Security Programming with Azure Active Directory

Lab Time: 60 minutes

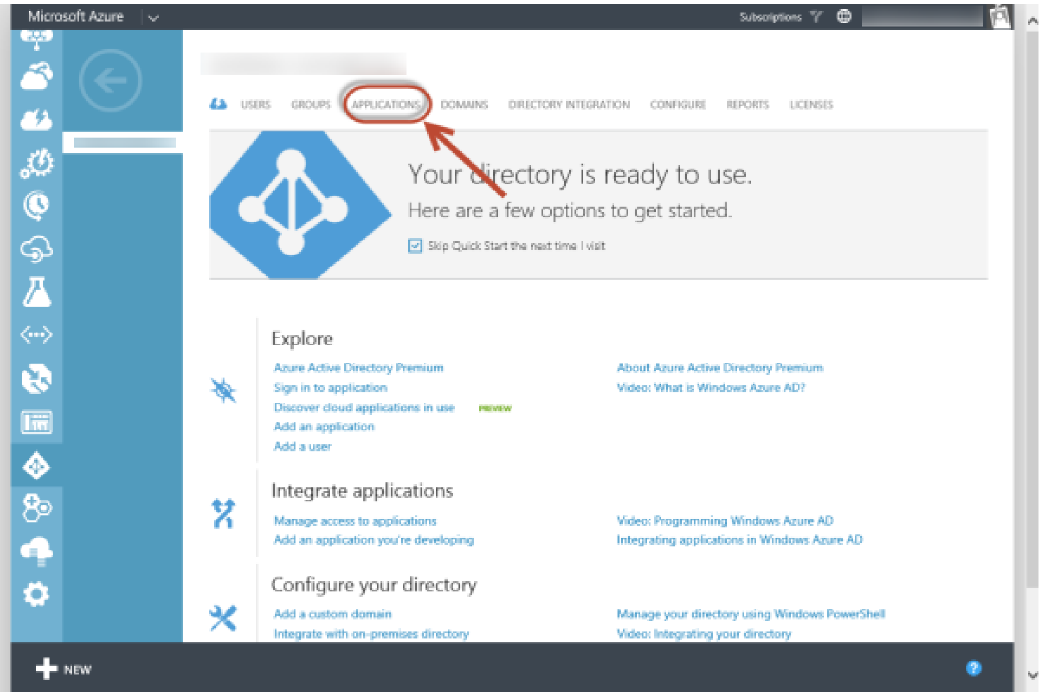
Lab Folder: c:\Student\Modules\AzureActiveDirectory\Labs

Lab Overview: In this lab you will configure your Azure Active Directory (Azure AD) tenant to provide authentication services to various application types.

Exercise 1: Register an application in Azure AD

Consuming O365 APIs requires all requests to be authorized, both the current user and the application. Azure AD provides the authentication for the app and user.

1. Launch Internet Explorer.
2. In Internet Explorer, navigate to **https://manage.windowsazure.com/**.
3. Enter the email address and password for your Office 365 developer account (e.g. student@cptlabs.onmicrosoft.com).
4. In the left-hand navigation, scroll down to and click on Active Directory.
5. Click on the name of the directory associated with the tenancy for your Office 365 developer account to select it and display it. Depending on the state of your portal, you will see the Quick Start page, or the list of Users. On either page, click Applications in the toolbar.


Microsoft Azure
Subscriptions
USERS
GROUPS
APPLICATIONS
DOMAINS
DIRECTORY INTEGRATION
CONFIGURE
REPORTS
LICENSES
Explore
Azure Active Directory Premium
Sign in to application
Discover cloud applications in use
PREVIEW
Add an application
Add a user
About Azure Active Directory Premium
Video: What is Windows Azure AD?
Integrate applications
Manage access to applications
Add an application you're developing
Video: Programming Windows Azure AD
Integrating applications in Windows Azure AD
Configure your directory
Add a custom domain
Integrate with on-premises directory
Manage your directory using Windows PowerShell
Video: Integrating your directory
NEW

Your directory is ready to use.

Here are a few options to get started.

