前文

本文所有内容来自官方文档，如果有写不明白的地方请下方留言或查看官方文档。

今天介绍Authorization Code模式，适用于保密类型的客户端，保密类型客户端可以理解为在服务器端生成页面（比如传统mvc）。

### 开始

我们需要改进的东西：

Is4项目

我们需要的东西：

一个mvc客户端

### Is4项目改进

#### 1）添加UI

因为Authorization Code中客户端涉及到用户（资源持有人），所以我们需要一个登陆页面。在is4项目文件夹中打开命令行执行 dotnet new is4ui，可以看到执行完成之后文件夹是这样的：



多了Views和Quickstart文件夹。

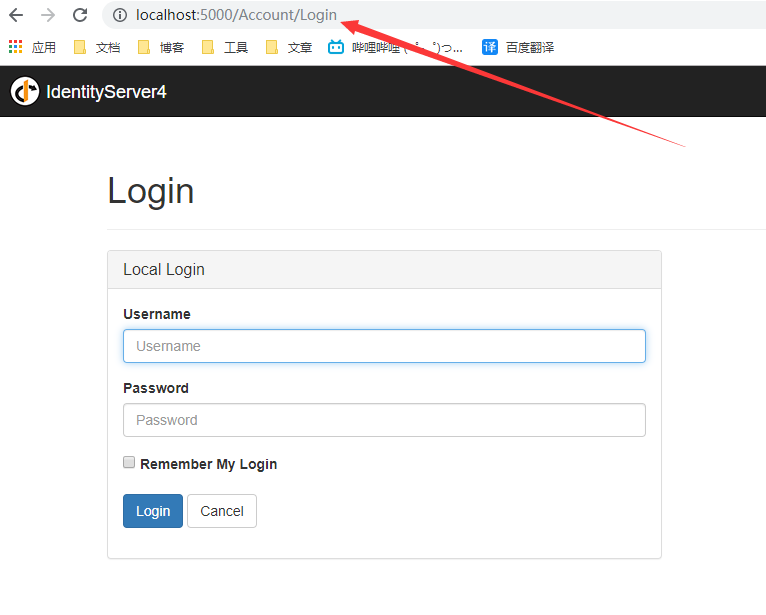
注：

Views：存放视图文件。

Quickstart：存放控制器，检查器等一些列核心工具。（比较重要）

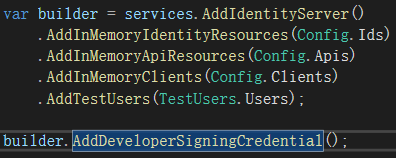
在依赖注入中添加控制器和视图：services.AddControllersWithViews();

此时执行的话会看到一个没有css的网页，因为我们还没有加静态文件中间件。在startup中app.UseIdentityServer();之前添加app.UseStaticFiles();这样运行应该会看到以下界面：

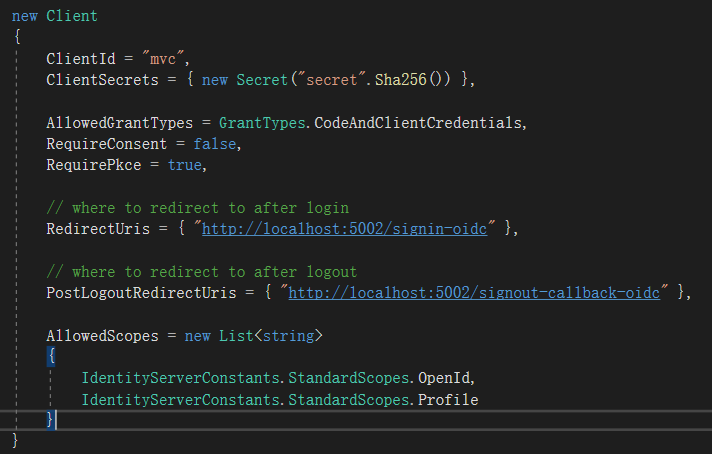


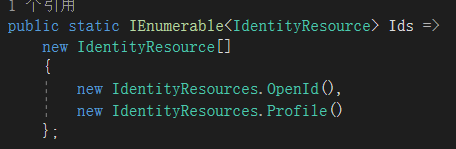
#### 2）添加测试用户

这个比较简单，因为生成UI的时候携带一个测试用户的文件叫做TestUser.cs，在Quickstart文件夹下可以找到，所以我们只需要使用AddTestUsers把测试用户添加进去即可，代码如下：



