**HRM CODE TUTORIAL COLLECTION**

**Collection By:** R&D Team

Contents

[OAuth and OpenID Overview 4](#_Toc62652169)

[OpenID 4](#_Toc62652170)

[OAuth 2.0 5](#_Toc62652171)

[1. Introduction 5](#_Toc62652172)

[1.1. Roles 5](#_Toc62652173)

[1.2. Protocol Flow 5](#_Toc62652174)

[1.3. Authorization Grant Overview 6](#_Toc62652175)

[2. Client Registration 6](#_Toc62652176)

[3. Protocol Endpoints 6](#_Toc62652177)

[Authorization Endpoint 7](#_Toc62652178)

[Token Endpoint 7](#_Toc62652179)

[Access Token Scope 7](#_Toc62652180)

[4. Obtaining Authorization 7](#_Toc62652181)

[4.1. Authorization Code Grant 8](#_Toc62652182)

[4.2. Implicit Grant 8](#_Toc62652183)

[4.3. Resource Owner Password Credentials Grant 8](#_Toc62652184)

[4.4. Client Credential Grant 8](#_Toc62652185)

[5. Issuing an Access Token 8](#_Toc62652186)

[Successful Response 8](#_Toc62652187)

[Error Response 9](#_Toc62652188)

[6. Refreshing an Access Token 9](#_Toc62652189)

[7. Accessing Protected Resource 9](#_Toc62652190)

[7.1. Access Token Types 9](#_Toc62652191)

[Grant Types 11](#_Toc62652192)

[What is an OAuth 2.0 Grant Type? 11](#_Toc62652193)

[The Authorization Code Flow 11](#_Toc62652194)

[Authorization Code Flow with Proof Key for Code Exchange (PKCE) 14](#_Toc62652195)

[OpenID Connect Core 19](#_Toc62652196)

[1. Introduction 19](#_Toc62652197)

[2. ID Token 20](#_Toc62652198)

[3. Authentication 21](#_Toc62652199)

[3.1. Authentication using the Authorization Code Flow 21](#_Toc62652200)

[3.2. Authentication using the Implicit Flow 22](#_Toc62652201)

[3.3. Authentication using the Hybrid Flow 22](#_Toc62652202)

[4. Initiating Login from a Third Party 23](#_Toc62652203)

[5. Claims 23](#_Toc62652204)

[6. Passing Request Parameters as JWTs 23](#_Toc62652205)

[7. Self-Issued OpenID Provider 23](#_Toc62652206)

[8. Subject Identifier Types 23](#_Toc62652207)

[9. Client Authentication 23](#_Toc62652208)

[10. Signatures and Encryption 23](#_Toc62652209)

[11. Offline Access 23](#_Toc62652210)

[12. Using Refresh Tokens 23](#_Toc62652211)

[13. Serializations 23](#_Toc62652212)

[14. String Operation 23](#_Toc62652213)

[15. Implemention Considerations 23](#_Toc62652214)

[16. Security Considerations 23](#_Toc62652215)

[17. Privacy Considerations 23](#_Toc62652216)

[18. IANA Considerations 23](#_Toc62652217)

[19. References 23](#_Toc62652218)

[OpenID Connect Single Logout (SLO) 23](#_Toc62652219)

[Overview 23](#_Toc62652220)

[OpenID Connect Logout 25](#_Toc62652221)

[OpenID Connect Back-Channel Logout 1.0 26](#_Toc62652222)

[OpenID Connect Front-Channel Logout 1.0 31](#_Toc62652223)

[OpenID Connect Session Management 1.0 32](#_Toc62652224)

[OAuth 2.0 and OIDC Support 33](#_Toc62652225)

[OAuth and OpenID Connect Concepts 33](#_Toc62652226)

[OAuth 2.0 Endpoints 33](#_Toc62652227)

[OAuth 2.0 and OIDC Workflows 33](#_Toc62652228)

[OAuth State Management 33](#_Toc62652229)

[Trusted Clients Management 33](#_Toc62652230)

[Proof Key for Code Exchange Support 33](#_Toc62652231)

[Reverse Proxy Configuration for OAuth and OIDC Provider 33](#_Toc62652232)

[Configuring OAuth 2.0 API Protection 33](#_Toc62652233)

[OIDC Claims Customization 33](#_Toc62652234)

[… More 33](#_Toc62652235)

[Bearer Token Usage 34](#_Toc62652236)

[Introduction 34](#_Toc62652237)

[Authenticated Requests 34](#_Toc62652238)

[Authorization Code Header Field 34](#_Toc62652239)

[Form-Encoded Body Parameter 34](#_Toc62652240)

[URI Query Parameter 34](#_Toc62652241)

[JSON Web Tokens (JWT) 35](#_Toc62652242)

[What is JWT 35](#_Toc62652243)

[When should you use JWTs 35](#_Toc62652244)

[Authorization 35](#_Toc62652245)

[Information Exchange 35](#_Toc62652246)

[What is the JWT Structure 35](#_Toc62652247)

[How do JWTs work 36](#_Toc62652248)

[Why should we use JWTs 36](#_Toc62652249)

[OpendID Connect Client Initiated Backchannel Authentication Flow - Draft 37](#_Toc62652250)

[Introduction 37](#_Toc62652251)

[OpendID Connect Note 38](#_Toc62652252)

[Authentication with OpenID Connect Relying Party 38](#_Toc62652253)

[Note 38](#_Toc62652254)

[Note 39](#_Toc62652255)

# Authorization Code with PKCE

Khi chúng ta có một Public Client (native app, single page app), chúng ta sẽ gặp phải vấn đề với Authorization Code Flow trong trường hợp phải lưu trữ Client Secret tại phía Client, nơi mà việc expose Secret này có thể xảy ra một cách dễ dàng

Và vì vậy OAuth 2.0 cung cấp thêm một phiên bản của Authorization Code có sử dụng PKCE.

PKCE-enhanced Authorization Code Flow giới thiệu thêm một Secret được tạo bởi Application, và Secret này có thể được Verify bởi Authorization Server, tên gọi của Secret này là Code Verifier. Và thêm nữa, Application từ Code Verifier sẽ tạo thêm một thứ gọi là Code Challenge và gửi Code Challenge này thông qua HTTPS để nhận Authorization Code.

Với phương pháp này, một Malicious Attacker chỉ có thể can thiệp vào Authorization Code, tuy nhiên sẽ không thể sử dụng Code này để đổi lấy một Access Token mà không có Code Verifier.

## OpenID