☑ Skip Quick Start the next time I visit

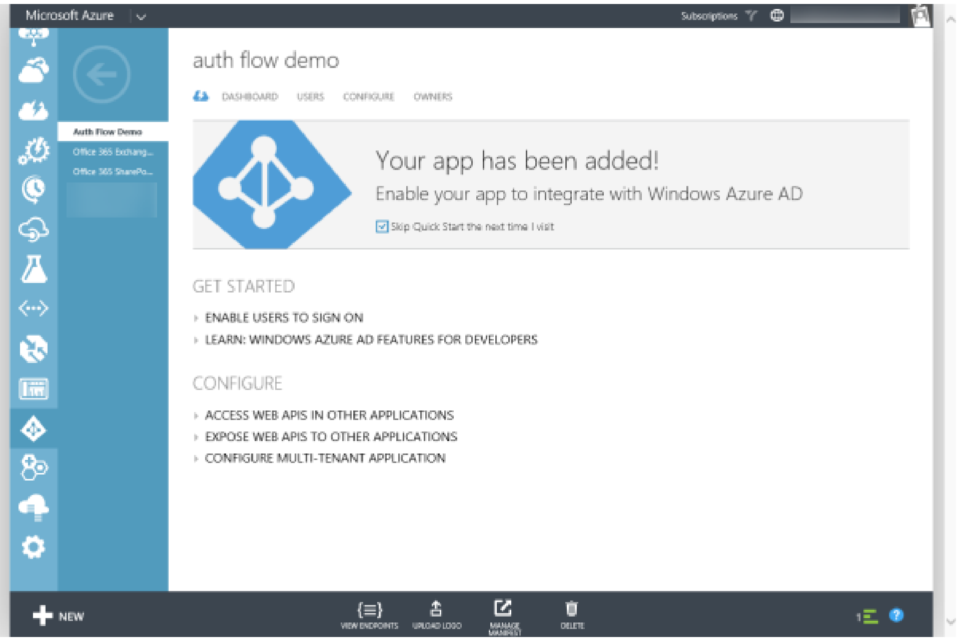
6. You should notice that the list of applications includes **Office 365 Exchange Online** and **Office 365 SharePoint Online**.
7. Click the **Add** button at the bottom of the display.
8. On the **What do you want to do page**, click **Add an application my organization is developing**. This will start the **Add Application** wizard.
9. In the **Add Application** wizard, enter a name of **Auth Flow Demo** and choose the type **Web Application and/or Web API**. Click the arrow to advance to the next page of the wizard.
10. In the App Properties page, enter a **SIGN-ON URL** of **http://authflowdemo.com**.

Remember that the Sign-On URL property is used to redirect the user's browser once authentication is complete. In this lab, we will not actually build a site at that URL. In a later exercise, you will monitor the authentication flow, but will not actually navigate to that URL

11. 10. Enter an **App ID Uri** of **http://**[your-domain]**.onmicrosoft.com/AuthFlowDemo**.

> The App ID Uri must be unique within the Azure tenancy. Using a host name that matches your tenant name helps to prevent confusion, and using a value for the path that matches the app name helps to enforce uniqueness. This value can be changed if the app name or purpose changes.

12. Click the check image in the lower right of the wizard to create the application. The application Quick Start page will display once the application is created.



auth flow demo

Your app has been added!

Enable your app to integrate with Windows Azure AD

☑ Skip Quick Start the next time I visit

13. On the application Quick Start page, click on **CONFIGURE** in the toolbar.

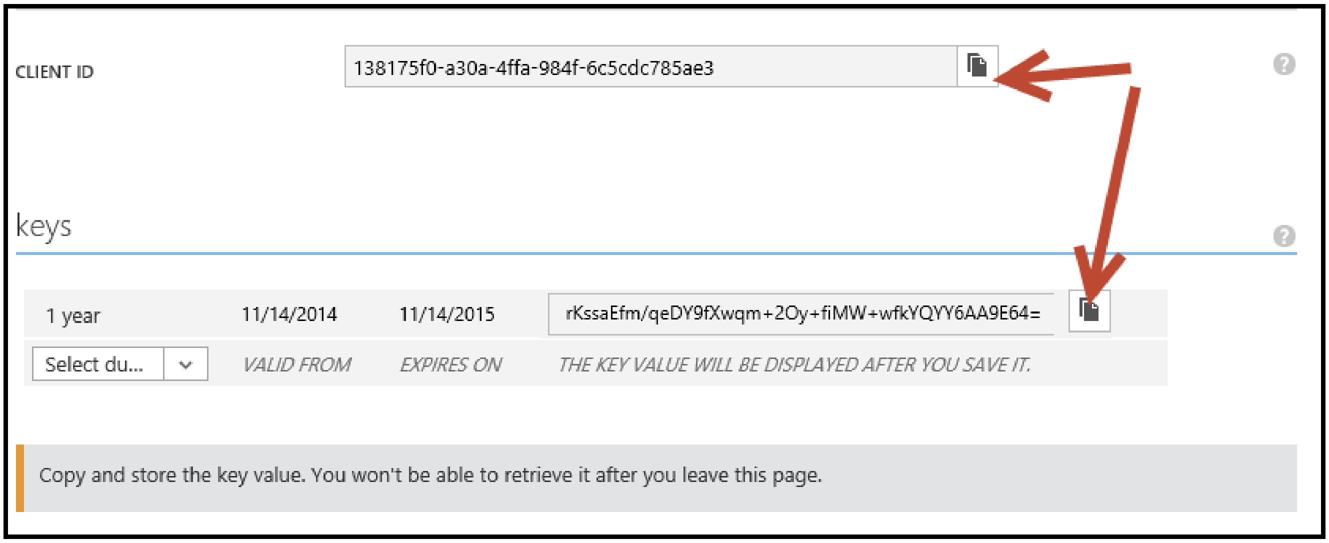
14. Scroll down to the Keys section.

