

API 手册——用户部分

用户部分

功能介绍

1. 用户注册
2. 用户登录
3. 用户详细信息提交
4. 修改用户详细信息
5. 修改密码
6. 用户登出

接口说明

1、功能：用户注册

URL: `http://127.0.0.1:8000/rest-auth/registration/`

Method: POST

Content:

```
{
    "username": "",
    "email": "",
    "password1": "",
    "password2": ""
}
```

Parameter: 无

测试记录:

一、正常情况:

Return: {
 "key": "d217cbb2e198af5a8baf7ddae60b638ffc655354"
}

返回 token，保持登录状态

二、输入错误:

1、空缺:

Return:

```
{
  "username": [
    "This field may not be blank."
  ],
  "password1": [
    "This field may not be blank."
  ],
  "password2": [
    "This field may not be blank."
  ]
}
```

2、密码不合法：//(每一个小分句都可能单独出现)

Return:

```
{
  "password1": [
    "This password is too short. It must contain at least 8 characters.",
    "This password is too common.",
    "This password is entirely numeric."
  ]
}
```

3、邮件格式不合法:

Return:

```
{
  "email": [
    "Enter a valid email address."
  ]
}
```

4、用户名格式不合法:

return:

```
{
  "username": [
    "Enter a valid username. This value may contain only letters, numbers, and @/./+/-/_ characters."
  ]
}
```

5、用户名超出长度: (邮箱和密码都没有长度限制)

Return:

```
{
  "username": [
    "Ensure this field has no more than 150 characters."
  ]
}
```

三、逻辑错误:

1、密码不一致：

Return:

```
{
  "non_field_errors": [
    "The two password fields didn't match."
  ]
}
```

2、重复用户名注册：

Return:

```
{
  "username": [
    "A user with that username already exists."
  ]
}
```

3、重复邮箱注册：

return:

```
{
  "email": [
    "A user is already registered with this e-mail address."
  ]
}
```

四、特殊情况：

1、当用户空缺且密码不合法时，只会报用户空缺

例如：Condition:

```
{"","","","abcdefgh","abcdefgh"}
```

Return:

```
{
  "username": [
    "This field may not be blank."
  ]
}
```

五、其他说明：

0. 前端需要对所有输入框中的非 `ascii` 部分（比如中文）进行排除

1. 用户名和邮箱不需要区分大小写

2. 在注册成功返回 `token` 之后，如果直接刷新页面会这样报错（重复登录测试），此处应该在测试 API 的页面有影响，但是调用的时候应该不会出现类似情况。

2、功能：用户登录

URL: <http://127.0.0.1:8000/rest-auth/login/>

Method: POST

Content:

```
{  
  "username": "",  
  "email": "",  
  "password": ""  
}
```

Parameter: 无

测试记录:

一、正常情况:

```
Return: {  
  "key": "2a73aec5b2a992074502fba1c0c4a78cc0f040a4"  
}
```

返回 token，保持登录状态

二、错误输入

1、空缺:

```
return:  
{  
  "password": [  
    "This field may not be blank."  
  ]  
}
```

2、未注册用户登录:

```
return:  
{  
  "non_field_errors": [  
    "Unable to log in with provided credentials."  
  ]  
}
```

3、错误密码登录:

```
return:  
{  
  "non_field_errors": [  
    "Unable to log in with provided credentials."  
  ]  
}
```

4、错误邮箱格式:

```
return:  
{  
  "email": [  
    "Enter a valid email address."  
  ]  
}
```

```
    ]  
}
```

5、填了邮箱但是空缺用户名:

condition: //空缺用户和已注册邮箱

```
{"","x1456776728@iClOud.cOm","fswxxz1456"}
```

return:

```
{  
  "non_field_errors": [  
    "Must include \"username\" and \"password\"."  
  ]  
}
```

三、其他说明

- 1、事实上这个模块中邮箱不会起到任何作用，只要格式别填错就不会报错；目前还没有设计通过邮箱来找回密码。
- 2、支持多终端登录，每个设备同一账号返回的 token 一样；
- 3、这个模块的 Username 没有做输入检查，如果具体实现里有涉及直接用用户名来进行操作的（比如查找文件，查找密码等），简易前端增加特殊符号检查（比如用户输入~/或..在查找文件时都可能引发严重错误）

3、功能：用户详细信息提交

URL: http://127.0.0.1:8000/api/create_users_info/

Method: POST

Content:

```
{  
  "company": "",  
  "tel": "",  
  "address": ""  
}
```

Parameter: 无

测试记录:

