

API文档

- API文档
 - **UserManagement**
 - token的使用方式
 - /user/signup
 - /user/login
 - /user/token_test

UserManagement

此模块主要用于进行用户的登录、注册、以及token的验证工作

现阶段用户名以及密码传输均采用明文传输，后续会考虑使用rsa加密传输

token的使用方式

拿到token后，使用方法为在http请求的header中添加一项(此方法手动构建header，可以考虑其他方法)

```
{
  "Authorization": "basic " + base64(token+":") # 注意basic后有一个空格，token后面
  需要添加一个冒号再做base64加密
}
```

如果token验证通过，服务器返回

```
{
  state: "success"
}
```

否则，服务器会返回状态码401(UNAUTHORIZED)，表明失效

对于后端任意函数，如果要求只有登录状态才能使用，须在函数定义前加

```
@auth.login_required
def example():
    # do something
```

这样就会自动使用上述token验证机制

/user/signup

此路径用于进行用户注册

支持Get方法和Post方法

输入：

```
{
  username: String, # 用户名
  password: String # 密码
}
```

输出：

成功时：

```
{
  state: "success"
}
```

失败时：

```
{
  state: "failed",
  description: String # 给出一个错误原因
}
```

/user/login

此路径用于进行用户登录

支持Get方法和Post方法

输入：

```
{
  username: String, # 用户名
  password: String # 密码
}
```

输出：

成功时：

```
{
  state: "success",
  UserId: String, # 用户ID, 请务必在本地保存, 后续需要使用
}
```

```
token: String # 后续操作使用的token  
}
```

失败时:

```
{  
  state: "failed",  
  description: String # 给出一个错误原因  
}
```

/user/token_test

用于验证token是否还有效。如果token无效需要重新登录获取token