用户部分API

功能：用户注册

URL: http://127.0.0.1:8000/rest-auth/registration/
Method：post

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

正常情况：

```
Return：{
    "key": "d217cbb2e198af5a8baf7ddae60b638ffc655354"
}
```

返回token，保持登录状态

Parameter：无

输入错误：

空缺：Return:

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

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."

                    ]
                }
```

邮件格式不合法：

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

用户名格式不合法：

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

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."
          ]
        }
```

逻辑错误：

密码不一致：

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

重复用户名注册：

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

重复邮箱注册：

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

特殊情况：

当用户空缺且密码不合法时，只会报用户空缺

例如：Condition:

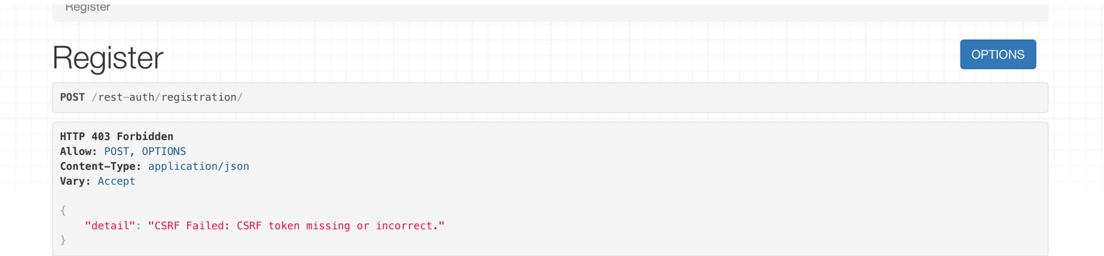
```
        {“”,””,”abcdefgh”,”abcdefgh”}
    Return:
        {
          "username": [
            "This field may not be blank."
          ]
        }
```

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）



```
POST /rest-auth/registration/
```

```
HTTP 403 Forbidden
Allow: POST, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "detail": "CSRF Failed: CSRF token missing or incorrect."
}
```

功能：用户登录

URL: http://127.0.0.1:8000/rest-auth/login/
Method：post
正常情况：

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

返回token，保持登录状态
Parameter：无

空缺：

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

未注册用户登录：

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

错误密码登录：

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

错误邮箱格式：

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

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\"."
                    ]
                }
```

/*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；
  支持多终端登录，返回的token一样；

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） */

功能：用户详细信息提交

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

Method：post

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

```
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": []
        }
    }
]
```

Parameter：无

重复提交：

```
        [
            "Detail Info has existed!"
        ]
```

/ * 同样地，建议前端做输入检查 * /

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

Return：

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": []
    }
}
```

Parameter：无

不存在详细信息:

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

未登录:

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

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） */

功能：修改密码

```
URL: http://127.0.0.1:8000/rest-auth/password/change/
Method：post
{
    "new_password1": "",
    "new_password2": ""
}
Return：{
    "detail": "New password has been saved."
}
Parameter：无
```

错误密码：

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

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."
                    ]
                }
```

两次新密码不一致：

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

/* 值得注意的是，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."
         }

 */
```

功能：用户登出

URL: http://127.0.0.1:8000/rest-auth/logout/
Method：get or post
Return：

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

Parameter：无

```
/* 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
  * logOut只会登出本设备的用户
  */
```