# **认证服务器配置**

# **配置**

**public** **void** ConfigureServices(IServiceCollection services)

{

services.AddIdentityServer()

// 生成临时密匙

.AddDeveloperSigningCredential()

// 添加Api资源

.AddInMemoryApiResources(Config.GetApiResources())

// 添加客户端

.AddInMemoryClients(Config.GetClients());

//添加用户名密码验证器

.AddResourceOwnerValidator<ResourceOwnerPasswordValidator>();

// ProfileService对发布给客户端的信息进行过滤

.AddProfileService<ProfileService>();

}

// 验证器如下

public class ResourceOwnerPasswordValidator : IResourceOwnerPasswordValidator

{

public async Task ValidateAsync(ResourceOwnerPasswordValidationContext context)

{

if(context.UserName!=”username” || context.Password != “123456”)

{

context.Result = new GrantValidationResult(TokenRequestErrors.InvalidGrant, “用户名或密码错误”);

return;

}

context.Result = new GrantValidationResult(

subject: ”username”,

authenticationMethod: "custom",

claims: new Claim[]

{

new Claim("Id", “1”),

new Claim("UserName", ”username”),

new Claim("EmailAddress", “123456@qq.com”),

}

);

}

}

# **客户端定义**

添加以下代码到你配置中的客户端定义中：

**public** **static** IEnumerable<Client> GetClients()

{

**return** **new** List<Client>

{

**new** Client

{

ClientId = "ro.client",

AllowedGrantTypes = GrantTypes.ResourceOwnerPassword, // 资源所有者密码

ClientSecrets =

{

**new** Secret("secret".Sha256())

},

AllowedScopes = { "api1" }

}

};

}

**客户端配置**

IdentityModel 的 TokenClient 在这里再次为我们提了供帮助：

// 从元数据中发现客户端

var disco = await DiscoveryClient.GetAsync("http://localhost:5000");

// 实例资源所有者客户端

var tokenClient = **new** TokenClient(disco.TokenEndpoint, "ro.client", "secret");

// 请求用户令牌

var tokenResponse = **await** tokenClient.RequestResourceOwnerPasswordAsync("alice", "password", "api1");

**if** (tokenResponse.IsError){

Console.WriteLine(tokenResponse.Error);

**return**;}

Console.WriteLine(tokenResponse.Json);Console.WriteLine("\n\n");