#### 3）添加客户端

我们需要在config文件中添加一个关于mvc的客户端，直接贴代码：



由于Authorization Code模式中存在用户，所以我们可以获取用户的某些信息（Ids中定义了可以获取的用户信息）。

此时运行程序，在登录页面可以正常的登录/登出，则证明配置成功。登陆地址可以在上面找到。

### 创建mvc客户端

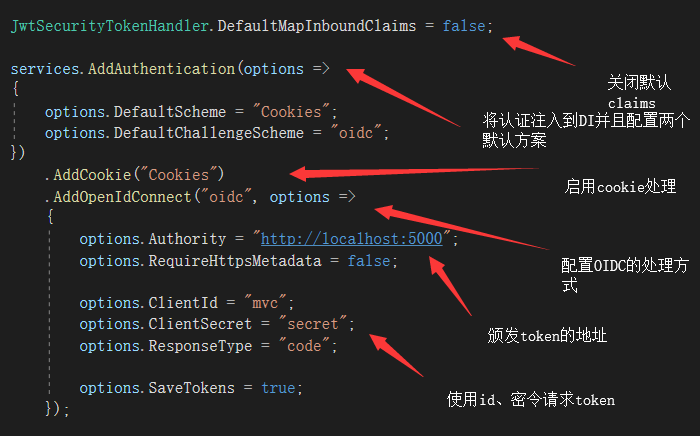
#### 创建项目

首先创建一个mvc客户端，并且安装Microsoft.AspNetCore.Authentication.OpenIdConnect包，很简单不贴图了

#### 配置中间件

需要两个中间件UseAuthentication和UseAuthorization这两个缺一不可。

#### 配置注入服务



#### 4) 显示用户信息

为了方便，我们在home/index中遍历出所有请求到的用户信息：只需要在index页面添加一下代码即可：

@using Microsoft.AspNetCore.Authentication

<h2>Claims</h2>

<dl>

@foreach (var claim in User.Claims)

{

<dt>@claim.Type</dt>

<dd>@claim.Value</dd>

}

</dl>

<h2>Properties</h2>

<dl>

@foreach (var prop in (await Context.AuthenticateAsync()).Properties.Items)

{

<dt>@prop.Key</dt>

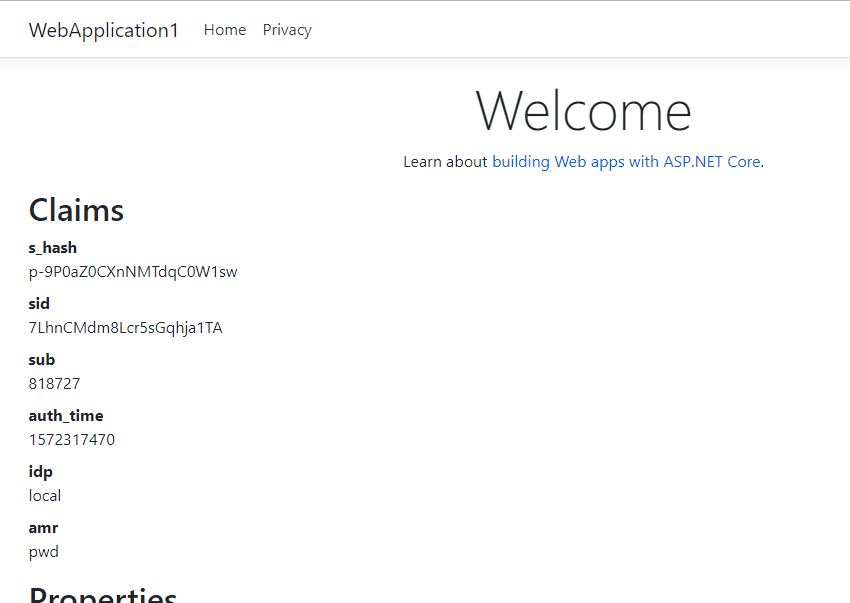
<dd>@prop.Value</dd>

}

</dl>

### 测试

好了我们来登入一下试试：



登录之后应该是这个样子的，具体信息可以自己看一下。

### 总结

如果上一篇理解了，那么这一篇是非常容易的，只需要在各个进程之间添加一些配置就大功告成了。

最后请教一个问题，我在做的时候进行登录的时候，点击登录出现了is4跳转mvc（这是正常的），然后mvc竟然又去is4请求token，两个进程之间疯狂跳转，后来也不知道怎么回事就好了。。。请大佬请指点（抱拳）。