Securing the Client Applications (OAuth 2.0 and OpenID Connect)

Kevin Dockx

@KevinDockx | http://blog.kevindockx.com/

MVC Client: Choosing the Correct Flow

- MVC is a **confidential** client
  - Authorization Code flow: access_token, refresh_token

- We also require **identity**
  - OpenID Connect: id_token & openid scope

- New flow: Hybrid flow
  - id_token & access_token, openid scope, refresh_token via authorization code

Hybrid Flow

Learn how to use the Hybrid Flow

UserInfo Endpoint

Learn how to use the UserInfo endpoint to get identity information

Claims Transformation

Learn how to transform claims and create a ClaimsIdentity that only holds the information we need

Role-Based Authorization

Learn how to add role-based authorization

External Providers (Facebook)

Learn how to add login functionality through an external provider

User-Specific Data – Client Responsibility

Learn how to request data belonging to the authenticated user, putting the responsibility on the client

WP Client: Choosing the Correct Flow

- Native Windows Phone is a **public** client
  - Implicit flow: access_token, no refresh token
- We also require **identity**
  - OpenID Connect: id_token & openid scope
- Implicit flow is a good fit
  - id_token & access_token, openid scope
  - Also for javascript (user-agent based) clients

IIS Express and Windows Phone

Learn how to work with a locally hosted STS in combination with Windows Phone

Implicit Flow

Learn how to use the implicit flow

Summary

For an MVC client, Authorization Code flow and Hybrid flow are the best fits

The [ResourceAuthorize] attribute separates authorization policy from controllers and business logic

We can use a filter for user-specific data

For a native client (or user-agent based clients), Implicit flow is advised