Google Cloud

Edge Security

OAuth - Miscellaneous

Refreshing tokens

Refresh tokens can be used to obtain new access tokens for the original end user

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OAuthV2 async="false" continueOnError="false" enabled="true" name="RefreshToken">
    <DisplayName>OAuth Refresh Token</DisplayName>
    <Operation>RefreshAccessToken</Operation>
    <ExpiresIn>86400000</ExpiresIn>
    <ReuseRefreshToken>true</ReuseRefreshToken>
    <GrantType>request.queryparam.grant_type</GrantType>
    <RefreshToken>request.formparam.refresh_token</RefreshToken>
    <GenerateResponse/>
</OAuthV2>
```

```
curl -v -u "vn0zG4cnSWaWIzdwBZgnREI1NGORDXXz:uZt7LeDbq7vX90NP"
-d "grant_type=refresh_token&refresh_token=LRYvVXXgWwbuEZivoe8XvZE2WPdzX9fp"
"http://myorg-test.apigee.net/v1/oauth/refresh_accesstoken"
```

Invalidating access and refresh tokens

Access and refresh tokens can be invalidated via policies
“cascade=true” propagates invalidation to all related access and refresh tokens

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OAuthV2 async="false" continueOnError="false" enabled="true" name="InvalidateToken">
    <DisplayName>OAuth Invalidate Token</DisplayName>
    <Operation>InvalidateToken</Operation>
    <Tokens>
      <Token type="accesstoken" cascade="true">request.queryparam.access_token</Token>
           <Token type="refreshtoken" cascade="true">request.queryparam.refresh_token</
Token>
    </Tokens>
</OAuthV2>
```

Access token attributes

Access/refresh tokens and authorization codes can contain attributes

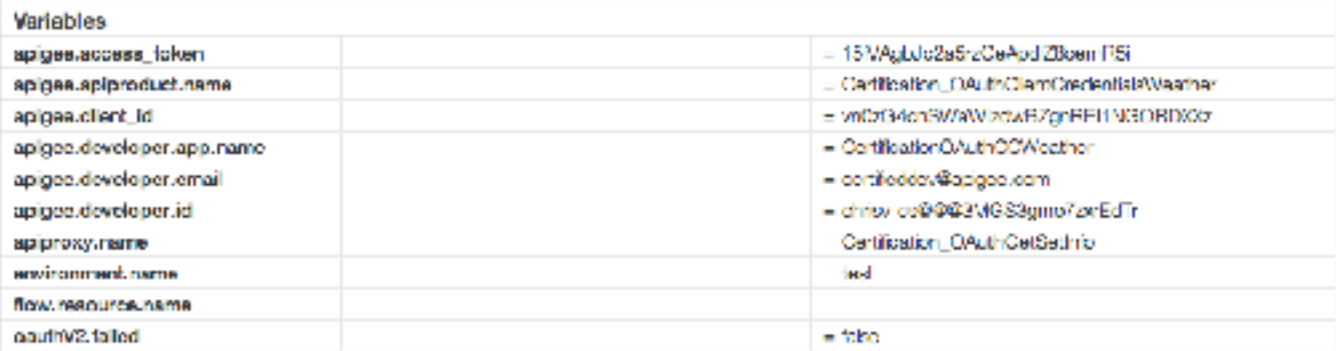
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OAuthV2 async="false" continueOnError="false" enabled="true" name="oauth-generate-token">
    <DisplayName>OAuth Generate  Token</DisplayName>
    <Operation>GenerateAccessToken</Operation>
    <ExpiresIn>86400000</ExpiresIn>
    <SupportedGrantTypes>
      <GrantType>client_credentials</GrantType>
    </SupportedGrantTypes>
    <GrantType>request.queryparam.grant_type</GrantType>
    <Attributes>
        <Attribute name=”attr_name1” ref=”flow.variable” display="true|false">value1</
Attribute>
        <Attribute name=”attr_name2” ref=”flow.variable” display="true|false">value2</
Attribute>
    </Attributes>
</OAuthV2>
```

GetOAuthV2Info to retrieve access token attributes

GetOAuthV2Info used to retrieve general info about an access token

GetOAuthV2Info is usually not needed – the OAuthV2 VerifyAccessToken operation now populates similar variables

```
<GetOAuthV2Info name="GetTokenAttributes">
  <AccessToken ref="request.queryparam.access_token"></AccessToken>
</GetOAuthV2Info>
```



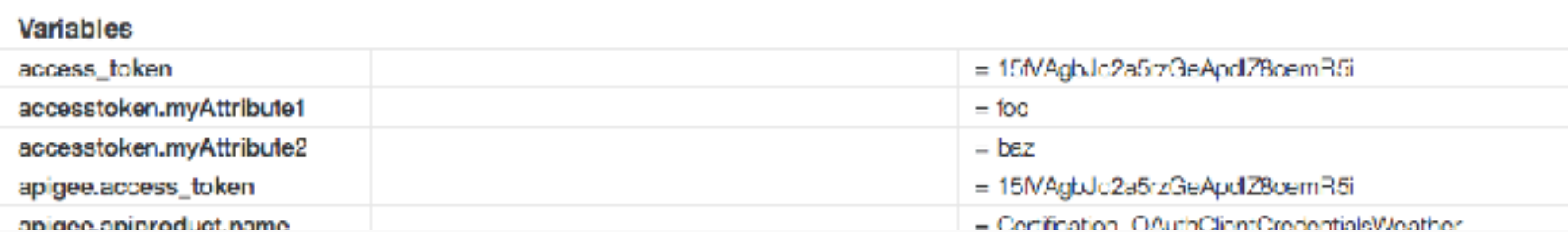
Variables		
apigee.access_token		= 15VAgLJc2a5rzCeApdZBoemF5i
apigee.apiproduct.name		= Certification_OAuthClientCredentialsWeather
apigee.client_id		= [illegible]
apigee.developer.app.name apigee.developer.email apigee.developer.id apiproxy.name		= CertificationOAuthCCWeather = certifieddev@apigee.com = [illegible] Certification_OAuthGetSetInfo
environment.name		test
flow.resource.name		
oauthV2.failed		= false

SetOAuthV2Info to set access token attributes

SetOAuthV2Info can be used to set attributes on an access token...

```
<SetOAuthV2Info name="SetOAuthV2Info">
  <AccessToken ref="request.queryparam.access_token"></AccessToken>
  <Attributes>
    <Attribute name="myAttribute1">foo</Attribute>
    <Attribute name="myAttribute2">baaz</Attribute>
  </Attributes>
</SetOAuthV2Info>
```

...which can then be retrieved by validating the token



Variables

access_token		= 15fVAgbJc2a5rzGeApdZ8oemR5i
accesstoken.myAttribute1		= foo
accesstoken.myAttribute2 apigee.access_token		= baz = 15fVAgbJc2a5rzGeApdZ8oemR5i
apigee.apiproduct.name		= Certification_OAuthClientCredentialsWeather

THANK YOU