   a) In the **Select Duration** dropdown, select **1 year**.

   b) Then click the **Save** button at the bottom of the page.

> The page will refresh and include the value of the key. In addition, a message is displayed advising that the key will not be shown a second time.

15. For both the Client ID and Key, copy the values to the clipboard and paste into Notepad. We will require these values later.



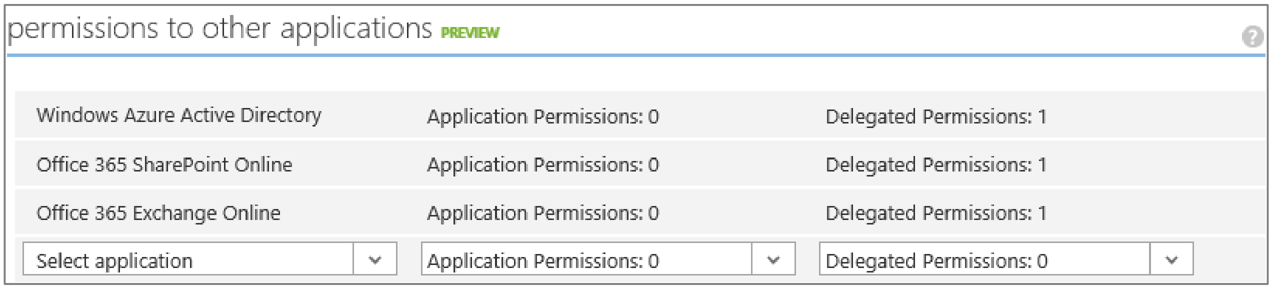
CLIENT ID 138175f0-a30a-4ffa-984f-6c5cdc785ae3

keys

1 year	11/14/2014	11/14/2015	rKssaEfm/qeDY9fXwqm+2Oy+fiMW+wfkYQYY6AA9E64=
Select du...	VALID FROM	EXPIRES ON	THE KEY VALUE WILL BE DISPLAYED AFTER YOU SAVE IT.

Copy and store the key value. You won't be able to retrieve it after you leave this page.

16. Scroll down to the permissions to other applications section.
    a) In the **Select Application** dropdown, select **Office 365 Exchange Online**.
    b) In the **Delegated Permissions** dropdown on the same line, choose **Read users' mail**.
    c) Again, in the **Select Application** dropdown, select **Office 365 SharePoint Online**.
    d) In the **Delegated Permissions** dropdown on the same line, choose **Read users' files**.



permissions to other applications PREVIEW

Windows Azure Active Directory	Application Permissions: 0	Delegated Permissions: 1
Office 365 SharePoint Online	Application Permissions: 0	Delegated Permissions: 1
Office 365 Exchange Online	Application Permissions: 0	Delegated Permissions: 1
Select application	Application Permissions: 0	Delegated Permissions: 0

    e) Click the **Save** button at the bottom of the page.

You have now completed the creation & registration of an application in Azure AD.

Exercise 2: Manually invoke and review the authentication flow

In this exercise, we will manually walk thru the steps of logging in, consenting the application and requesting an access token for an Office 365 service.

