

# 第14期-定义身份资源与API资源

2020年6月16日 14:34

OpenID Connect 和 OAuth 令牌服务的最终目的是控制对资源的访问。

```
public class InMemoryResourcesStore : IResourceStore
```

## 定义身份资源

代表有关用户的声明，例如用户ID，显示名称，电子邮件地址等。

```
public static IEnumerable<IdentityResource> GetIdentityResources()
{
    return new List<IdentityResource>
    {
        new IdentityResource(
            name: "openid",
            claimTypes: new[] { "sub" },
            displayName: "Your user identifier")
    };
}
```

可以对常用的身份资源定义进行简写，以下代码等价。

```
public static IEnumerable<IdentityResource> GetIdentityResources()
{
    return new List<IdentityResource>
    {
        new IdentityResources.OpenId()
    };
}
```

支持更加详细的配置和从配置文件中读取身份资源。

### Identity Resource Reference

以下示例显示了一个名为 profile 的自定义身份资源

```
public static IEnumerable<IdentityResource> GetIdentityResources()
{
    return new List<IdentityResource>
    {
        new IdentityResource(
            name: "profile",
            claimTypes: new[] { "name", "email", "website" },
            displayName: "Your profile data")
    };
}
```

```
};
}
```

定义资源后，需要使用 AllowedScopes 选项将访问权限授予客户端。

```
var client = new Client
{
    ClientId = "client",

    AllowedScopes = { "openid", "profile" }
};
```

然后，客户端可以使用scope参数（其他参数省略）请求资源：

GET /authorize?client\_id=client&scope=openid profile

这些身份资源最终通过 IProfileService 提供服务，默认实现为 DefaultProfileService 类。

## 定义 API 资源

代表客户端想要访问的功能。通常，它们是基于HTTP的终结点，也可以是其它。

每个API 也可能具有作用域。某些范围可能是该资源专有的，而某些范围可能是共享的。

```
public static IEnumerable<ApiScope> GetApiScopes()
{
    return new List<ApiScope>
    {
        new ApiScope(name: "read",    displayName: "Read your data."),
        new ApiScope(name: "write",   displayName: "Write your data."),
        new ApiScope(name: "delete",  displayName: "Delete your data.")
    };
}
```

### API Scope Reference

然后，您可以将范围分配给各种客户端，例如：

```
var webViewer = new Client
{
    ClientId = "web_viewer",

    AllowedScopes = { "openid", "profile", "read" }
};

var mobileApp = new Client
{
```

```
ClientId = "mobile_app",

AllowedScopes = { "openid", "profile", "read", "write", "delete" }
}
```

当客户要求一个范围（并且该范围是通过配置允许的，而不是通过同意拒绝）时，该范围的值将作为类型范围的声明（对于JWT和自省）包含在结果访问令牌中。

```
{
  "typ": "at+jwt"
}.
{
  "client_id": "mobile_app",
  "sub": "123",

  "scope": "read write delete"
}
```

访问令牌的使用者可以使用该数据来确保实际上允许客户端调用相应的功能。

**请注意：范围仅用于授权客户端，而不是用户。**

您可以通过从范围请求中派生其他声明来添加有关用户的更多身份信息。

```
var writeScope = new ApiScope(
    name: "write",
    displayName: "Write your data.",
    userClaims: new[] { "user_level" });
```

以便访问令牌的使用者可以将此声明数据用作授权决策或业务逻辑的输入。

[API Resource Reference](#)