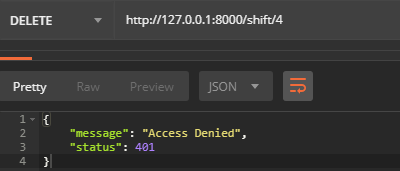
{

"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>

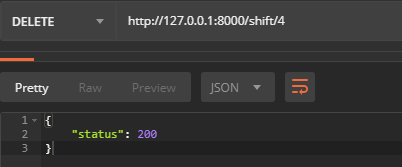
##### 预期输出：

{

"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：

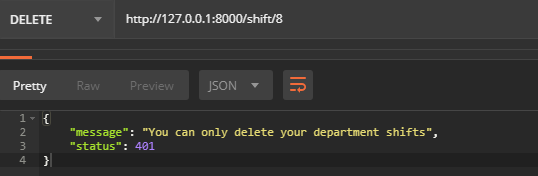
{

"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

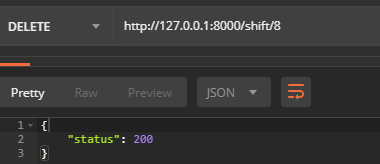
##### 预期输出：

{

"status": 200

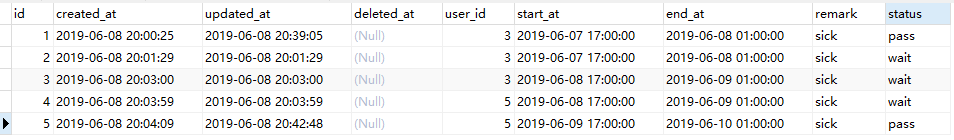
}

##### 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：

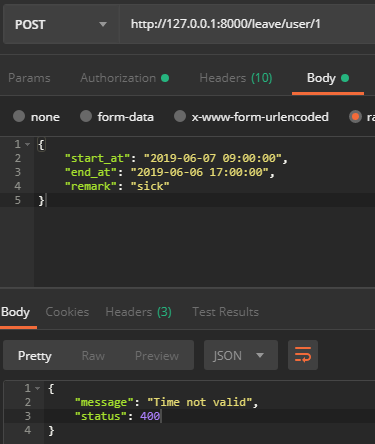
{

"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：

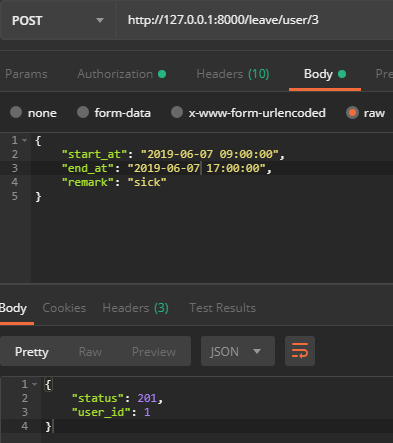
{

"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：

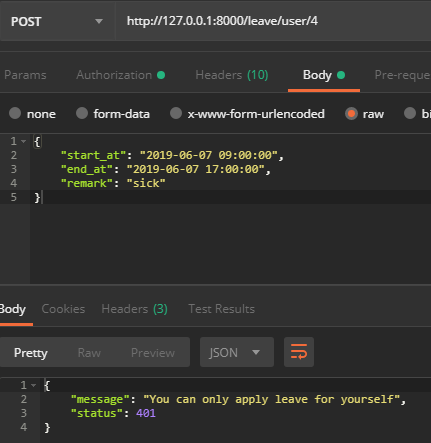
{

"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：

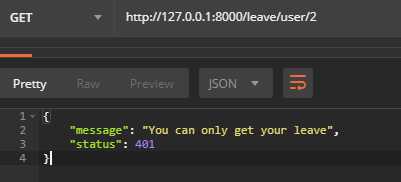
{

"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：

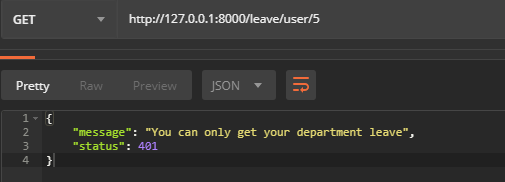
{

"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"