1. Launch an InPrivate session of Internet Explorer.
2. Navigate to the Azure AD authorize page.

```
https://login.windows.net/common/oauth2/authorize?resource=Microsoft.SharePoint&redirect_uri=http://authflowdemo.com/&response_type=code&client_id=[CLIENT-ID]
```

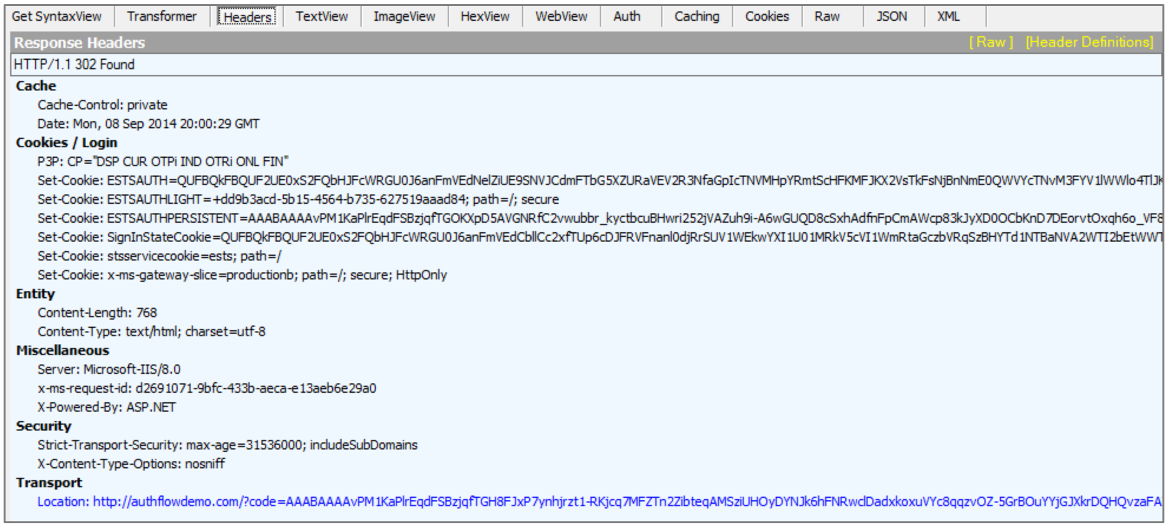
Replace the placeholder [CLIENT-ID] with the actual Client Id copied from the application configuration page.

3. The following are the query string parameters that are part of this request:
    a) **resource**: the Azure AD-secured resource for which authorization is being performed.
    b) **redirect_uri**: the URL to which the authorization code is to be sent upon completion of the authorization & consent process
    c) **response_type**: indicates that the authorization should respond with an authorization code instead of an access token. The Office 365 APIs utilize the code approach, since different tokens are required for the various services (Exchange, SharePoint, etc.). An authorization code can be used to request multiple access tokens.
    d) **client_id**: used to identity the application making the request
4. If prompted, enter the user name and password
5. Before continuing, launch Fiddler. Fiddler will intercept the conversation between the browser and server, providing the ability to view the details of the request & response.
6. In Internet Explorer, click OK to continue. The browser will display a DNS error.

Remember, we configured the application to redirect to a url of http://authflowdemo.com, but we did not create a website at that address. Notice the address bar of Internet Explorer also contains query string parameters.

7. Switch to Fiddler. Near the bottom of the session list will be an entry in red with a result code of 502. Before that, there will be a session with a response code of 302.
8. Select "302" entry.
9. Notice that the Hostname of the entry is login.windows.net.
10. On the right side of the Fiddler display, select the Inspectors tab. The Inspectors tab shows the request from the browser on the top and the response from the server on the bottom.

