Cloud Access Manager 8.1.4

How to Configure Single Sign-On for Native IOS Applications

Legend

WARNING: A WARNING icon indicates a potential for property damage, personal injury, or death.

CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

IMPORTANT, NOTE, TIP, MOBILE, or VIDEO: An information icon indicates supporting information.

Cloud Access Manager How to Configure Single Sign-On for Native IOS Applications
Updated - November 2018
Version - 8.1.4

Contents

1

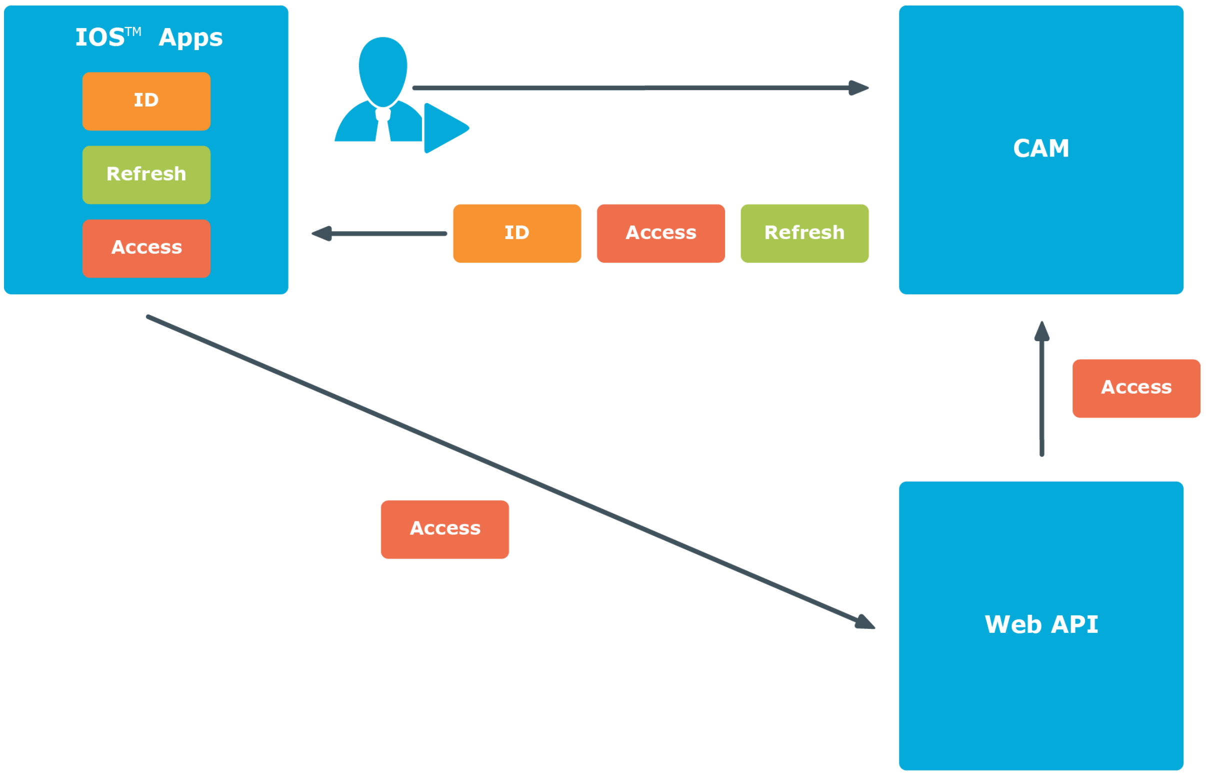
Introduction

This guide describes how to deploy Single Sign-On (SSO) for native IOS applications using the OpenID Connect protocol.

Overview

Using the OpenID Connect protocol, the IOS application authenticates the user against Cloud Access Manager and retrieves a set of three security tokens, as shown in Figure 1. The security tokens are known as the ID Token, Refresh Token and Access Token.