}

],

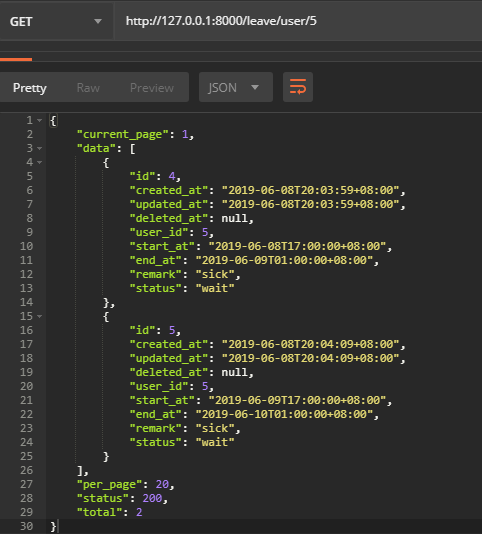
"per\_page": 20,

"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：

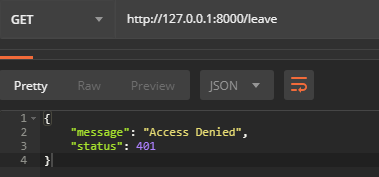
{

"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"

}

],

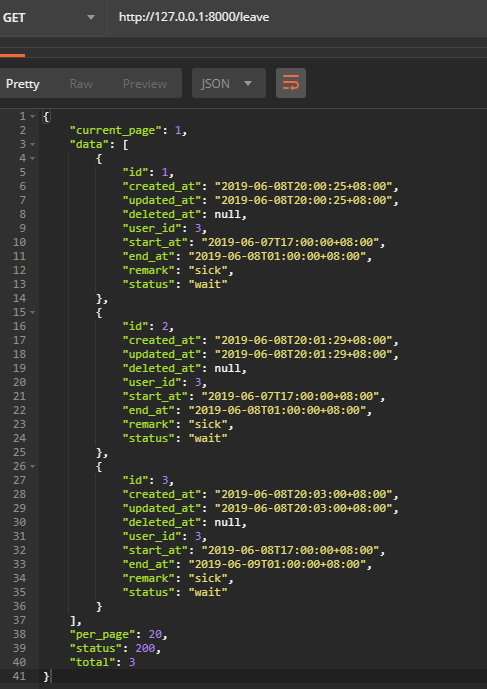
"per\_page": 20,

"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"

},

......

],

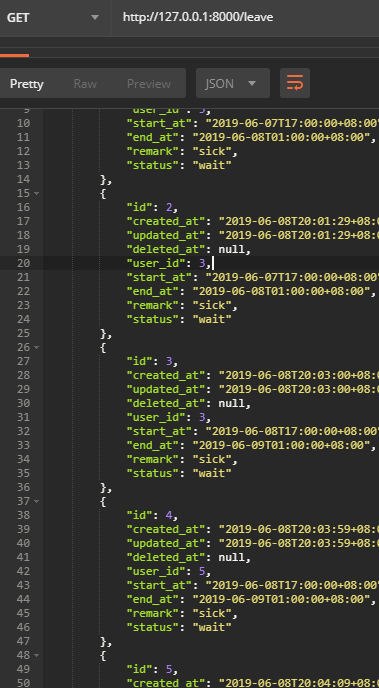
"per\_page": 20,

"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：

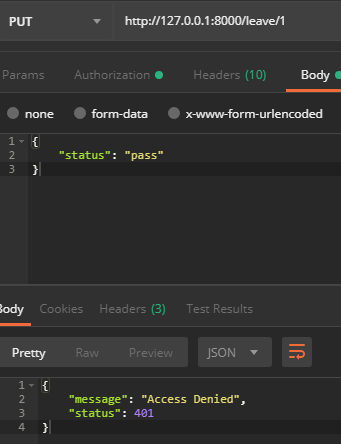
{

"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"

