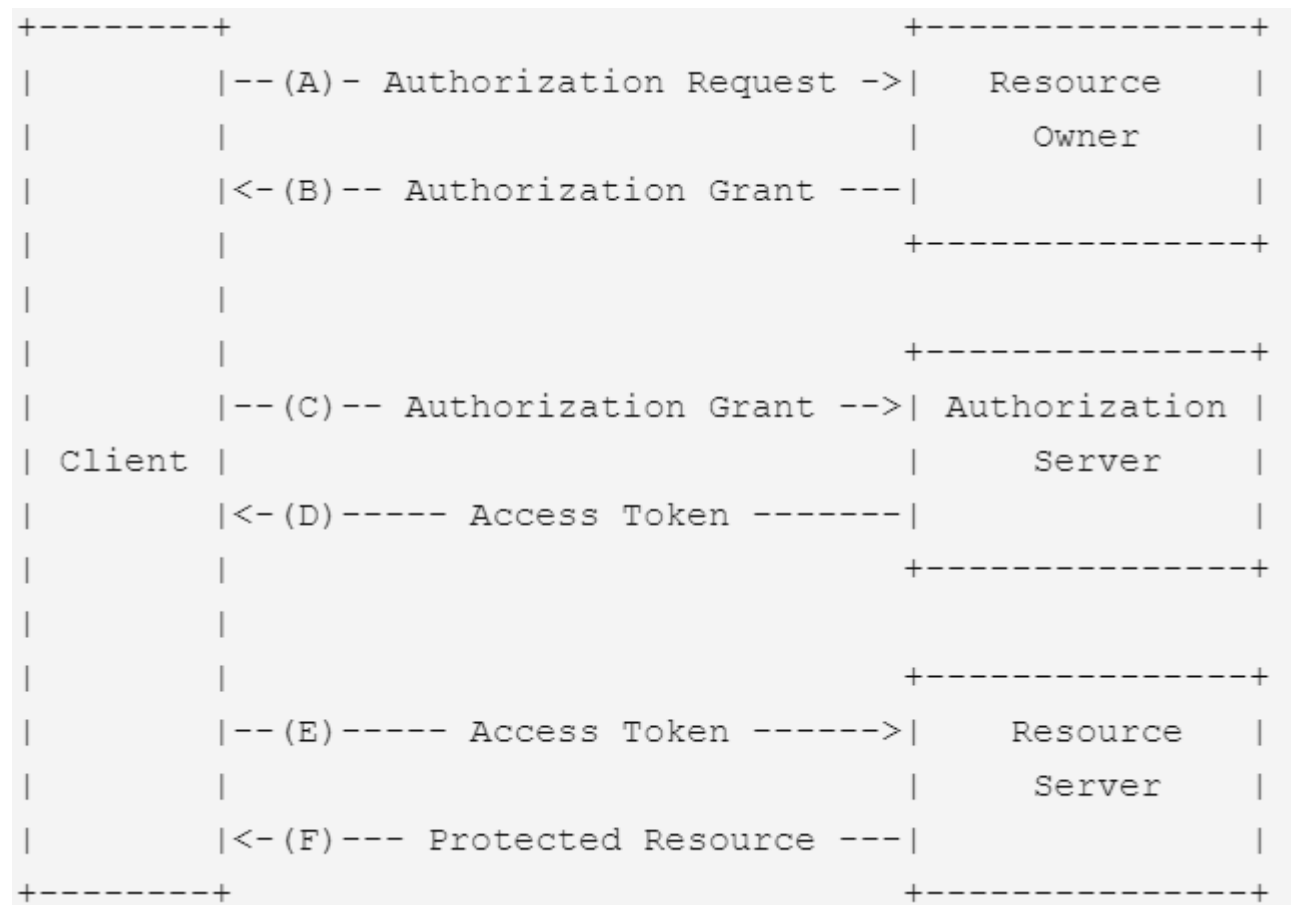


# 第10期-使用JwtBearer授权API接口

2020年6月16日 14:34



## 首先搞清楚 JWT 和 Bearer 的区别

JSON Web Token 是一种开放的行业标准 RFC 7519方法，用于在双方之间安全地表示声明。

HTTP 协议中的 Authorization 请求消息头含有服务器用于验证用户代理身份的凭证。

Bearer 是HTTP 协议中传递令牌的方式 Authorization: Bearer <jwt token>

**Cookie 与 Bearer 在传递令牌上的区别（跨域，移动端友好，避免CSRF）**

## 将令牌设置到 HttpClient 请求标头

```
public async Task<IActionResult> CallApi()
{
    var accessToken = await HttpContext.GetTokenAsync("access_token");

    var client = new HttpClient();
    client.DefaultRequestHeaders.Authorization = new
```

```

AuthenticationHeaderValue("Bearer", accessToken);
    var content = await client.GetStringAsync("https://localhost:6001/identity");

    ViewBag.Json = JObject.Parse(content).ToString();
    return View("json");
}

```

如果引入了 IdentityModel 包，可以直接用提供的扩展方法。

```
httpClient.SetBearerToken("<access token>")
```

## 获取令牌的方式

```

string accessToken = await
HttpContext.GetTokenAsync(OpenIdConnectParameterNames.AccessToken);
string idToken = await HttpContext.GetTokenAsync(OpenIdConnectParameterNames.IdToken);
string refreshToken = await
HttpContext.GetTokenAsync(OpenIdConnectParameterNames.RefreshToken);

```

## 基于 SPA 单页应用的 JavaScript 客户端

<https://github.com/IdentityModel/oidc-client-js>

<https://github.com/IdentityModel/oidc-client-js/wiki>

- 添加您的HTML和JavaScript文件
- 将客户端注册添加到IdentityServer的JavaScript客户端
- 允许使用CORS对Web API进行Ajax调用

## Fetch API 和 Promise

## 参考资料

[ScopeToClaimsMapping.cs](#) 与 [ClaimTypeMapping.cs](#)