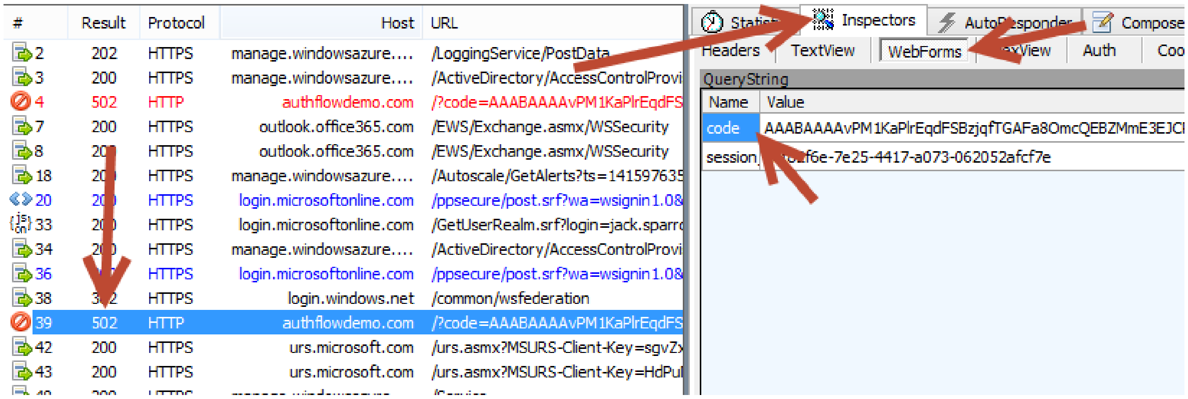
11. In the response inspector (lower half), click the Headers tab. The headers include instructions for the browser to store cookies from Azure AD (login.windows.net) to persist the login as well as a Location header to redirect the browser. The address for the redirect is the Sign-On Url of the application in Azure AD. The redirect request includes query string parameters.



```
HTTP/1.1 302 Found
Cache
    Cache-Control: private
    Date: Mon, 08 Sep 2014 20:00:29 GMT
Cookies / Login
    P3P: CP="DSP CUR OTPi IND OTRi ONL FIN"
    Set-Cookie: ESTSAUTH=QUFBQkFBQUF2UE0xS2FQbHJFcWRGU0J6anFmVEdNelZiUE9SNVJCdmFTbG5XZURaVEV2R3NfaGpIcTNVMHpYRmtScHFKMFJKX2VsTkFsNjBnNmE0QWVYcTNvM3FYV1lWWlo4TlJK
    Set-Cookie: ESTSAUTHLIGHT=+dd9b3acd-5b15-4564-b735-627519aaad84; path=/; secure
    Set-Cookie: ESTSAUTHPERSISTENT=AAABAAAAvPM1KaPlrEqdFSBzjqfTGOKXpD5AVGNRfC2vwubbr_kyctbcuBHwri252jVAZuh9i-A6wGUQD8cSxhAdfnFpCmAWcp83kJyXD0OCbKnD7DEorvtOxqh6o_VF8
    Set-Cookie: SignInStateCookie=QUFBQkFBQUF2UE0xS2FQbHJFcWRGU0J6anFmVEdCbllCc2xfTUp6cDJFRVFnanl0djRrSUV1WEkwYXI1U01MRkV5cVI1WmRtaGczbVRqSzBHYTd1NTBaNVA2WTI2bEtWWT
    Set-Cookie: stsservicecookie=ests; path=/
    Set-Cookie: x-ms-gateway-slice=productionb; path=/; secure; HttpOnly
Entity
    Content-Length: 768
    Content-Type: text/html; charset=utf-8
Miscellaneous
    Server: Microsoft-IIS/8.0
    x-ms-request-id: d2691071-9bfc-433b-aeca-e13aeb6e29a0
    X-Powered-By: ASP.NET
Security
    Strict-Transport-Security: max-age=31536000; includeSubDomains
    X-Content-Type-Options: nosniff
Transport
    Location: http://authflowdemo.com/?code=AAABAAAAvPM1KaPlrEqdFSBzjqfTGH8FJxP7ynhjrzt1-RKjcq7MFZTn2ZibteqAMSziUHOyDYNJk6hFNRwclDadxkoxuVYc8qqzvOZ-5GrBOuYYjGJXkrDQHQvzaFA
```

12. In the Fiddler session list, select the "502" entry which should be the next one immediately after the 302.

    a) In the Inspectors tab, view the Request (upper half) WebForms tab. This will display the query string parameters in a grid format. Notice a code parameter - this is the authorization code returned by AzureAD for this request. The code will be unique to the application and user.



#	Result	Protocol	Host	URL
2	202	HTTPS	manage.windowsazure....	/LoggingService/PostData
3	200	HTTPS	manage.windowsazure....	/ActiveDirectory/AccessControlProvi
4	502	HTTP	authflowdemo.com	/?code=AAABAAAAvPM1KaPlrEqdFS
7	200	HTTPS	outlook.office365.com	/EWS/Exchange.asmx/WSSecurity
8	200	HTTPS	outlook.office365.com	/EWS/Exchange.asmx/WSSecurity
18	200	HTTPS	manage.windowsazure....	/Autoscale/GetAlerts?ts=141597635
20	200	HTTPS	login.microsoftonline.com	/ppsecure/post.srf?wa=wsignin1.0&
33	200	HTTPS	login.microsoftonline.com	/GetUserRealm.srf?login=jack.sparro
34	200	HTTPS	manage.windowsazure....	/ActiveDirectory/AccessControlProvi
36	[illegible]	HTTPS	login.microsoftonline.com	/ppsecure/post.srf?wa=wsignin1.0&
38	302	HTTPS	login.windows.net	/common/wsfederation
39	502	HTTP	authflowdemo.com	/?code=AAABAAAAvPM1KaPlrEqdFS
42	200	HTTPS	urs.microsoft.com	/urs.asmx?MSURS-Client-Key=sgvZx
43	200	HTTPS	urs.microsoft.com	/urs.asmx?MSURS-Client-Key=HdPul