}

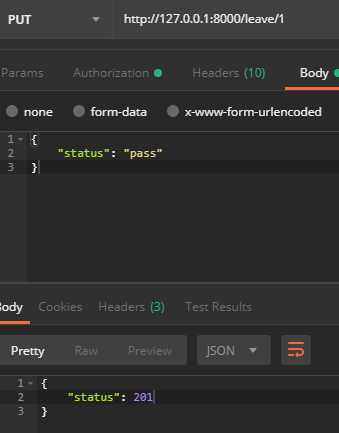
##### 预期输出：

{

"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：

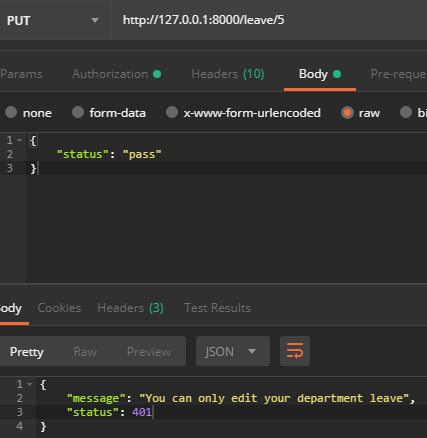
{

"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"

}

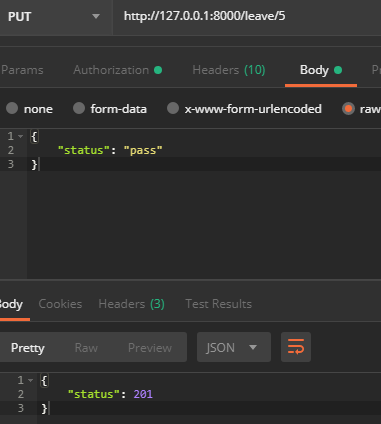
##### 预期输出：

{

"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

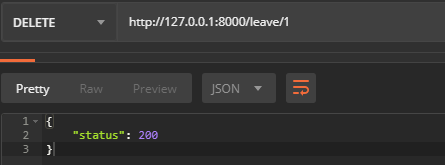
##### 预期输出：

{

"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：

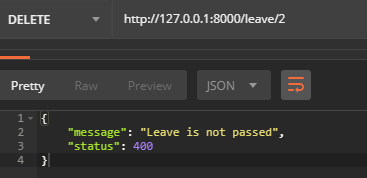
{

"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：

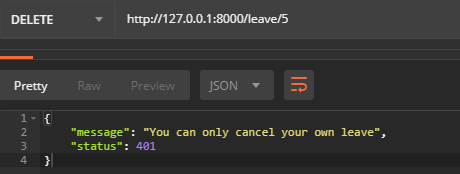
{

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

"status": 401

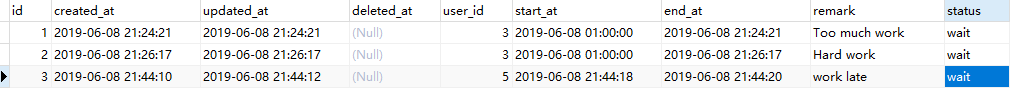
}

##### 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：

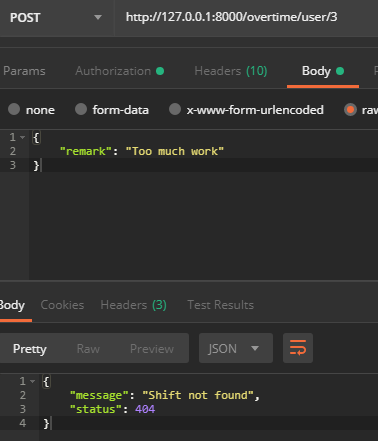
{

"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：

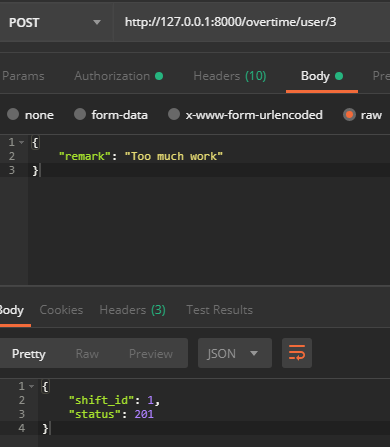
{

"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"

}

],

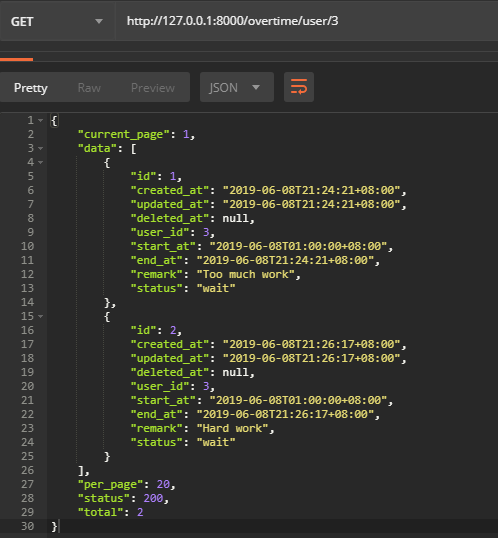
"per\_page": 20,

"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：

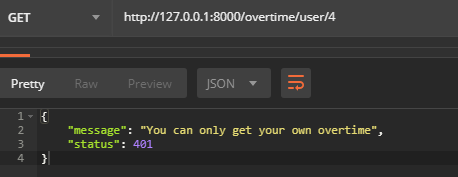
{

"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"

