# “使用微信扫一扫登录”管理平台开发实战

目录

[第1课、 课程来源 2](#_Toc112951591)

[第2课、 创建一个Web Api 2](#_Toc112951592)

[第3课、 微信公众号网页授权（1） 2](#_Toc112951593)

[第4课、 微信公众号网页授权（2） 3](#_Toc112951594)

[第5课、 微信公众号网页授权示例 3](#_Toc112951595)

[第6课、 微信公众号网页授权的限制与突破 5](#_Toc112951596)

[第7课、 仿造一个网页授权服务器 5](#_Toc112951597)

[第8课、 为页面授权服务器添加数据缓存功能 6](#_Toc112951598)

[第9课、 网页授权的代理服务器 6](#_Toc112951599)

[第10课、 网页授权的代理服务器的实现 7](#_Toc112951600)

[第11课、 对现有项目进行重构 7](#_Toc112951601)

[第12课、 将项目发布到Windows Server 2022的IIS服务器 7](#_Toc112951602)

[第13课、 在IIS服务器上启用HTTPS 7](#_Toc112951603)

[第14课、 将项目发布到Linux服务器 8](#_Toc112951604)

[第15课、 使用Nginx来发布Web 8](#_Toc112951605)

[第16课、 让项目以服务器的方式在Linux系统中运行 9](#_Toc112951606)

[第17课、 在Nginx中启用HTTPS(1) 10](#_Toc112951607)

[第18课、 在Nginx中启用HTTPS(2) 11](#_Toc112951608)

[第19课、 “公众号网页授权”与“网站应用微信登录” 12](#_Toc112951609)

[第20课、 域名、ICP备案及SSL证书 13](#_Toc112951610)

[第21课、 在微信开放平台中创建网站应用 14](#_Toc112951611)

[第22课、 “微信扫一扫登录”的实现 14](#_Toc112951612)

## 课程来源

* ASP.NET Core 6 Web Api
* 网站接入

## 创建一个Web Api

课程源码

[https://github.com/**zmrbak**/WxOAuthManager](https://github.com/zmrbak/WxOAuthManager)

## 微信公众号网页授权（1）

如果用户**在微信客户端中访问第三方网页**，公众号可以通过微信网页授权机制，来获取用户基本信息，进而实现业务逻辑。

* **在微信客户端中访问**
* **第三方网页**
* 获取用户基本信息

网页授权回调域名

* + 必须是全域名： [www.qq.com](http://www.qq.com)
  + 不添加访问协议
  + 不添加端口号
  + 全域名下的任意路径文件都可以进行OAuth2权鉴认证。

关于网页授权的两种 scope 的区别

* + snsapi\_base
    - 获取进入页面的用户的 openid
    - 静默授权并自动跳转到回调页
    - 用户感知：直接进入了回调页（往往是业务页面）
  + snsapi\_userinfo
    - 获取进入页面的用户的 openid
    - 获取用户的基本信息
    - 这种授权需要用户手动同意
    - 在授权后获取该用户的基本信息

网页授权流程：

* 引导用户进入授权页面同意授权，获取code
* 通过 code 换取网页授权access\_token

引导客户获取code的URL

public string Get()

{

var AppId = "wx520c15f417810387";

var RedirectUri = "https://chong.qq.com/php/index.php?d=&c=wxAdapter&m=mobileDeal&showwxpaytitle=1&vb2ctag=4\_2030\_5\_1194\_60";

var Scope = "snsapi\_base";

var **State** = "123";

var url = $"https://open.weixin.qq.com/connect/oauth2/authorize?appid={AppId}&redirect\_uri={HttpUtility.UrlEncode(RedirectUri)}&response\_type=code&scope={Scope}&state={State}#wechat\_redirect";

return url;

}

用户同意授权后

* 跳转至 redirect\_uri/?code=**CODE**&state=**STATE**。
* code作为换取access\_token的票据
* 每次用户授权带上的 code 将不一样
* code只能使用一次
* 5分钟未被使用自动过期。

通过 code 换取网页授权access\_token

<https://api.weixin.qq.com/sns/oauth2/access_token?appid=APPID&secret=SECRET&code=CODE&grant_type=authorization_code>