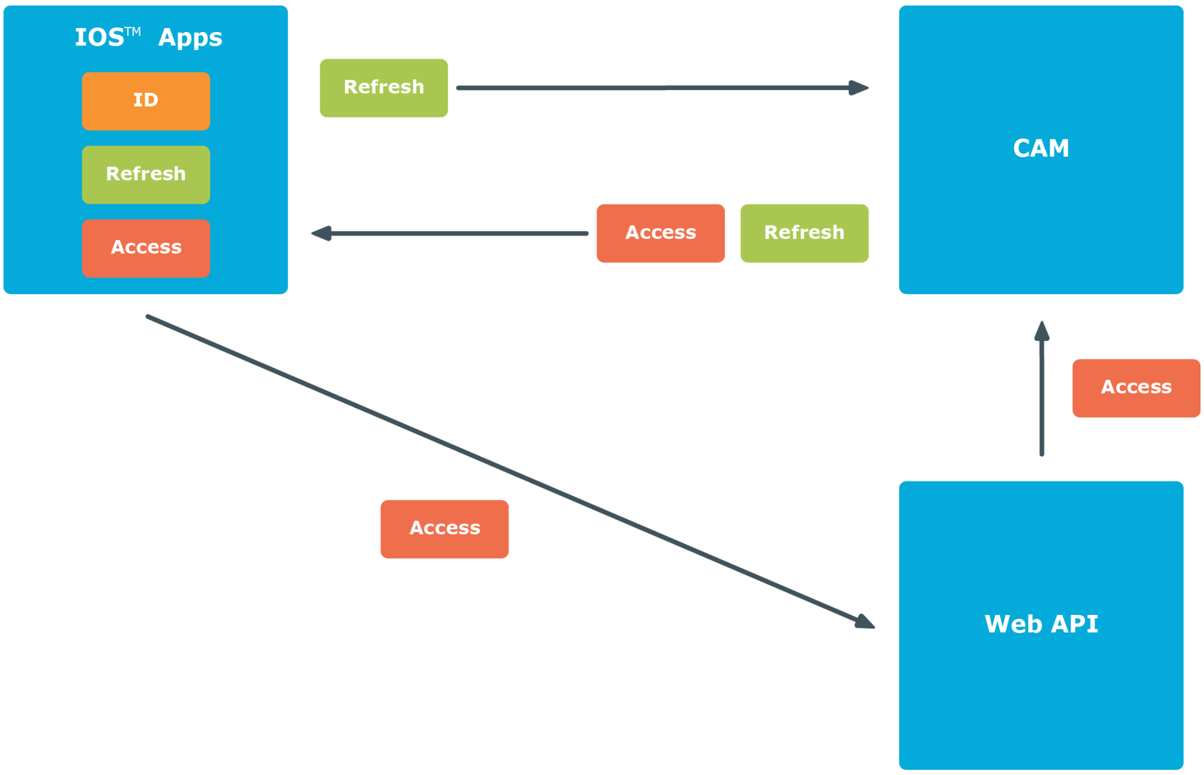
Figure 1: Single Sign-On (SSO) procedure for native IOS applications



The ID Token contains a collection of identity claims about the user that can be used by the IOS application to identify the user. The Access Token allows the IOS application to securely access OAuth2 protected Web APIs on behalf of the user. When the Access Token expires, the Refresh Token is used by the IOS application to obtain a new Access Token, without the need for the user to re-authenticate as shown in Figure 2.

The Web API validates the Access Token by using it to obtain a set of claims about the user from Cloud Access Manager. The claims are then used by the Web API to identify the user and control the user's access.

Figure 2: Using the Refresh Token to obtain a new Access Token



Application walkthrough

This sample application consists of two components:

- IOS OpenID Connect application
- .NET OAuth2 protected Web Application Programming Interface (API)

The sample IOS application contains a package called **openidconnect** which can be used in a standard IOS project to authenticate users using the OpenID Connect Code Flow. The sample Web API contains a .NET Open Web Interface (OWIN) middleware called **CAMBearerTokenAuthentication**. This can be used in a standard .NET Web API project to authenticate the IOS application using the Access Tokens obtained from Cloud Access Manager.