}

],

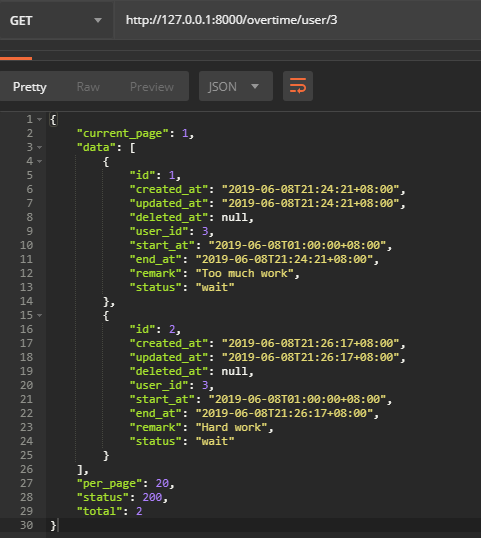
"per\_page": 20,

"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：

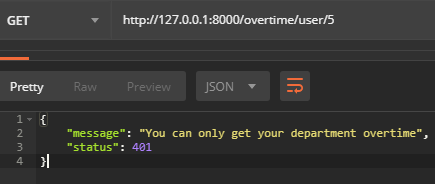
{

"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"

}

],

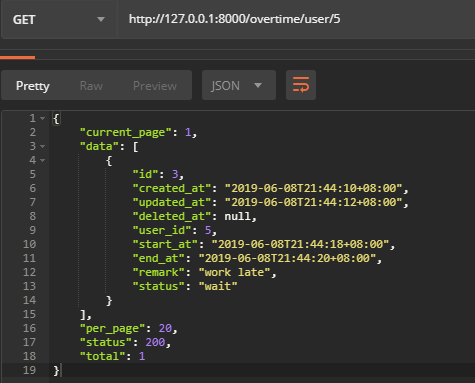
"per\_page": 20,

"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：

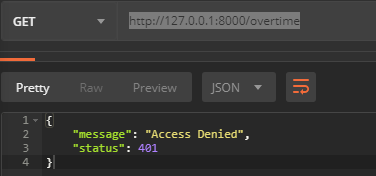
{

"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"

}

],

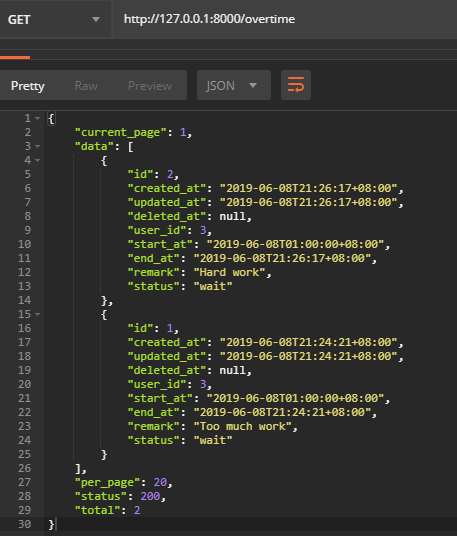
"per\_page": 20,

"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：

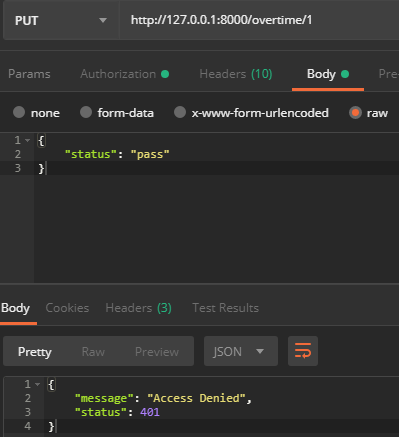
{

"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"