Name	Value
code	AAABAAAAvPM1KaPlrEqdFSBzjqfTGAFa8OmcQEBZMmE3EJCl
session_	[illegible]f6e-7e25-4417-a073-062052afcf7e

13. Copy this code to Notepad. The code will be required in the next step.

> Now it's time to request the access token.

14. In Fiddler, click on the Composer tab.
15. Change the verb to POST
16. Change the url to:

```
https://login.windows.net/common/oauth2/token
```

17. Add the following to Request Headers.

```
Content-Type: application/x-www-form-urlencoded
```

18. The Request Body must be UrlEncoded, so there is a preparatory step before completing the form.
19. Launch Windows PowerShell ISE
20. Run the following cmdlets:

```
[System.Reflection.Assembly]::LoadWithPartialName("System.Web") | out-null
```

```
[System.Web.HttpUtility]::UrlEncode("[CLIENT-SECRET]")
```

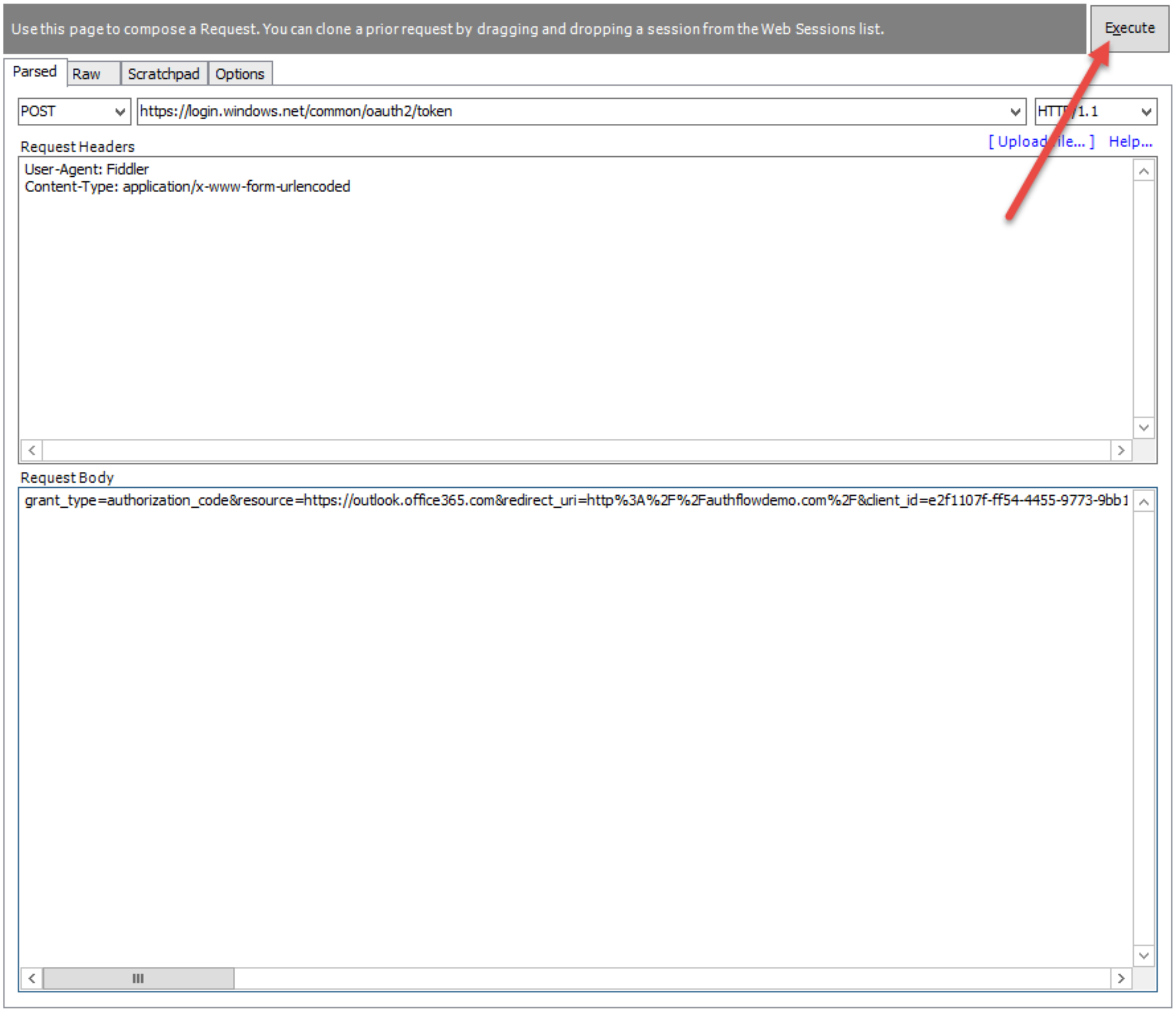
Replace the placeholder [CLIENT-SECRET] with the actual key copied from the application configuration page previously.