The function of the sample application components

1. When the application starts it checks for an existing ID Token stored from a previous authentication. If an ID Token does not exist the application sends an authentication request using the system browser to start the OpenID Connect Authorization Code Flow. However if an ID Token does exist, the application skips to Step 4.

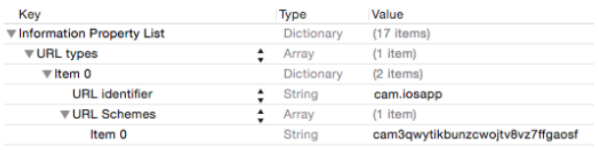
   ```
   tokenStore = SimpleTokenStore.loadDefault()
   codeFlow = CodeFlow(tokenStore: tokenStore!, settings: Config.SETTINGS)

   if (tokenStore?.getIdToken() == nil) {
       // Redirect to login page.
       self.performSegueWithIdentifier("showLogin", sender: self)
   }
   ```

2. The user is then prompted to authenticate to Cloud Access Manager using the system browser. After a successful authentication to Cloud Access Manager, the user is redirected back to the application with an authorization code using a custom URI scheme unique to that application.

   ```
   let url = codeFlow!.authenticationRequestUri()
   UIApplication.sharedApplication().openURL(url)
   ```

   The application's custom URI scheme is registered in the application's property list file (Info.plist). The scheme is used by IOS to determine which application to use to open the URI, so the chosen scheme must be unique. In this example the custom URI scheme is the application's OpenID Connect ClientId in lowercase, prefixed with the name of the SSO provider, Cloud Access Manager (CAM).



Key	Type	Value
▼Information Property List	Dictionary	(17 items)
▼URL types	Array	(1 item)
▼Item 0	Dictionary	(2 items)
URL identifier	String	cam.iosapp
▼URL Schemes	Array	(1 item)
Item 0	String	cam3qwytikbunzcwojtv8vz7ffgaosf

3. The application uses the authorization code within the redirect URI to obtain an ID Token, Refresh Token and Access Token from Cloud Access Manager. The tokens are stored on the device in an app private area. The Access Token is scoped for use with Cloud Access Manager and the sample Web API. The scope of the Access Token is specified in the authentication request described in Step 1.

   ```
   codeFlow?.tokenRequestAsync(appDelegate.appLaunchUrl, onSuccess: { Void in
       // Switch back to the previous view after a successful authentication.
       self.navigationController?.popViewControllerAnimated(true)

   }, onFailure: { error in
       self.loginMsgLabel.text = error
   })
   ```

4. The application can now access the Web API using the Access Token as authorization. The Access Token is included in the authorization header of each request.

   ```
   let url = NSURL(string: resourceServerEndpoint + path)!
   let request = NSMutableURLRequest(URL: url)

   codeFlow.getValidAccessToken(onSuccess: { accessToken in
       request.addValue("Bearer " + accessToken, forHTTPHeaderField: "Authorization")
   ```

5. The Web API validates the Access Token by using it to call the Cloud Access Manager

User Info Endpoint. The validation is performed using the provided OWIN middleware which will cache the User Info responses. The OWIN middleware will also verify that the Access Token was scoped for itself by checking that the User Info response contains at least one of its scopes. The claims returned from the User Info Endpoint are used by the Web API to identify the user and control their access.

The OWIN authentication middleware is registered and configured using:

```
app.UseCAMBearerTokenAuthentication(new CAMBearerTokenAuthenticationOptions
{
    UserInfoEndpoint = "https://cam.company.local/CloudAccessManager/RPSTS/OAuth2/User.aspx",
    RequiredScopes = new[] { "sampleAPI" }
});
```

The standard **Authorize** attribute can be used on the Web APIs to restrict access. The **Authorize** attribute supports restrictions based on role and user claims which, by default map to the claim names **role** and **preferred_username**.

```
[Authorize]
public class ClaimsController : ApiController
{
    // GET: api/Claims
    [Authorize(Roles="read")]
    public IEnumerable<Claim> Get()
    {
        var principal = User as ClaimsPrincipal;
        return principal.Claims;
    }
```