}

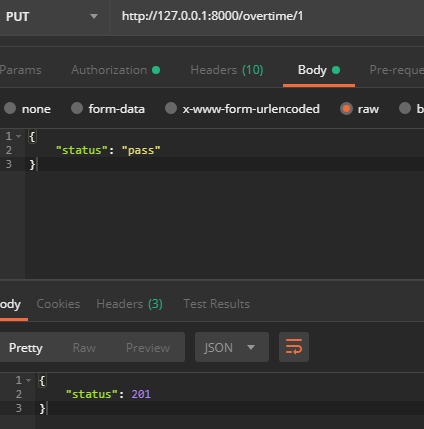
##### 预期输出：

{

"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：

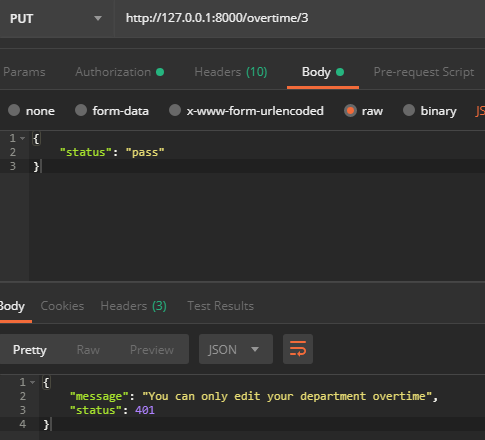
{

"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：**

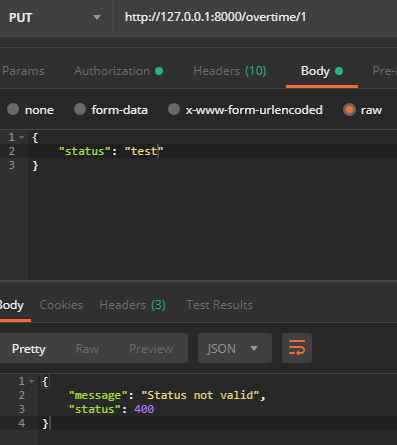
{

"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：

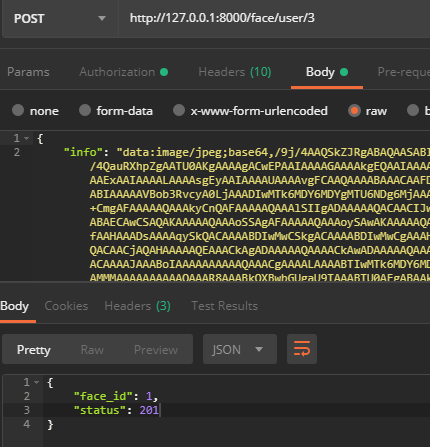
{

"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：

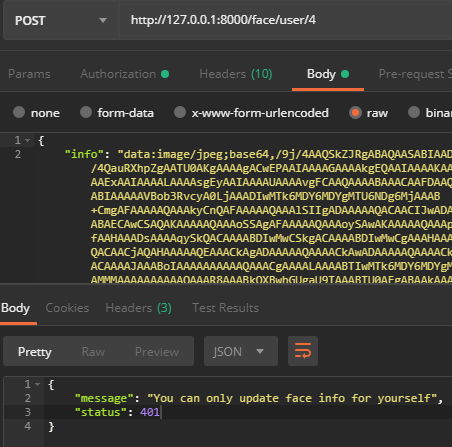
{

"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：

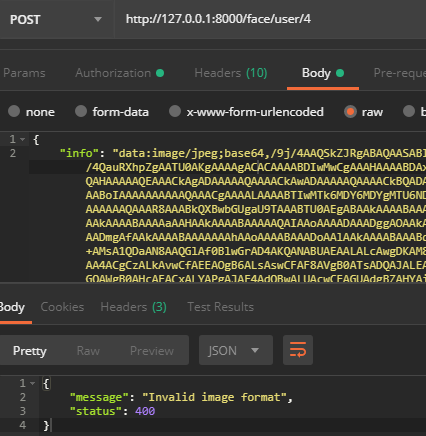
{

"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",

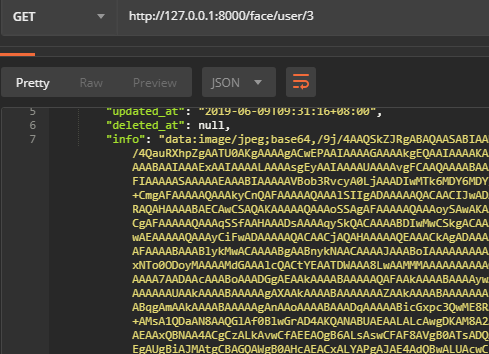
"deleted\_at": null,

"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：

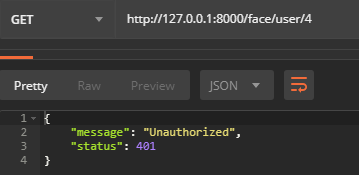
{

"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：

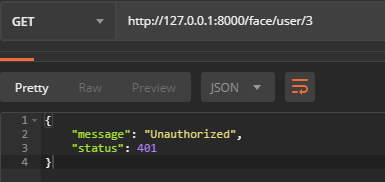
{

"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",

