# How to use the Dapr OAuth2clientcredentials Middleware Component?

by George Stevens, Solid Value Software

6/10/21, Revision2 of original of 3/3/21

**Information Required for the Dapr OAuth Middleware (Authorization Code) Component Implementation of Client Credentials Flow**

I am having trouble “filling out” the below oauth2clientcredentials middleware component. Specifically :

1. I do not understand where the information comes from to plug into some of the fields of the below yml.
2. It appears there is missing data.

Both of these issues are detailed subsequent to showing the “blank” yml, with no fields filled in directly below

Here is the Dapr OAuth Middleware (Authorization Code) Componentpatterned after that shown in the Dapr Documentation at [https://docs.dapr.io/operations/security/oauth/#define-a-client-credentials-grant-component](https://docs.dapr.io/operations/security/oauth/%23define-a-client-credentials-grant-component). Below, **only the metadata:name: svs-oauth2clientcredentials-aad** has been changed from the originalat the above link.

apiVersion: dapr.io/v1alpha1

kind: Component

metadata:

name: svs-oauth2clientcredentials-aad

spec:

type: middleware.http.oauth2clientcredentials

version: v1

metadata:

- name**: clientId**

value: "<your client ID>"

- name: **clientSecret**

value: "<your client secret>"

- name: **scopes**

value: "<comma-separated scope names>"

- name: **tokenURL**

value: "<token issuing URL>"

- name: **headerName**

value: "<header name under which the secret token is saved>"

- name: **endpointParamsQuery**

value: "<list of additional key=value settings separated by ampersands or semicolons forwarded to the token issuing service>"

# authStyle:

# "0" means to auto-detect which authentication

# style the provider wants by trying both ways and caching

# the successful way for the future.

# "1" sends the "client\_id" and "client\_secret"

# in the POST body as application/x-www-form-urlencoded parameters.

# "2" sends the client\_id and client\_password

# using HTTP Basic Authorization. This is an optional style

# described in the OAuth2 RFC 6749 section 2.3.1.

- name: **authStyle**

value: "<see comment>"

Taking it field by field from the above yml, here are my questions for the fields I am having trouble with, plus ***OK*** with those I am not.

apiVersion: dapr.io/v1alpha1

kind: Component

metadata:

name: svs-oauth2clientcredentials-aad

spec:

type: middleware.http.oauth2clientcredentials

version: v1

metadata:

- name**: clientId**

value: "<your client ID>"

***OK* ClientId** or AppId: Obtained from the Azure Portal AAD App Registrations, Overview screen.

- name: **clientSecret**

value: "<your client secret>"

***OK* ClientSecret**: Obtained from the Azure Portal AAD App Registrations, Certificates and Secrets screen.

Note this really needs to support certificates as well. See Issue #719 in components-contrib repo.

- name: **scopes**

value: "<comma-separated scope names>"

***OK***The “claims” I am using in my C# code is a single claim (scope), <https://servicebus.azure.net/.default>. The ASB itself will verify that the Service Principal sending the request has been previously authorized to do the specific Send, Receive, etc. ASB Topic operation.

- name: **tokenURL**

value: "<token issuing URL>"

***NEED HELP*** -- What should be used here?

Note that in the Azure code examples this is **required only in interactive clients**, and not confidential clients. For example in the IConfidentialClientApplication builder for an interactive app it does the following:

.WithRedirectUri(ConfigurationManager.AppSettings["redirectURI"])

But that is not used in confidential clients as you can see from the above IConfidentialClientApplication builder code. ***TO WHAT VALUE SHOULD I SET THIS***?

- name: **headerName**

value: "<header name under which the secret token is saved>"

***NEED HELP*** -- Should this be Authorization? Or what -- the exact word, please.

- name: **endpointParamsQuery**

value: "<list of additional key=value settings separated by ampersands or semicolons forwarded to the token issuing

service>"

***NEED HELP*** -- Below are items I needed to supply in C# code when I coded up Client Credentials without using Dapr:

1. **TenantId** or Directory (aad) Id: Obtained from the Azure Portal AAD App Registrations, Authentication screen.
   1. My code sample is for a line-of-business app which uses a single Azure Active Directory, aka tenant. A single AAD tenant is represented by a Guid.
   2. Thus it seems that tenantId:tenantIdValue name/value pair should be in the component under endpointParamsQuery. Is this the way to do it?
   3. But MAYBE NOT since in C# code **tenantId is concatenated with another string to produce the “authority” value**, shown directly below. ***NEED HELP.*** ***DO I SET TenantId***? **Or instead use in in the below authority**?
2. **Authority**: Obtained from code examples.
   1. WHERE DOES THIS GO in the component yaml? My code sample uses this to build an instance of IConfidentialClientApplication using the below code:

IConfidentialClientApplication app =

ConfidentialClientApplicationBuilder

.Create(m\_PubCredsNInfo.ClientId)

.WithAuthority(authority)

.WithClientSecret(m\_PubCredsNInfo.ClientSecret)

.Build();

* 1. The value used for “authority” in my code is the string <https://login.windows.net/> **concatenated** with TenantId, as shown by the below code from my sample C# app :

string authority =

$"{AadAuthnAuthzConstants.LoginForAuthNAuthority}{m\_PubCredsNInfo.TenantId}";

//or more clearly

authority = **“**[**https://login.windows.net/**](https://login.windows.net/)**TenantIdGuidValue”**;

* 1. Thus, It seems that authority:authorityValue name/value pair should be in the component under endpointParamsQuery. ***NEED HELP.*** ***Please PROVIDE AN EXAMPLE*** of the value to set this, and the format.

# authStyle:

# "0" means to auto-detect which authentication

# style the provider wants by trying both ways and caching

# the successful way for the future.

# "1" sends the "client\_id" and "client\_secret"

# in the POST body as application/x-www-form-urlencoded parameters.

# "2" sends the client\_id and client\_password

# using HTTP Basic Authorization. This is an optional style

# described in the OAuth2 RFC 6749 section 2.3.1.

- name: **authStyle**

value: "<see comment>"

***OK*** Clearly the authStyle should be 1.

## Information Required for a Native Code Implementation of the Client Credentials Flow

In March 2021 I completed the implementation of a native .NET C# version of the Client Credentials flow for Azure Active Directory (aad) with a C# service publishing items to an Azure Service Bus Topic and another such service subscribing to the topic. The primary Microsoft example code sources I followed are below, and might be useful to review while reading this document since they define many of the Azure and .NET specific terms and techniques:

1. <https://github.com/Azure/azure-service-bus/blob/master/samples/DotNet/Microsoft.Azure.ServiceBus/RoleBasedAccessControl/Program.cs>
2. <https://docs.microsoft.com/en-us/azure/active-directory/develop/scenario-daemon-app-configuration?tabs=dotnet>
3. <https://docs.microsoft.com/en-us/azure/active-directory/develop/scenario-daemon-acquire-token?tabs=dotnet#acquiretokenforclient-api> and its related documents.

Please note that as of March 2021 the above links all use the “old style” of using Azure Identity Client Credentials since the Azure Service Bus had not yet converted to the much simpler-to-code “new style” of Client Credentials. Azure Storage, plus some other Azure Services as well, had been converted to the “new style” in March 2021. As of this 2nd revision on June 10, 2021 I have no idea if the “new style” has been implemented for Azure Service Bus. The “new style” is the Azure Identity Client at <https://docs.microsoft.com/en-us/dotnet/api/overview/azure/identity-readme>. However, regardless of the version of the Client Credentials flow implemented by the Azure Service Bus I still need a Dapr component to implement this feature.