[POST] user/register
Register users

Request Param
{
    String username,
    String password,
    String email
}
return type: String
If the username already exists, the system returns "Name is existed".
If the Email already exists, the system returns "Email is existed".
If successful, return “succeed”
If failed, return "failed"

[POST] user/login
Login

Request Param

{
    String username,
    String password
}
return type: String
If successful, return authority [‘teacher’, ‘student’, ‘user’, ‘admin’] and a cookie named “Auth”
If no user is found or the password is incorrect, the system will return "failed"

[GET] user/getusers
Get all user information, this returns a list of User entities. Return in JSON format

{
    String id,
    String username,
    String email,
    String img,
    String introduction
}

No request params are required
Administrator access is required
Determine permissions based on the cookie "Auth" carried in the request header
If the authentication fails, return "unauthorized"

[GET] user/getuser/{id}
Obtain information about a user by ID, return in JSON format

{
String id,
String username,
String email,
String img,
String introduction
}
Img will provide a string image URL

[POST] user/setauthority
Set the permissions of an account to [‘teacher’, ‘student’, ‘user’, ‘admin’].

Request Param

{
String username,
String authority
}

Administrator access is required
Determine permissions based on the cookie "Auth" carried in the request header
If the authentication fails, return "unauthorized"
If authority is not equal to [‘teacher’, ‘student’, ‘user’, ‘admin’], return "Illegal type"
If successful, return “succeed”
If failed, return "failed"

[POST] user/setPassword
Set password

Request Param
{
String password
}
Determine permissions based on the cookie "Auth" carried in the request header
If the user is not logged in or user does not exist, return "unauthorized"
If successful, return “succeed”
If failed, return "failed"

[POST] user/uploadPic
Uploading user profile photo
Request Param
{
MultipartFile image
}
Determine permissions based on the cookie "Auth" carried in the request header
If the authentication fails, return "unauthorized"
If successful, return “succeed”
If failed, return "failed"

[GET] user/getimg/{id}
Get user profile photo
{id} is the user ID

If successful, return picture
If failed, return "failed"
If picture doesn’t exist, return " Not found"

[POST] user/updateIntroduction

Update the user's self-introduction

```
{
    String introduction
}
```

Determine permissions based on the cookie "Auth" carried in the request header
If the authentication fails, return "unauthorized"
If successful, return “succeed”
If failed, return "failed"