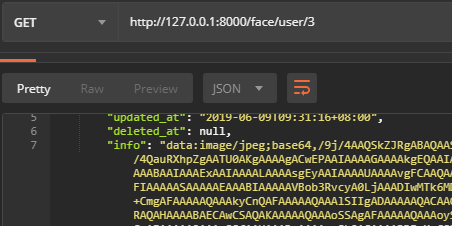
"deleted\_at": null,

"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：

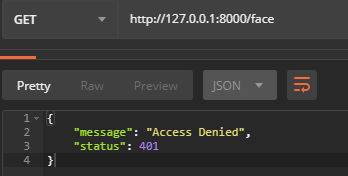
{

"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

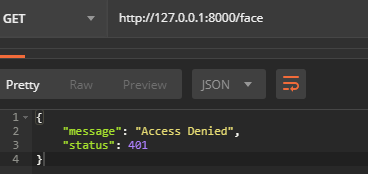
{

"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

}

}

],

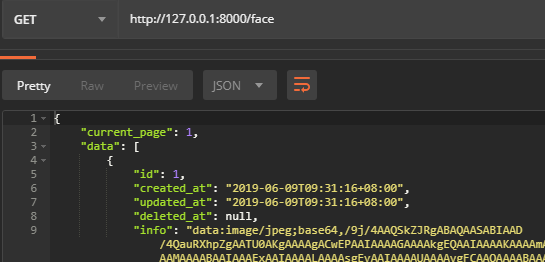
"per\_page": 20,

"status": 200,

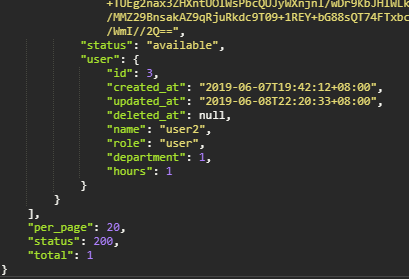
"total": 1

}

##### 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：

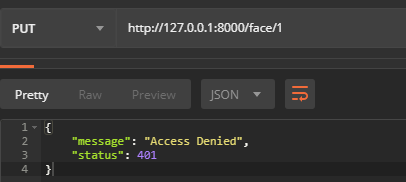
{

"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：

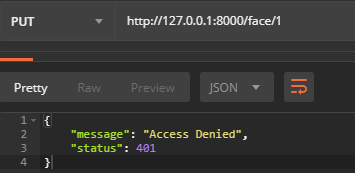
{

"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

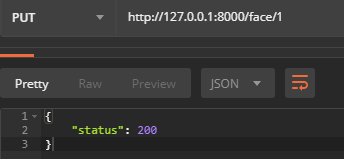
##### 预期输出：

{

"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",

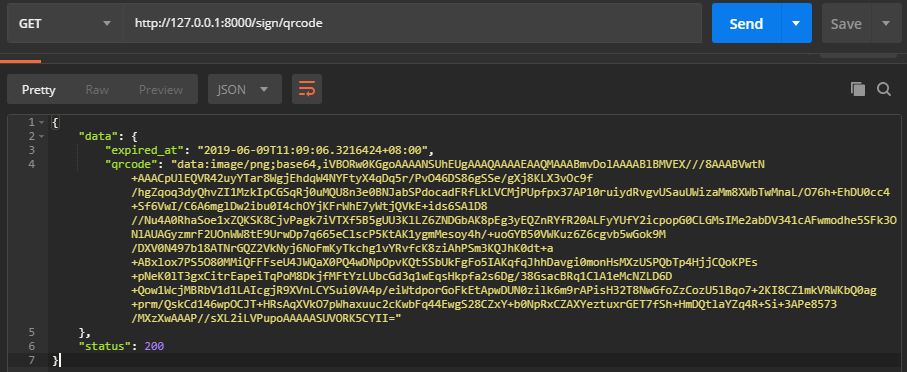
"qrcode": "data:image/png;base64,"