To utilize other claims, a custom **AuthorizeAttribute** can be created. For example:

```
public class RequireCustomClaimAttribute : AuthorizeAttribute
{
    protected override bool IsAuthorized(HttpActionContext context)
    {
        var principal = context.Request.GetRequestContext().Principal as ClaimsPrincipal;
        return principal.Claims.Any(c => c.Type == "custom-claim-name" && c.Value == "true");
    }
}
```

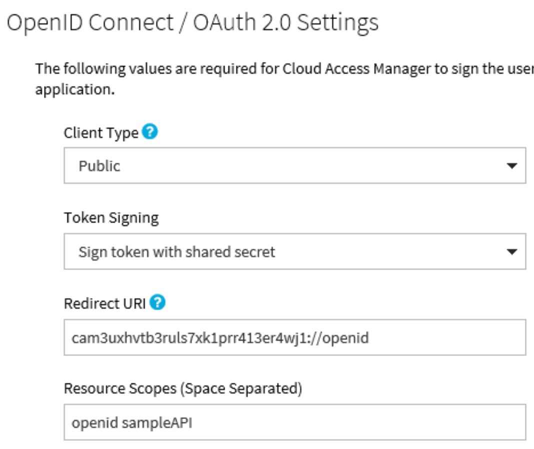
6. The application uses the Refresh Token to pre-emptively obtain a new Refresh Token and Access Token from Cloud Access Manager when the stored Access Token has expired.

Cloud Access Manager configuration

Perform the following configuration steps within Cloud Access Manager to enable single sign-on to native IOS applications.

To configure Cloud Access Manager for single sign-on to native IOS applications

1. Make sure that the settings on the **OpenID Connect / OAuth 2.0 Settings** page are as follows:

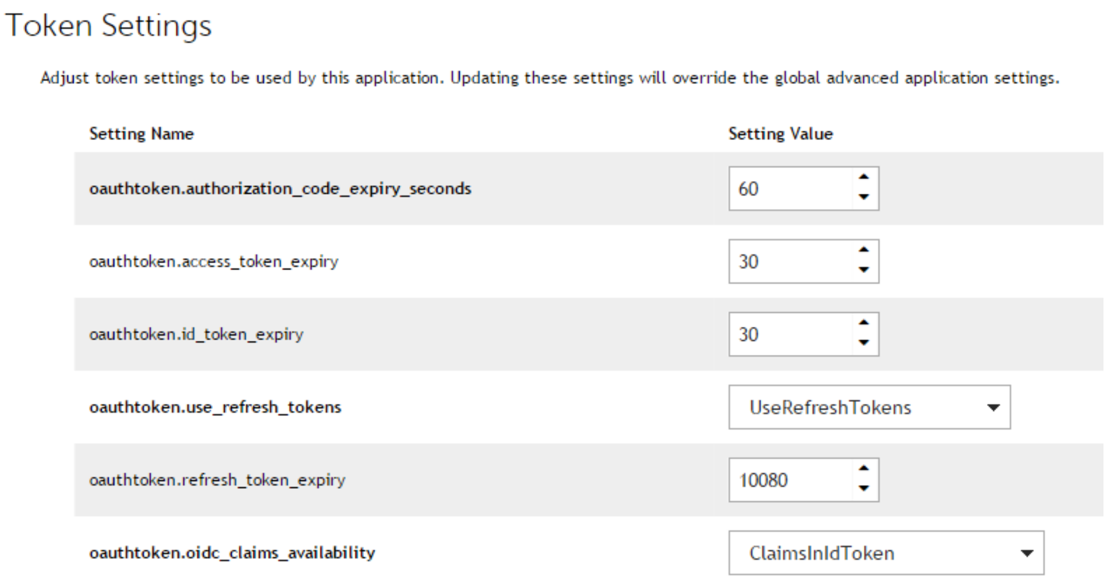


OpenID Connect / OAuth 2.0 Settings

The following values are required for Cloud Access Manager to sign the user in to the application.