URL: <http://127.0.0.1/abc.aspx?code=063jPmll2aKIE94IdNnl2PiwT60jPmlu&state=123>

https://api.weixin.qq.com/sns/oauth2/access\_token?appid=wx6dbd50066c719ab7&secret=5231ed39553611b4f44edf1a94880ec3&code=063jPmll2aKIE94IdNnl2PiwT60jPmlu&grant\_type=authorization\_code

{

"access\_token": "59\_OZ1ie1M8BInD\_JjhcZMIB8yUenF-ZSmIACQhbK7KV-3gtiOkLkEVF7lzkL6TT1i09OiWXiOBRwRMN6Q9WLxO7bkftiUdeWsmVk1sMZJaJAg",

"expires\_in": 7200,

"refresh\_token": "59\_Dom9qM0uI0pnFXG6L3Oxkf57-eLbHjOQPp8bkCPUJfy4W2VE6f\_pLiDAwMtlI-yVu5Y7k7QY\_le4FN5GzchuzLdwKpeuZkcYktXHJZPq0Cc",

"**openid**": "oKhgkuM3QGSu3LROjP9FLcDi5DMg",

"scope": "snsapi\_base"

}

## 微信公众号网页授权（2）

## 微信公众号网页授权示例

<http://localhost:5188/api/WxOAuth2/063ivXFa1HHZFD09iFGa1HAPl20ivXF3>

http://127.0.0.1/abc.aspx?code=063ivXFa1HHZFD09iFGa1HAPl20ivXF3&state=123

[**http://localhost:5188/api/WxOAuth2?**code=0338yj0w34Vo0Z2Oz54w3ND9ri38yj05&state=123](http://localhost:5188/api/WxOAuth2?code=0338yj0w34Vo0Z2Oz54w3ND9ri38yj05&state=123)

<http://127.0.0.1/abc.aspx?code=8b77d895-01f3-45b8-9fc5-9b560aa2d629&state123>

http://127.0.0.1/abc.aspx?code=20027d6b351d43a0ab3648d7ede50b6d&state=123

"profiles": {

"WxOAuthManager": {

"commandName": "Project",

"dotnetRunMessages": true,

"launchBrowser": true,

"launchUrl": "swagger",

"applicationUrl": "http://192.168.240.232:5188",

"environmentVariables": {

"ASPNETCORE\_ENVIRONMENT": "Development"

}

},

"IIS Express": {

"commandName": "IISExpress",

"launchBrowser": true,

"launchUrl": "swagger",

"environmentVariables": {

"ASPNETCORE\_ENVIRONMENT": "Development"

}

}

}

"WeChatMp": {

"AppId": "wx6dbd50066c719ab7",

"AppSecret": "5231ed39553611b4f44edf1a94880ec3",

"RedirectUri": "http://192.168.240.232:5188/api/WxOAuth2"

}

**http://192.168.240.232:5188/api/WxOAuth2/123.aspx**

[HttpGet]

[Route("/api/[controller]/123.aspx")]

public IActionResult Get()

{

var Scope = "snsapi\_base";

var State = "123";

var url = $"https://open.weixin.qq.com/connect/oauth2/authorize?appid={configuration["WeChatMp:AppId"]}&redirect\_uri={HttpUtility.UrlEncode(configuration["WeChatMp:RedirectUri"])}&response\_type=code&scope={Scope}&state={State}#wechat\_redirect";

return Redirect(url);

}

**http://192.168.240.232:5188/api/WxOAuth2**

[HttpGet]

public string Get(string code,string state)

{

var Code = code;

var url = $"https://api.weixin.qq.com/sns/oauth2/access\_token?appid={configuration["WeChatMp:AppId"]}&secret={configuration["WeChatMp:AppSecret"]}&code={Code}&grant\_type=authorization\_code";

var httpClient = new HttpClient();

var result = httpClient.GetStringAsync(url).Result;

return JsonSerializer.Deserialize<WebAccessToken>(result)!.openid;

