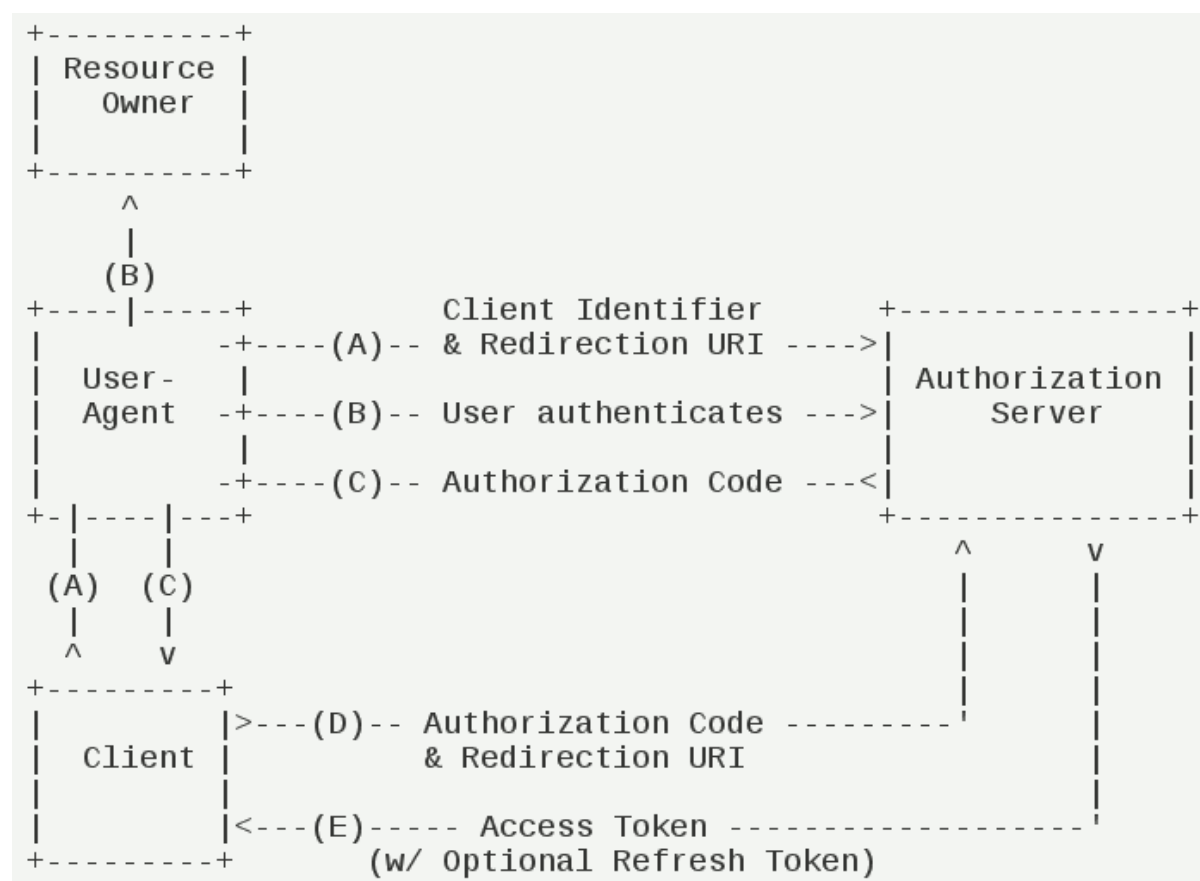


第05期-使用交互式授权码模式

2020年6月9日 19:34

授权码模式（authorization code）是功能最完整、流程最严密的授权模式。它的特点就是通过客户端的后台服务器，与"服务提供商"的认证服务器进行互动。

流程图



请求步骤

(A) 用户访问客户端，后者将前者导向认证服务器。

(B) 用户选择是否给予客户端授权。

(C) 假设用户给予授权，认证服务器将用户导向客户端事先指定的"重定向URI"（redirection URI），同时附上一个授权码。

(D) 客户端收到授权码，附上早先的"重定向URI"，向认证服务器申请令牌。这一步是在客户端的后台的服务器上完成的，对用户不可见。

(E) 认证服务器核对了授权码和重定向URI，确认无误后，向客户端发送访问令牌（access token）和更新令牌（refresh token）。

所需要的参数

A步骤中，客户端申请认证的URI，包含以下参数：

response_type: 表示授权类型，必选项，此处的值固定为"code"
client_id: 表示客户端的ID，必选项
redirect_uri: 表示重定向URI，可选项
scope: 表示申请的权限范围，可选项
state: 表示客户端的当前状态，可以指定任意值，认证服务器会原封不动地返回这个值。
下面是一个例子。

```
GET /authorize?response_type=code&client_id=s6BhdRkqt3&state=xyz
    &redirect_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb HTTP/1.1
Host: server.example.com
```

C步骤中，服务器回应客户端的URI，包含以下参数：

code: 表示授权码，必选项。该码的有效期应该很短，通常设为10分钟，客户端只能使用该码一次，否则会被授权服务器拒绝。该码与客户端ID和重定向URI，是一一对应关系。
state: 如果客户端的请求中包含这个参数，认证服务器的回应也必须一模一样包含这个参数。
下面是一个例子。

```
HTTP/1.1 302 Found
Location: https://client.example.com/cb?code=SplxlOBeZQQYbYS6WxSbIA
    &state=xyz
```

D步骤中，客户端向认证服务器申请令牌的HTTP请求，包含以下参数：

grant_type: 表示使用的授权模式，必选项，此处的值固定为"authorization_code"。
code: 表示上一步获得的授权码，必选项。
redirect_uri: 表示重定向URI，必选项，且必须与A步骤中的该参数值保持一致。
client_id: 表示客户端ID，必选项。
下面是一个例子。

```
POST /token HTTP/1.1
Host: server.example.com
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
Content-Type: application/x-www-form-urlencoded
```

```
grant_type=authorization_code&code=SplxlOBeZQQYbYS6WxSbIA
&redirect_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb
```

E步骤中，认证服务器发送的HTTP回复，包含以下参数：

access_token: 表示访问令牌，必选项。
token_type: 表示令牌类型，该值大小写不敏感，必选项，可以是bearer类型或mac类型。
expires_in: 表示过期时间，单位为秒。如果省略该参数，必须其他方式设置过期时间。
refresh_token: 表示更新令牌，用来获取下一次的访问令牌，可选项。

scope: 表示权限范围, 如果与客户端申请的范围一致, 此项可省略。
下面是一个例子。

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
```

```
{
  "access_token": "2YotnFZFEjr1zCsicMWpAA",
  "token_type": "example",
  "expires_in": 3600,
  "refresh_token": "tGzv3JOkF0XG5Qx2TlKWIA",
  "example_parameter": "example_value"
}
```

从上面代码可以看到, 相关参数使用JSON格式发送 (Content-Type: application/json)。此外, HTTP头信息中明确指定不得缓存。

IdentityServer 集成UI界面

```
dotnet new -i IdentityServer4.Templates
dotnet new is4empty
dotnet new is4ui
```