## **定义服务器到服务器通信的客户端**

在这种情况下，没有交互式用户 - 服务（也称为客户端）想要与API（aka范围）进行通信：

**public** **class** **Clients**{

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

{

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

{

**new** Client

{

ClientId = "service.client",

ClientSecrets = { **new** Secret("secret".Sha256()) },

AllowedGrantTypes = GrantTypes.ClientCredentials,

AllowedScopes = { "api1", "api2.read\_only" }

}

};

}

}

## **定义服务器端Web应用程序（例如MVC）以进行使用身份验证和委托API访问**

var mvcClient = **new** Client{

ClientId = "mvc",

ClientName = "MVC Client",

ClientUri = "http://identityserver.io",

AllowedGrantTypes = GrantTypes.Hybrid,

AllowOfflineAccess = **true**,

ClientSecrets = { **new** Secret("secret".Sha256()) },

RedirectUris = { "http://localhost:21402/signin-oidc" },

PostLogoutRedirectUris = { "http://localhost:21402/" },

FrontChannelLogoutUri = "http://localhost:21402/signout-oidc",

AllowedScopes =

{

IdentityServerConstants.StandardScopes.OpenId,

IdentityServerConstants.StandardScopes.Profile,

IdentityServerConstants.StandardScopes.Email,

"api1", "api2.read\_only"

},};