21. The UrlEncode method will output the encoded client secret. Copy this value to Notepad.
22. The Request Body contains the following parameters that are required to get an access token.
    a) grant_type: specifies that we are providing an authorization code instead of requesting the user to enter credentials.
    b) resource: the resource to which an access token is requested.
    c) redirect_uri: the URL to which the access token is to be sent upon validation of the request
    d) client_id: used to identity the application making the request
    e) client_secret: used to authenticate the application.
23. Enter the following in the Request Body, replacing the [tokens] with values as described below.

```
grant_type=authorization_code&resource=https://outlook.office365.com&
redirect_uri=http%3A%2F%2Fauthflowdemo.com%2F&
client_id=[CLIENT-ID]&client_secret=[CLIENT-SECRET]
&code=[AUTHORIZATION-CODE]
```

Keep this in mind when replacing the token. Make sure line breaks are removed... the Request Body should be on a single line. Line breaks were added in the code above for readability. The client_id is copied from the application configuration page in the Azure portal. The client_secret is coped from the UrlEncode method in Windows PowerShell ISE. The code id copied from the Fiddler Inspector tab of the "502" request.

24. Click the Execute button to issue the request and get the access token.



Use this page to compose a Request. You can clone a prior request by dragging and dropping a session from the Web Sessions list.

POST https://login.windows.net/common/oauth2/token HTTP/1.1

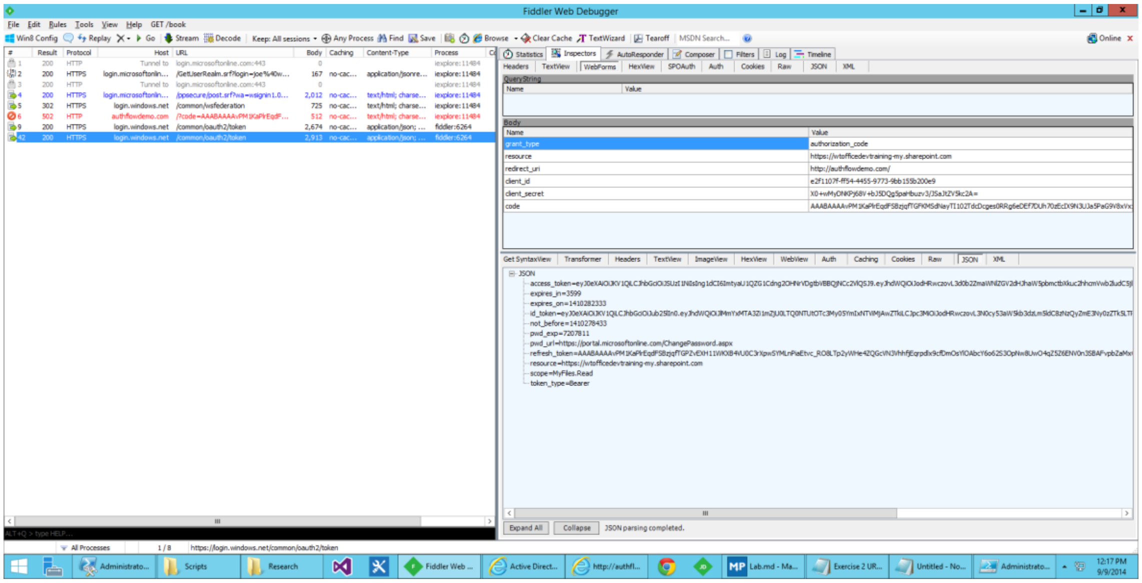
Request Headers

```
User-Agent: Fiddler
Content-Type: application/x-www-form-urlencoded
```