Client Type	Public
Token Signing	Sign token with shared secret
Redirect URI	cam3uxhvtb3ruls7xk1prr413er4wj1://openid
Resource Scopes (Space Separated)	openid sampleAPI

2. Make sure that the settings on the **Token Settings** page are as follows:

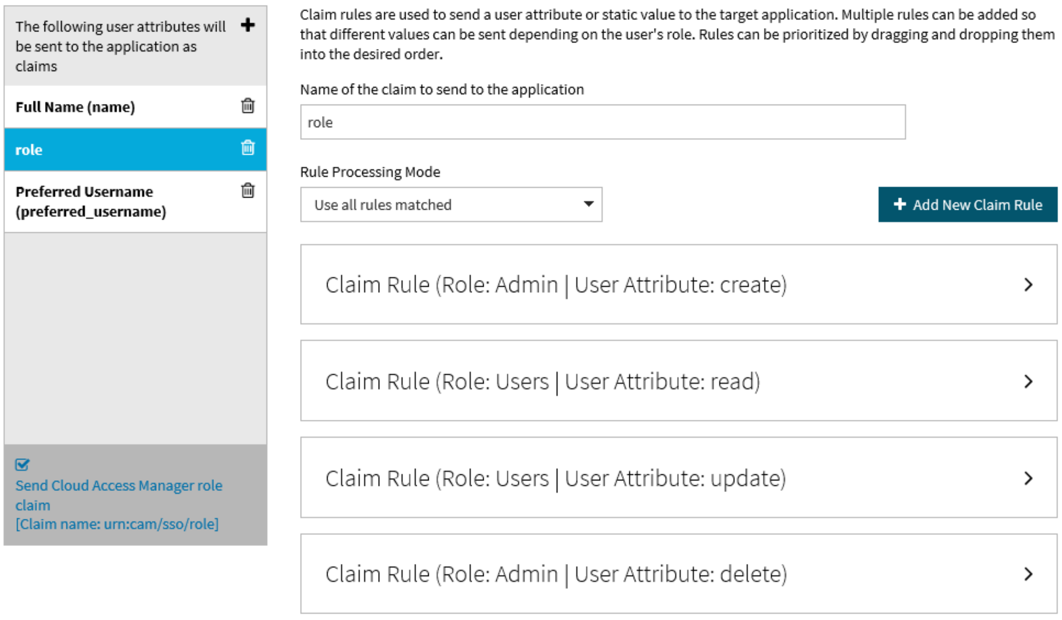


Token Settings

Adjust token settings to be used by this application. Updating these settings will override the global advanced application settings.

Setting Name	Setting Value
oauthtoken.authorization_code_expiry_seconds	60
oauthtoken.access_token_expiry	30
oauthtoken.id_token_expiry	30
oauthtoken.use_refresh_tokens	UseRefreshTokens
oauthtoken.refresh_token_expiry	10080
oauthtoken.oidc_claims_availability	ClaimsInIdToken

3. Make sure that the settings on the **Claim Mapping** page are as follows:



The following user attributes will be sent to the application as claims

Full Name (name)

role

Preferred Username (preferred_username)

Send Cloud Access Manager role claim [Claim name: urn:cam/sso/role]

Claim rules are used to send a user attribute or static value to the target application. Multiple rules can be added so that different values can be sent depending on the user's role. Rules can be prioritized by dragging and dropping them into the desired order.

Name of the claim to send to the application: role

Rule Processing Mode: Use all rules matched

Claim Rule (Role: Admin | User Attribute: create)

Claim Rule (Role: Users | User Attribute: read)

Claim Rule (Role: Users | User Attribute: update)

Claim Rule (Role: Admin | User Attribute: delete)

About us

One Identity solutions eliminate the complexities and time-consuming processes often required to govern identities, manage privileged accounts and control access. Our solutions enhance business agility while addressing your IAM challenges with on-premises, cloud and hybrid environments.

Contacting us

For sales or other inquiries, visit https://www.oneidentity.com/company/contact-us.aspx or call +1-800-306-9329.

Technical support resources

Technical support is available to One Identity customers with a valid maintenance contract and customers who have trial versions. You can access the Support Portal at https://support.oneidentity.com/.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to videos at www.YouTube.com/OneIdentity
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product