Question
Stephen Wilson · Aug 13, 2019

Adding custom claims to in OAuth 2.0?

I have an OAuth 2.0 development environment where Caché is serving all three roles as the Authorization Server, Client and Resource Server based on a great 3-part series on OAuth 2.0 by @Daniel Kutac. I have a simple password grant type where an *x-www-form-urlencoded* body (as described in this post) is sent as a POST to the token endpoint at https://localhost:57773/oauth2/token and a response body with a HTTP Response 200 header is returned. The response body looks something like this.

```
{

"access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJqdGkiOiJodHRwczovL2NhY2NvbnMx
Lmhwc3Mubi1pLm5ocy51azo1Nzc3My9vYXV0aDIuazZKZlpSZnpBRU02NHpzRDJYUFZsSHRBOElnIiwiaXNzI
joiaHR0cHM6Ly9jYWNjb25zMS5ocHNzLm4taS5uaHMudWs6NTc3NzMvb2F1dGgyIiwic3ViIjoidGVzdDEiLC
JleHAiOjE1NjU2ODc4OTcsImF1ZCI6IkhwSHlfTDA2MVJLVExsaW1OS3FnWjJHR2xkQnE3dWJTNWlZNE5UNFN
fVFkifQ.",

"token_type": "bearer",

"expires_in": 180,

"scope": "createModify openid profile publish"

}
```

In this example, the token generation class is the %OAuth2.Server.JWT class.

The **expiresin** property is the 'Access Token Interval' set in seconds via System Management Portal OAuth options. I have a low interval for the purposes of testing token expiry responses.

Scopes are sorted alphabetically by default. I have added the custom scopes 'createModify' and 'publish' here but these don't really make sense as they are passed into the Request body before the user is authenticated via the token endpoint. You don't know what application roles a user has until they have been authenticated so I would like to return these application roles as claims to the response body. I think I should remove these scopes and replace them with 'MyCustomApplication'. This should be the same for all users. I can sometimes get confused between scopes and application roles! Any thoughts?

I want to customize this response body to retrieve a set of custom claims the user has when they successfully generate an access token. An example of this using a .NET Core demo I was playing with looks something like this.

```
{


"bearerToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJQU2hlcmlmZiIsImp0aSI6
IjliMzJhNGJhLTdkOWEtNGQ5MS04NWMwLTA2NGM5MTlkNWNmZCIsIkNhbkFjY2Vzc1Byb2R1Y3RzIjoidHJ1Z
SIsIkNhbkFkZFByb2R1Y3QiOiJ0cnVlIiwiQ2FuU2F2ZVByb2R1Y3QiOiJ0cnVlIiwiQ2FuQWNjZXNzQ2F0ZW
dvcmllcyI6InRydWUiLCJDYW5BZGRDYXRlZ29yeSI6InRydWUiLCJuYmYiOjE1NjU2OTIyNDksImV4cCI6MTU
2NTY5MjMwOSwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdDo1MDAwIiwiYXVkIjoiUFRDVXNlcnMifQ.wLXZ-b7Q-
xu2WWYDCvnKVN_8vurEtkpftjFOAHHu8Fs",

"expires": "13 August 2019 11:31:49",


"claims": [

{


"claimType": "CanAccessProducts",

"claimValue": "true"

},

{


"claimType": "CanAddProduct",

"claimValue": "true"

},

{


"claimType": "CanSaveProduct",

"claimValue": "true"

},

{


"claimType": "CanAccessCategories",

"claimValue": "true"

},

{


"claimType": "CanAddCategory",

"claimValue": "true"

}

]

}
```

So what if I wanted to add these additional claims to the response body. Where can this be done?

I found an interesting comment in the classmethod ##class(%OAuth2.Server.Validate).ValidateUser() that read '*Use the Cache roles for the user to setup a custom property*' and I can see the roles for my user being set via Do properties.CustomProperties.SetAt(roles,"roles") but I can't see the roles being written to the JSON in the response. In addition to roles, I cannot see any of the OpenID Connect claims such as sub, preferredusername, email, updatedat being written to the response body even though they are setup in this classmethod.

Summary

1. How do I customize what is returned in the response body of the token endpoint?
2. Any thoughts on scope design principles vs. user-based application roles?

#OAuth2 #ObjectScript #Caché

Source URL:https://community.intersystems.com/post/adding-custom-claims-oauth-20