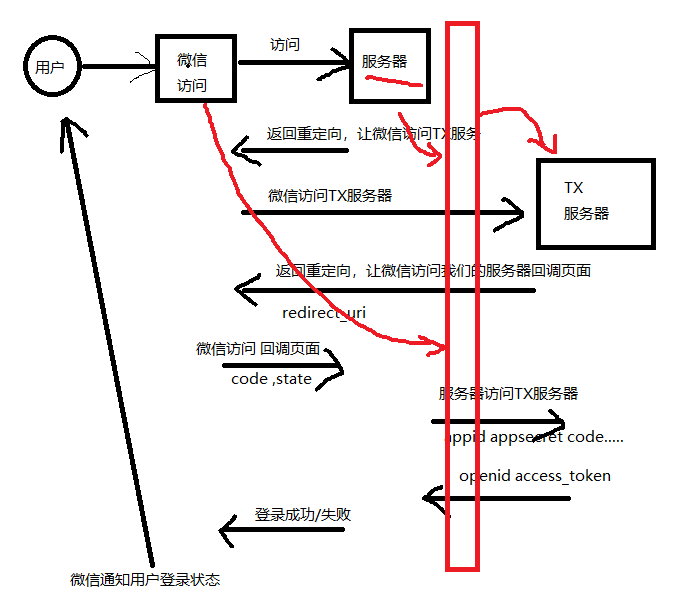
}

## 微信公众号网页授权的限制与突破

一般流程

1. 网页/二维码
2. **用户在微信中打开网页链接/用户使用微信扫描二维码**
3. 登录成功，使用系统

如果用户未将自己的账号与微信公众号绑定，则需要进行一次绑定操作。



## 仿造一个网页授权服务器

var url = $"**https://open.weixin.qq.com/connect/oauth2/authorize?**appid={configuration["WeChatMp:AppId"]}&redirect\_uri={HttpUtility.UrlEncode(configuration["WeChatMp:RedirectUri"])}&response\_type=code&scope={Scope}&state={State}#wechat\_redirect";

var url = $"**https://api.weixin.qq.com/sns/oauth2/access\_token?**appid={configuration["WeChatMp:AppId"]}&secret={configuration["WeChatMp:AppSecret"]}&code={Code}&grant\_type=authorization\_code";

http://192.168.240.232:5188/connect/oauth2/authorize?redirect\_uri=http%3A%2F%2F127.0.0.1%2Fabc.aspx&scope=scope&state=state http://192.168.240.232:5188/sns/oauth2/access\_token?code=code

http://127.0.0.1/abc.aspx?code=063ivXFa1HHZFD09iFGa1HAPl20ivXF3&state=123

http://127.0.0.1/abc.aspx?code=20027d6b351d43a0ab3648d7ede50b6d&state=123

http://192.168.240.232:5188/connect/oauth2/authorize?redirect\_uri=&scope=snsapi\_base&state=123

{"access\_token":"access\_token","expires\_in":7200,"refresh\_token":"refresh\_token","openid":"open\_id","scope":"scope"}

## 为页面授权服务器添加数据缓存功能

builder.Services.AddMemoryCache();

private readonly IMemoryCache memoryCache;

public WxMpProxyController(IMemoryCache memoryCache)

{

this.memoryCache = memoryCache;

}

memoryCache.Set(code, DateTime.Now, TimeSpan.FromSeconds(5));

if (memoryCache.TryGetValue(code, out DateTime dateTime) == true)

{

}

## 网页授权的代理服务器

访问主页

判断用户是否已经完成登录

如果已经登录，则执行与用户相关的逻辑

如果没有登录，重定向到登录页面：[http://**localhost:5095**/api/WxOAuth2/123.aspx](http://localhost:5095/api/WxOAuth2/123.aspx)

Scope = "snsapi\_base"

State = "123"

**应用程序的回调页面** redirect\_uri=“http://**localhost:5095**/api/WxOAuth2”

重定向到：[http://**192.168.240.232:5188**/connect/oauth2/authorize](http://192.168.240.232:5188/connect/oauth2/authorize)?

var code = Guid.NewGuid().ToString().Replace("-", "")

memoryCache.Set(**code**, DateTime.Now, TimeSpan.FromSeconds(5));

重定向到：redirect\_uri [http://**localhost:5095**/api/WxOAuth2](http://localhost:5095/api/WxOAuth2) **应用程序的回调页面**

Code

State

访问[http://](http://192.168.240.232:5188/sns/oauth2/access_token?code={Code})**[192.168.240.232:5188](http://192.168.240.232:5188/sns/oauth2/access_token?code={Code})**[/sns/oauth2/access\_token?code={Code}](http://192.168.240.232:5188/sns/oauth2/access_token?code={Code})

