前端提交给后端格式均使用json的数据格式，后端同样返回json数据

<http://127.0.0.1:8000/>是django本地服务器的ip，后端提供给前端的url接口均为

<http://127.0.0.1:8000/>\*\*\*/的格式

一、登陆：

前端🡪后端：

1.前端提供数据格式为：{“account”:\*\*\*, ”password”: \*\*\*,”account type”:\*\*\*}，例如账号为2017000001，密码为12345，账户类型为学生，则前端提供的数据应该为{“account”:“2017000001”,”password”:“12345”,”account type”:”student”}

2.后端返回的数据格式为{“result”: \*\*\* }，登陆成功返回{“result”:”success”}，密码错误返回{“result”:”fail”}。

3.后端提供url接口为：http://127.0.0.1:8000/login/

后端🡪数据库：

数据库提供三个存储过程，用于查询账户的密码。后端根据不同的账户类型调用不同的存储过程，给出的数据为：账号、密码；数据库返回的数据是该账户的密码，如果该账户不存在，返回查询结果为空的sql查询即可。

get\_student\_password( sno )

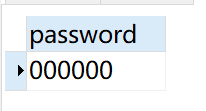
get\_teacher\_password( tno )

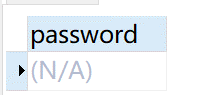
get\_temporary\_password( account )

输入参数为账号，调用方式如图：



输出为密码，格式如图：

查询成功

返回空值

二、查询：

查询分为

前端🡪后端：

1.前端提供数据格式为：{“account”:\*\*\*,”account type”:\*\*\*,“query type”:\*\*\*,”query info”:{“pno”:\*\*\*,“ano”:\*\*\*,”query account type”:\*\*\*}}，分别对应账户、账户类型、查询类型、查询信息，其中查询类型是为了区分教师查询自己和查询他人的劳务费。

分以下情况：

a.查询自己的劳务费信息，”query type” 设为 “w“ ，”query account type”需要与”account type”一致，”ano”要与”account”一致，也就是此时查询自己的劳务费信息

例：{“account”:”123,”account type”:”teacher”,“query type”:”w”,”query info”:{“pno”:”001”,“ano”：“123” , ”query account type” : ”teacher” }}是账号为123的老师查询自己参加的001号项目的劳务费情况

如果没有指定项目号，”pno”处置为””，即空字符串，例：

{“account”:”123,”account type”:”teacher”,“query type”:”w”,”query info”:{“pno”:””,“ano”：“123” , ”query account type” : ”teacher” }}是账号为123的老师查询自己参加的所有项目的劳务费情况

后端返回的数据格式均为{“1”：{“pname”:\*\*\*,”aname”:\*\*\*,”wage”:\*\*\*,”pay time”:\*\*\*},”2”: {“pname”:\*\*\*,”aname”:\*\*\*,”wage”:\*\*\*,”pay time”:\*\*\*}…}分别对应多条记录中的项目名称、账户名、劳务费、发放时间，如果未参加该项目或者没有相关信息，返回的是{“result”:”fail”}

b.如果账户类型为教师，查询的是他人的劳务费信息，则需指定”query type”=“p”，并且需要用户指定待查询账户的类型

例：

{“account”:”123,”account type”:”teacher”,“query type”:”p”,”query info”:{“pno”:”001”,“ano”：“456” , ”query account type” : ”student” }}

表示的是账号为123的教师查询自己管理的456号学生参加的001号项目的劳务费情况。

同样的，如果”pno”没有指定时，此处数据设为空字符串，如果”ano”没有指定，则”ano”和”query account type”处数据都置为空字符串

后端返回的数据同上述格式，如果该教师没有查询待查询人的操作权限，返回的是{“result”:”error”}，如果查询结果为空，返回的是{“result”:”fail”}

后端提供的接口为：<http://127.0.0.1:8000/query/>

后端🡪数据库：

数据库提供存储过程，后端提供的数据格式为账户、项目号、待查询账号、待查询账户类型。其中如果项目号未指定，则后端提供的数据为’#’，如果带查询账户未指定，则待查询账户和其类型均为’#’

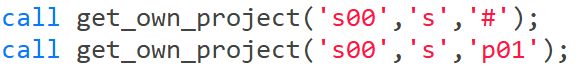
提供了两个存储过程，分别为查询参与项目、查询负责项目

1. 查询参与项目

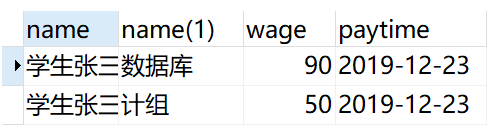
get\_own\_project（ sno/tno/account, type, pno ）