Request Body

```
grant_type=authorization_code&resource=https://outlook.office365.com&redirect_uri=http%3A%2F%2Fauthflowdemo.com%2F&client_id=e2f1107f-ff54-4455-9773-9bb1
```

25. In the session list, click on the just-issued request, then click on the Inspectors tab.
26. In the Response window (lower half), click the JSON tab. Notice that the response contains an access token and a refresh token.
27. Click on the Composer tab.
28. In the RequestBody, change the resource to **https://[your-domain]-my.sharepoint.com**. Click Execute.



Fiddler Web Debugger
File Edit Rules Tools View Help GET /book
Win8 Config Replay Go Stream Decode Keep: All sessions Any Process Find Save Browse Clear Cache TextWizard Tearoff MSDN Search... Online
Statistics Inspectors AutoResponder Composer Filters Log Timeline
Headers TextView WebForms HexView SPOAuth Auth Cookies Raw JSON XML
Get SyntaxView Transformer Headers TextView ImageView HexView WebView Auth Caching Cookies Raw JSON XML
Expand All Collapse
ALT+Q > type HELP...
All Processes 1 / 8

#	Result	Protocol	Host	URL	Body	Caching	Content-Type	Process
1	200	HTTP	Tunnel to	login.microsoftonline.com:443	0			iexplore:11484
2	200	HTTPS	login.microsoftonlin...	/GetUserRealm.srf?login=joe%40w...	167	no-cac...	application/jsonre...	iexplore:11484
3	200	HTTP	Tunnel to	login.microsoftonline.com:443	0			iexplore:11484
4	200	HTTPS	login.microsoftonlin...	/ppsecure/post.srf?wa=wsignin1.0...	2,012	no-cac...	text/html; charse...	iexplore:11484
5	302	HTTPS	login.windows.net	/common/wsfederation	725	no-cac...	text/html; charse...	iexplore:11484
6	502	HTTP	authflowdemo.com	/?code=AAABAAAAvPM1KaPlrEqdF...	512	no-cac...	text/html; charse...	iexplore:11484
9	200	HTTPS	login.windows.net	/common/oauth2/token	2,674	no-cac...	application/json; ...	fiddler:6264
42	200	HTTPS	login.windows.net	/common/oauth2/token	2,913	no-cac...	application/json; ...	fiddler:6264

QueryString

Name	Value

Body

Name	Value
grant_type	authorization_code
resource	https://wtofficedevtraining-my.sharepoint.com
redirect_uri	http://authflowdemo.com/
client_id	e2f1107f-ff54-4455-9773-9bb155b200e9
client_secret	X0+wMyDNKPj68V+bJ5DQg5paHbuzv3/JSaJtZV5kc2A=
code	AAABAAAAvPM1KaPlrEqdFSBzjqfTGFKMSdNayTI102TdcDcges0RRg6eDEf7DUh70zEcDX9N3UJa5PaG9V8xVx

```
JSON
  access_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6ImtyaU1QZG1Cdng2OHNrVDgtbVBBQjNCc2VlQ5J9.eyJhdWQiOiJodHRwczovL3d0b2ZmaWNlZGV2dHJhaW5pbmctbXkuc2hhcmVwb2ludC5j
  expires_in=3599
  expires_on=1410282333
  id_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJhdWQiOiJlMmYxMTA3Zi1mZjU0LTQ0NTUtOTc3My05YmIxNTViMjAwZTkiLCJpc3MiOiJodHRwczovL3N0cy53aW5kb3dzLm5ldC8zNzQyZmE3Ny0zZTk5LTI
  not_before=1410278433
  pwd_exp=7207811
  pwd_url=https://portal.microsoftonline.com/ChangePassword.aspx
  refresh_token=AAABAAAAvPM1KaPlrEqdFSBzjqfTGPZvEXH11WKXB4VU0C3rXpwSYMLnPiaEtvc_RO8LTp2yWHe4ZQGcVN3VhhfjEqrpdlx9cfDmOsYlOAbcY6o62S3OpNw8UwO4qZ5Z6ENV0n3SBAFvpbZaMx
  resource=https://wtofficedevtraining-my.sharepoint.com
  scope=MyFiles.Read
  token_type=Bearer
```

JSON parsing completed.

https://login.windows.net/common/oauth2/token

You have successfully retrieved an access code for two different resources using the same authorization code.