一、正常情况

Return: [

```
{  
  "id": 1,  
  "company": "rinc",  
  "tel": "1234567",  
  "address": "asasasasa",
```

```

        "user": {
            "id": 3,
            "password":
"pbkdf2_sha256$36000$MO32aGo4yQuK$pkMAJ4Y0IPJ+bEarMs2j6CBX4n8x9Y376S1Nv1yZ1NI=",
            "last_login": "2017-07-06T05:11:24.945000Z",
            "is_superuser": false,
            "username": "zhang",
            "first_name": "",
            "last_name": "",
            "email": "asasasaa111@163.com",
            "is_staff": false,
            "is_active": true,
            "date_joined": "2017-07-06T05:11:24.481000Z",
            "groups": [],
            "user_permissions": []
        }
    }
]

```

二、重复提交

每一个注册用户只能对应一个用户详细信息，再次提交用户详细信息就会出现如下错误提示：

```

{
    "detail": "This User Detail Info has been existed!"
}
/* 此处建议前端做输入检查 */

```

三、输入错误：

Return:

```

{
    "company": [
        "Ensure this field has no more than 100 characters."
    ],
    "tel": [
        "Ensure this field has no more than 20 characters."
    ],
    "address": [
        "Ensure this field has no more than 100 characters."
    ]
}

```

4、功能：显示和修改用户详细信息

URL: http://127.0.0.1:8000/api/users_display/

Method: GET PUT

Parameter: 无

测试记录：

一、正常情况：

```
{
  "id": 1,
  "company": "rinc",
  "tel": "1234567890",
  "address": "江苏省南京市仙林大道 163 号南京大学计算机系",
  "user": {
    "id": 19,
    "password":
"pbkdf2_sha256$36000$ACP2ZYKS5Xx4$DgNrpQxd77AH6g7zXwlW7zyXrBOR8+5HK6bvt71ID8k=",
    "last_login": "2017-07-17T10:56:33.601000Z",
    "is_superuser": false,
    "username": "zhangxu",
    "first_name": "",
    "last_name": "",
    "email": "takeyourminds@qq.com",
    "is_staff": false,
    "is_active": true,
    "date_joined": "2017-07-17T10:56:32.634000Z",
    "groups": [],
    "user_permissions": []
  }
}
```

二、不存在详细信息：

```
{
  "detail": "Not found."
}
```

三、未登录：

```
{
  "detail": "Authentication credentials were not provided."
}
```

四、输入超出长度：

```
return:
{
  "company": [
    "Ensure this field has no more than 100 characters."
  ]
}
```

```

    ],
    "tel": [
        "Ensure this field has no more than 20 characters."
    ],
    "address": [
        "Ensure this field has no more than 100 characters."
    ]
}

```

/* 有长度检查，没有对输入的字符合法性进行检查（特殊的是，这个模块应该允许输入中文，但是建议警惕特殊符号） */

5、功能：修改密码

URL: http://127.0.0.1:8000/rest-auth/password/change/

Method: POST

Content:

```

{
    "new_password1": "",
    "new_password2": ""
}

```

Parameter: 无

测试记录：

一、正常情况

```

Return: {
    "detail": "New password has been saved."
}

```

二、错误密码：

return:

```

{
    "old_password": [
        "Invalid password"
    ]
}

```

三、新密码合法性检查（报错情况和注册时一样）：

return:

```

{
    "new_password2": [

```



```
        "This password is too short. It must contain at least 8 characters.",
        "This password is too common.",
        "This password is entirely numeric."
    ]
}
```

四、两次新密码不一致:

```
return:
{
    "new_password2": [
        "The two password fields didn't match."
    ]
}
```

五、多点登录问题

在修改密码成功后，所有设备上的用户会自动被登出,这时候再次试图改密码会出现这个

```
return:
{
    "detail": "Authentication credentials were not provided."
}
```

六、其他说明

密码修改成功后，将会自动执行登出命令，此时 token 被清除，用户需要进入登录界面使用新的密码登录。

6、功能：用户登出

URL: <http://127.0.0.1:8000/rest-auth/logout/>

Method: GET POST

Parameter: 无

测试记录

一：正常情况

```
Return:
{
    "detail": "Successfully logged out."
}
```

二：其余说明

- 1、用户登出 Test:无论是否有正在登录的用户，返回值都是描述的那样，功能正常
- 2、多点登录的时候，logOut 只会登出本设备的用户