},

"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：

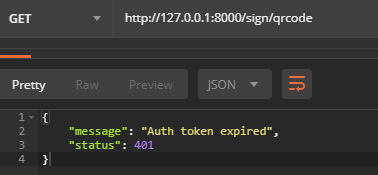
{

"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：

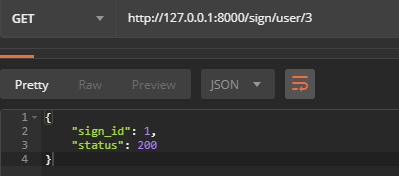
{

"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：

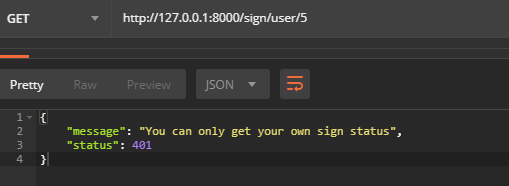
{

"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：

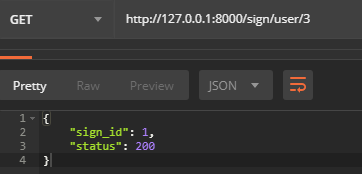
{

"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：

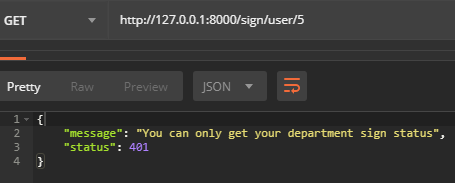
{

"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：

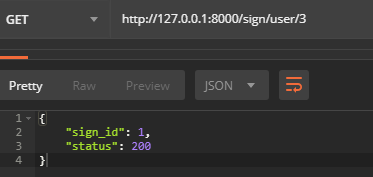
{

"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：

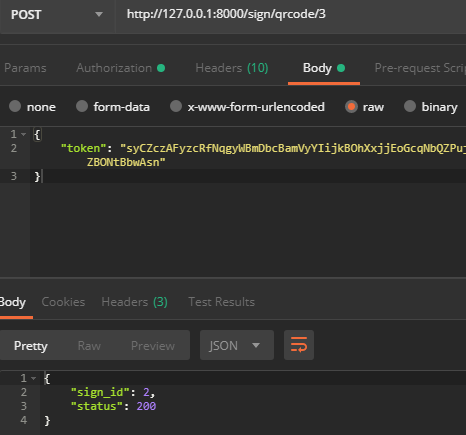
{

"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：

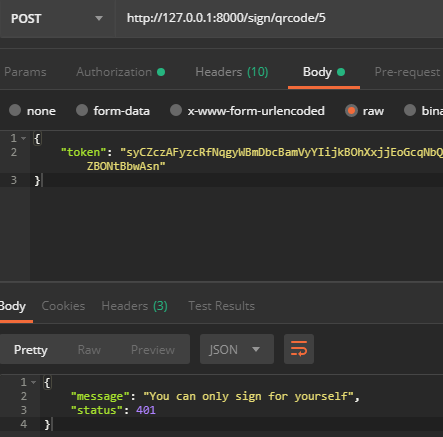
{