根据返回的数据，判断用户是否登录成功

## 网页授权的代理服务器的实现

访问主页

判断用户是否已经完成登录

如果已经登录，则执行与用户相关的逻辑

如果没有登录，重定向到登录页面：[http://**localhost:5095**/api/WxOAuth2/123.aspx](http://localhost:5095/api/WxOAuth2/123.aspx)

Scope = "snsapi\_base"

State = "123"

**应用程序的回调页面** redirect\_uri=“http://**localhost:5095**/api/WxOAuth2”

重定向到：[http://**192.168.240.232:5188**/connect/oauth2/authorize](http://192.168.240.232:5188/connect/oauth2/authorize)?

var code = Guid.NewGuid().ToString().Replace("-", "")

memoryCache.Set(**code**, DateTime.Now, TimeSpan.FromSeconds(5));

//添加到TX服务器的授权逻辑

//重定向到TX服务器

重定向到：redirect\_uri [http://**localhost:5095**/api/WxOAuth2](http://localhost:5095/api/WxOAuth2) **应用程序的回调页面**

Code

State

访问[http://**192.168.240.232:5188**/sns/oauth2/access\_token?code={Code}](http://192.168.240.232:5188/sns/oauth2/access_token?code=%7bCode%7d)

根据返回的数据，判断用户是否登录成功

## 对现有项目进行重构

## 将项目发布到Windows Server 2022的IIS服务器

1、安装IIS，使用默认配置

2、ASP.NET Core Runtime Hosting Bundle

3、将发布的Web Api复制到Web根目录

4、测试

curl -X "GET" ^

"http://localhost/WeatherForecast" ^

-H "accept: text/plain"

curl http://localhost/WeatherForecast

## 在IIS服务器上启用HTTPS

Programe.cs

app.UseHttpsRedirection();

launchSettings.json

"sslPort": 44396

"applicationUrl": https://localhost:7291;http://localhost:5291

## 将项目发布到Linux服务器

**安装aspnetcore运行时**

dotnet

yum makecache

yum search dotnet

yum search aspnetcore

yum install -y aspnetcore-runtime-6.0

dotnet

dotnet --info

**复制文件**

**运行程序**

cd publish/

dotnet WebApi9.dll

6.0.0-rc.2.21470.23

**修改配置**

vi WebApi9.runtimeconfig.json

6.0.0-rc.2.21470.23

**运行程序**

dotnet WebApi9.dll

**测试**

curl http://localhost:5000/WeatherForecast/

ip a

curl <http://192.168.240.134:5000/WeatherForecast/>

**设置端口**

dotnet WebApi 9.dll --urls="http://\*:5001"

curl <http://192.168.240.134:5001/WeatherForecast/>

**外部访问**

curl <http://192.168.240.134:5001/WeatherForecast/>

systemctl stop firewalld

curl <http://192.168.240.134:5001/WeatherForecast/>

systemctl start firewalld

curl <http://192.168.240.134:5001/WeatherForecast/>

## 使用Nginx来发布Web

Kestrel

Nginx

**安装nginx代理**

yum install -y nginx

systemctl enable nginx

/usr/lib/systemd/system/nginx.service

systemctl start nginx

**配置nginx**

**https://docs.microsoft.com/en-us/aspnet/core/host-and-deploy/linux-nginx?view=aspnetcore-6.0**

vi /etc/nginx/nginx.conf

location / {

proxy\_pass http://127.0.0.1:5000;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection keep-alive;

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

nginx -s reload

**添加防火墙策略**

firewall-cmd --add-port=80/tcp --permanent

firewall-cmd --reload

**SELinux设置**

setenforce 0

curl http://192.168.240.134/WeatherForecast/

sestatus

setenforce 1

**SELinux排错工具**

yum install -y setroubleshoot

cat /var/log/audit/audit.log

tail -n 3 /var/log/audit/audit.log

tail -n 3 /var/log/audit/audit.log | tee /tmp/audit

audit2why -i /tmp/audit

setsebool -P httpd\_can\_network\_connect 1

## 让项目以服务器的方式在Linux系统中运行

**创建服务配置文件**

vi /etc/systemd/system/kestrel-WebApi009.service

**添加内容**

[Unit]