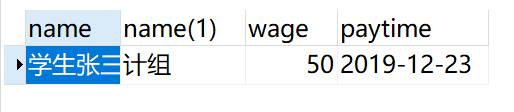
其中type表示查询账户类型，’t’表示教师，’s’表示学生，’tem’表示临时雇用人员；pno为项目号，pno为’#’表示不指定项目。

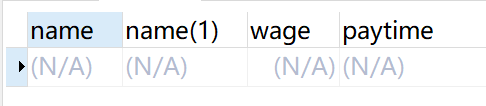
调用方式如图：



返回结果如图：

不指定项目

指定项目

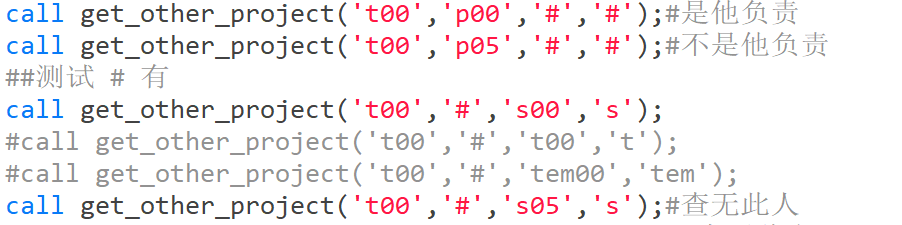
返回空值

（2）查询负责项目

get\_other\_project（ tno, pno, sno/tno/account , type ）

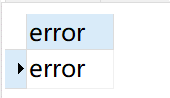
其中tno为负责人账号，sno/tno/account为待查询人账号，type为待查询人账号类型，pno和sno/tno/account为’#’时表示不指定。

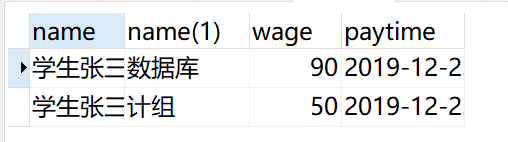
调用方式如图：

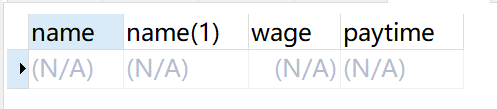


结果如图：

查询成功

该项目不由此教师负责，返回’error’

查询成功

返回空值

三、创建项目（仅提供给教师用户）：

前端🡪后端：

前端提供的数据格式为：{“pno”:\*\*\*,”leader”:\*\*\*,” ,”name”:\*\*\*funds”:\*\*\*,”start time”:\*\*\*,”end time”:\*\*\*,”funds for teacher”:\*\*\*,”funds for student”:\*\*\*,”funds for temporary”:\*\*\*}，数据中funds、funds for teacher、funds for student、funds for temporary都是浮点数，其他的均为字符串，其中项目号、项目名、管理人账户、项目经费、起始时间、终止时间是必填项目，经费比例是非必要的输入，如果用户没有给出，则前端不必给出。

例：

{“pno”:”123”,”leader”:”001”,”name”:”abc”,”funds”:100000,”start time”:”2019-01-01”,”end time”:”2019-01-02” }表示创建一个项目，并且不指定劳务费分配比例

{“pno”:”123”,”leader”:”001”,”name”:”abc”,”funds”:100000,”start time”:”2019-01-01”,”end time”:”2019-01-02” , ”funds for teacher”:0.1,”funds for student”0.4,”funds for temporary”:0.5}表示创建一个项目，并且指定分配比例为0.1, 0.4, 0.5

后端返回数据格式为：

如果插入成功，返回{“result”:”success”}

插入失败，返回{“result”:”fail”}

后端提供接口为

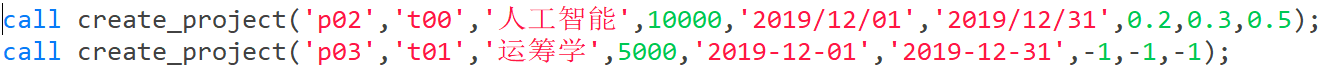
<http://127.0.0.1:8000/create_project/>

后端🡪数据库

后端提供一个存储过程实现项目表的插入，数据库提供数据与project表中数据一一对应，但是分配比例不一定给出，如果用户未给出分配比例，则后端提供的分配比例数据均为-1

create\_project( ) 括号里给出各个参数

调用方式如图：



如果给出的pno已存在，返回’error’