"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：

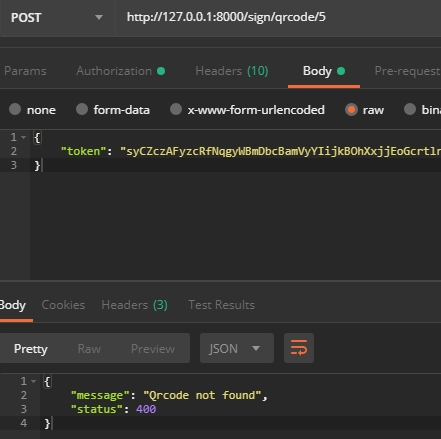
{

"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

}

}

],

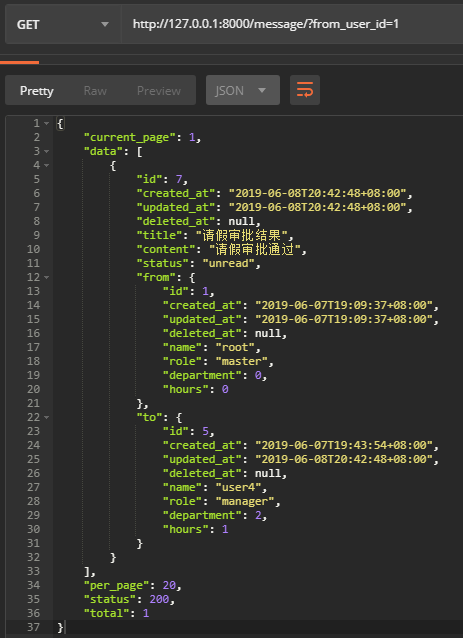
"per\_page": 20,

"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

}

}

],

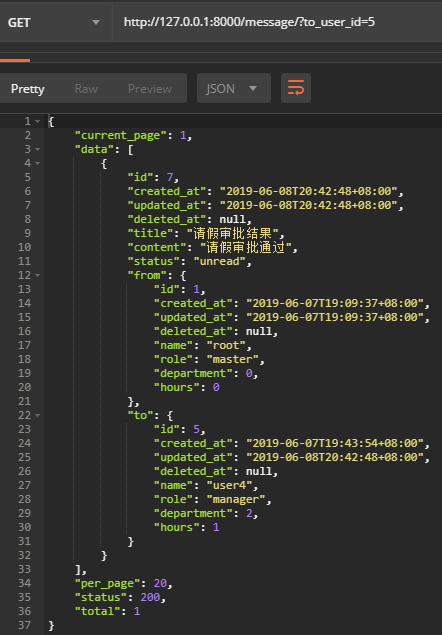
"per\_page": 20,

"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：

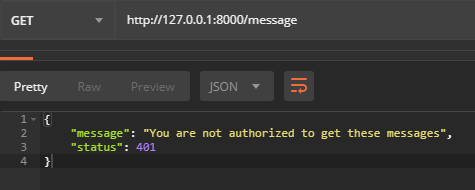
{

"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：

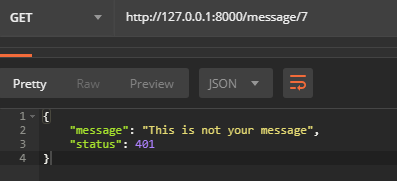
{

"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

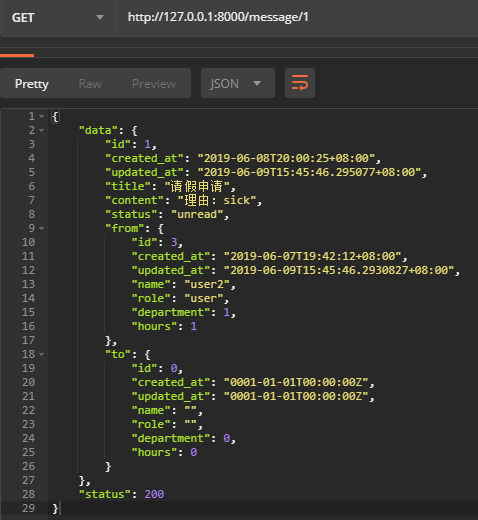
}

},

"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

}

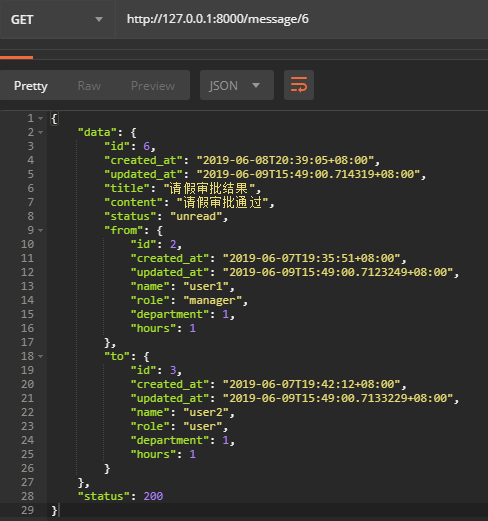
},

"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：