Description=Example .NET Web API App running on Ubuntu

[Service]

WorkingDirectory=/root/publish/

ExecStart=/usr/bin/dotnet /root/publish/WebApi009.dll

Restart=always

# Restart service after 10 seconds if the dotnet service crashes:

RestartSec=10

KillSignal=SIGINT

SyslogIdentifier=dotnet-example

User=root

Environment=ASPNETCORE\_ENVIRONMENT=Production

Environment=DOTNET\_PRINT\_TELEMETRY\_MESSAGE=false

[Install]

WantedBy=multi-user.target

**启用服务**

systemctl enable kestrel-WebApi009

**启动服务**

systemctl start kestrel-WebApi009

**参考文档**

<https://docs.microsoft.com/en-us/aspnet/core/host-and-deploy/linux-nginx?view=aspnetcore-6.0>

## 在Nginx中启用HTTPS(1)

http://192.168.240.232:5188/connect/oauth2/authorize?redirect\_uri=123&scope=12313&state=123131

curl -v -X GET 'http://127.0.0.1:5000/connect/oauth2/authorize?redirect\_uri=123&scope=12313&state=123131'

curl -v -X GET 'http://127.0.0.1/connect/oauth2/authorize?redirect\_uri=123&scope=12313&state=123131'

curl -v -X GET 'http://192.168.0.72/connect/oauth2/authorize?redirect\_uri=123&scope=12313&state=123131'

curl -v -X GET 'http://114.115.182.214/connect/oauth2/authorize?redirect\_uri=123&scope=12313&state=123131'

systemctl status firewalld

sestatus

华为云控制台，防火墙规则，80/tcp放行

redirectUri: http://114.115.182.214:/connect/oauth2/callback

curl -v -X GET 'https://114.115.182.214/connect/oauth2/authorize?redirect\_uri=123&scope=12313&state=123131'

curl -I -X GET 'https://a1.cdut.edu.cn/connect/oauth2/authorize?redirect\_uri=123&scope=12313&state=123131'

"Proxy": "https://a1.cdut.edu.cn"

C:\Windows\System32\drivers\etc

**hosts**

URL: **http:**//a1.cdut.edu.cn/connect/oauth2/callback?code=053doM000FUOnO15Tp100DrmqC2doM00&state=fd2a326cbbb1431f914b96de98ef0319

Error: ERR\_CONNECTION\_REFUSED (-102)

ERR\_CONNECTION\_REFUSED

App-(**https**)-Nginx-(**http**)- Kestrel

## 在Nginx中启用HTTPS(2)

**vi /etc/nginx/proxy.conf**

proxy\_redirect off;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

client\_max\_body\_size 10m;

client\_body\_buffer\_size 128k;

proxy\_connect\_timeout 90;

proxy\_send\_timeout 90;

proxy\_read\_timeout 90;

proxy\_buffers 32 4k;

**vi /etc/nginx/nginx.conf**

# For more information on configuration, see:

# \* Official English Documentation: http://nginx.org/en/docs/

# \* Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;

worker\_processes auto;

error\_log /var/log/nginx/error.log;

pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.

include /usr/share/nginx/modules/\*.conf;

events {

worker\_connections 1024;

}

http {

include /etc/nginx/proxy.conf;

limit\_req\_zone $binary\_remote\_addr zone=one:10m rate=5r/s;

server\_tokens off;

sendfile on;

# Adjust keepalive\_timeout to the lowest possible value that makes sense

# for your use case.

keepalive\_timeout 29;

client\_body\_timeout 10; client\_header\_timeout 10; send\_timeout 10;

upstream helloapp{

server 127.0.0.1:5000;

}

server {

listen 443 ssl http2;

listen [::]:443 ssl http2;

server\_name example.com \*.example.com;

ssl\_certificate /etc/nginx/ssl/6529196\_\_cdut.edu.cn.pem;

ssl\_certificate\_key /etc/nginx/ssl/6529196\_\_cdut.edu.cn.key;

ssl\_session\_timeout 1d;

ssl\_protocols TLSv1.2 TLSv1.3;

ssl\_prefer\_server\_ciphers off;

ssl\_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384;

ssl\_session\_cache shared:SSL:10m;

ssl\_session\_tickets off;

ssl\_stapling off;

add\_header X-Frame-Options DENY;

add\_header X-Content-Type-Options nosniff;

