用户系统接口文档

用户系统接口共分为四类：创建(create)，查询(find)，修改(update)，删除(delete)。

以上四个功能简称为CFUD

**1)创建用户**

def create\_user(user\_list):

**函数功能：**

创建新用户

**函数参数：**

user\_list是一个元素为dict的list，形如[dict1,dict2,…..]

每个dict里的内容都是一位新用户的信息，包含的key : value对如下

“user\_name” : 用户登录名(string)

“real\_name” : 用户真实姓名(string)

“email” : 用户邮箱(string)

“password” : 登录密码(string)

“department” : 用户所在部门名字组成的list (list[string,string…..])

“tel” : 用户联系方式(string)

“status” : 用户状态(string)

“school\_id” : 学工号(string)

**函数返回值：**

创建成功返回新用户list，失败则抛出异常

**函数异常：**

IntegrityError : 用户名重复

FormatInvaidError : 邮箱、用户名、密码、联系方式其中之一的格式不正确

KeyError : key关键字不正确或者缺失

ValueError : 除user\_name, password, tel, school\_id以外的其他key对应的变量类型错误

TypeError : user\_name, password, tel, school\_id格式错误

**函数说明：**

1. 函数定义在accoun.interface.py里，FormatInvalidError同样定义在这个文件里

2. 使用IntegrityError，请import django.db

3. 新用户的权限默认为最低

4. 参数user\_list不能为空，且user\_list里的每一个dict都必须包含上述的所有的key-value对，且value不能为空，否则将返回上述定义的异常

**2)查询用户**

def find\_users(filt, exclude)

**函数功能：**

根据筛选条件查询用户

**函数返回值：**

满足filt里的条件且不包含exclude条件的用户组成的list，list可以为空

**函数参数：**

filt：用户需要包含的性质

exclude: 用户不能包含的性质

filt和exclude的格式要求完全一致，下面统一说明

参数是一个dict，可能包含的key：value对如下

“user\_name” : 用户登录名(string)

“real\_name” : 用户真实姓名(string)

“email” : 用户邮箱(string)

“department” : 用户所在部门的名字组成的list (list[string,string…..])

“tel” : 用户联系方式(string)

“status” : 用户状态(string)

“permission” : codename组成的list

“school\_id” : 学工号(string)

**函数异常:**

KeyError : key关键字错误

**函数说明：**

1.如果查找不到符合要求的用户，则返回空list，既[]

2.用户可以通过传入更复杂的key关键字来实现更加精细的查找功能，具体用法请参见

[https://docs.djangoproject.com/en/1.7/ref/models/querysets/#field-lookups](https://docs.djangoproject.com/en/1.7/ref/models/querysets/%23field-lookups)

**3)更新用户信息**

def update\_user(account\_id, update\_content)

**函数功能：**

修改用户信息

**函数返回值：**

修改成功返回修改后的对象，失败则抛出异常

**函数参数：**

account\_id：待修改信息的用户id，类型为int

update\_content: 更新信息的内容，类型为dict，可能包含的key-value对如下

“real\_name” : 用户真实姓名(string)

“email” : 用户邮箱(string)

“password” : 登录密码(string)

“department” : 用户所在部门的名字list (list[string,string…..])

“tel” : 用户联系方式(string)

“status” : 用户状态(string)

“school\_id” : 学工号(string)

**函数异常：**

ValueError : 除user\_name, password, tel, school\_id以外的其他key对应的变量类型错误

TypeError : user\_name, password, tel, school\_id格式错误

UserDoesNotExistError : user\_id对应的用户不存在

KeyError : key关键字错误

FormatInvalidError : 邮箱、密码、联系方式格式不正确

IntegrityError : key关键字包含id或者user\_name

**函数说明：**

1.用户的id和user\_name不可以修改，如果用户的update\_content.keys()中含有”id”或者”user\_name”关键字，程序会抛出IntegrityError错误

2.函数和UserDoesNotExistError, FormatInvalidError定义在interface.py里

**4)删除用户**

def delete\_user(user\_id)

**函数功能：**

删除用户

**函数返回值：**

用户删除成功则返回True，失败抛出异常

**函数参数:**

待删除用户的id(int)

**函数异常：**

UserDoesNotExistError : user\_id对应的用户不存在

**函数说明：**

1.函数和UserDoesNotExistError定义在interface.py里

**5)创建部门**

def create\_department(depart\_name)

**函数功能：**

创建新部分或者查找已存在部门

**函数返回值：**

已存在或者新创建的部分

**函数参数：**

部门名字(string)

**函数异常：**

无

**函数说明：**

1.如果部门不存在，则创建新部门后返回，否则找到后直接返回

6)检查格式

**def check\_style(pattern, stype)**

**函数功能：**

给定正则表达式和待检查文本，检查文本格式是否正确

**函数返回值：**

符合格式返回True，否则返回False

**函数参数：**

pattern: 正则表达式

text: 待检查文本

**函数异常：**

无

**函数说明：**

无

备注：

department相关的部分需要修改，可以暂时勉强使用