#Redirects all traffic

location / {

proxy\_pass http://helloapp;

limit\_req zone=one burst=10 nodelay;

}

}

}

## “公众号网页授权”与“网站应用微信登录”

**网页授权**

如果用户**在微信客户端**中**访问第三方网页**，公众号可以通过微信网页授权机制，来获取用户基本信息，进而实现业务逻辑。

<https://developers.weixin.qq.com/doc/offiaccount/OA_Web_Apps/Wechat_webpage_authorization.html>

scope

snsapi\_base和snsapi\_userinfo

**网站应用微信登录**

<https://open.weixin.qq.com/>

开发者资质认证：

300元

非个人认证（需要单位认证）

<https://developers.weixin.qq.com/doc/oplatform/Mobile_App/Resource_Center_Homepage.html>

scope

snsapi\_login

区别

<https://open.weixin.qq.com/connect/oauth2/authorize>?

**appid**=APPID&redirect\_uri=REDIRECT\_URI&response\_type=code&scope=SCOPE&state=STATE#wechat\_redirect

https://open.weixin.qq.com/connect/ qrconnect?

**appid**=APPID&redirect\_uri=REDIRECT\_URI&response\_type=code&scope=SCOPE&state=STATE#wechat\_redirect

redirect\_uri?code=CODE&state=STATE

redirect\_uri?code=CODE&state=STATE

https://api.weixin.qq.com/sns/oauth2/access\_token?appid=APPID&secret=SECRET&code=CODE&grant\_type=authorization\_code

https://api.weixin.qq.com/sns/oauth2/access\_token?appid=APPID&secret=SECRET&code=CODE&grant\_type=authorization\_code

## 域名、ICP备案及SSL证书

域名

<https://www.aliyun.com/>

ICP备案

https://beian.ctyun.cn/beian/index

https://beian.miit.gov.cn/#/Integrated/index

SSL证书

<https://cloud.tencent.com/>

<https://console.cloud.tencent.com/ssl>

## 在微信开放平台中创建网站应用

## “微信扫一扫登录”的实现

systemctl stop kestrel-www

vi /etc/systemd/system/kestrel-www.service

systemctl daemon-reload

systemctl start kestrel-www

systemctl restart kestrel-www

<http://localhost:5095/LoginCallback.aspx?code=0619Dn0w3GCo7Z2Ggc1w3KxpCP19Dn0B&state=123>

APP:<br>{"errcode":40029,"errmsg":"invalid code, rid: 630b0a63-0b382575-3164165b"}

APP:<br>{"access\_token":"60\_lffrul9kBouhLLBH3fSkmuiA2gF8hH8AAQzCgFzVdtmBkrQGA6dbd6wkjQrCEKUiAXicA-5CPn4HLP5pIEnf5w8uTF-I16KIbKb3P72H4eg","expires\_in":7200,"refresh\_token":"60\_1JytsmV4CvaIUN37Ffg5CpFrVaKm\_REWhMxWGf4pMgV0dD-hPEXoii-2n8UNp0ddUCZBED3CCszHW3WZVA-V5kii7vmZ\_risFXAx\_16uP3E","**openid**":"o1icI6joUrUkZ6wE41K9pF2x7Ncc","scope":"snsapi\_login","unionid":"ogBOp5gLMJp93zoLn9DJRycCLRqI"}

<http://localhost:5095/WeChatScan.aspx>

[https://open.weixin.qq.com/connect/qrconnect?appid=wx57f37c905504f034&redirect\_uri=https%3a%2f%2f**www.my8421.com**%2fconnect%2foauth2%2fcallback&response\_type=code&scope=snsapi\_login&state=0cc719b8bd1b467c81bc644704ebdb62#wechat\_redirect](https://open.weixin.qq.com/connect/qrconnect?appid=wx57f37c905504f034&redirect_uri=https%3a%2f%2fwww.my8421.com%2fconnect%2foauth2%2fcallback&response_type=code&scope=snsapi_login&state=0cc719b8bd1b467c81bc644704ebdb62#wechat_redirect)

<http://localhost:5095/LoginCallback.aspx?code=031b2i100OXttO1YQc4006zydi0b2i1V&state=123>