WorkshopPLUS – Modern Authentication and Authorization

OAuth2 and OpenIDConnect

Labs prerequisites –

Web browser

Fiddler

Postman

Student Lab Manual

Instructor Edition (Book Title Hidden Style)

Version 2.0

**Conditions and Terms of Use**

**Microsoft Confidential - For Internal Use Only**

This training package is proprietary and confidential, and is intended only for uses described in the training materials. Content and software is provided to you under a Non-Disclosure Agreement and cannot be distributed. Copying or disclosing all or any portion of the content and/or software included in such packages is strictly prohibited.

The contents of this package are for informational and training purposes only and are provided "as is" without warranty of any kind, whether express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

Training package content, including URLs and other Internet Web site references, is subject to change without notice. Because Microsoft must respond to changing market conditions, the content should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

© 2021 Microsoft Corporation. All rights reserved.

**Copyright and Trademarks**

© 2021 Microsoft Corporation. All rights reserved.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

For more information, see Use of Microsoft Copyrighted Content at  
[*http*://www.microsoft.com/about/legal/permissions/](http://www.microsoft.com/about/legal/permissions/)

Microsoft®, Internet Explorer®, and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other Microsoft products mentioned herein may be either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners.

Contents

[Introduction 6](#_Toc65427374)

[Exercise 1: Get access token for an application 7](#_Toc65427375)

[Exercise 2. Get an access token for a user 8](#_Toc65427376)

# Introduction

#### Objectives

In this lab students will use common two common http tools: Postman and Fiddler to acquire OAuth2 tokens and call an API (MS Graph).

#### Registration data

You will be using a web browser, Postman to obtain, view and use OAuth2/OIDC tokens. A set of applications has already been registered with an Azure AD tenant. Details of the tenant and application registrations are provided below.

|  |  |
| --- | --- |
| **Property** | **Value** |
| *URL of authz server metadata url* | <https://login.microsoftonline.com/0a56b010-8d4f-4deb-ba4f-006cb7af65a1/v2.0/.well-known/openid-configuration> |
| *Tenant id* | 0a56b010-8d4f-4deb-ba4f-006cb7af65a1 |
| *Confidential (web) client id* | ca676371-37a9-44ad-92de-e615015453c4 |
| *Confidential client URI* | https://jwt.ms/ |
| *Confidential client secret* | TTx8Q~T7NdkEe3X6Jby2rh7ECIY3xkEwTsZBichB |
| *Confidential client redirect uri* | https://oidcdebugger.com/debug |
| *Public client id* | b87304e8-8487-4548-b229-db761adf48fd |
| *Public client callback url* | app://publicclient1 |
| *User id* | [simon@ocag547outlook.onmicrosoft.com](mailto:simon@ocag547outlook.onmicrosoft.com) |
| *User password* | \*Al0ndr4123\* |
| *User id* | user4@modernauthn.onmicrosoft.com |
| *User password* | Phoenix04 |
| *Graph Resource Id* | https://graph.microsoft.com |

OAuth2 specification: <https://tools.ietf.org/html/rfc6749>

#### Estimated time to complete this lab

45 minutes

# Exercise 1: Get access token for an application

1. Install Postman desktop client (web version cannot be used with Fiddler) from: <https://www.postman.com/downloads/>
2. Enter the following http request in Postman:

GET <https://graph.microsoft.com/v1.0/users>

1. Press Send to execute the request. You should get 401 Unauthorized response
2. Select Authorization menu option. It appears below the request url line.
3. Select the OAuth2 authorization type from the presented options. Make sure that ‘Add authorization data’ has ‘Request headers’ selected.
4. Set the following values in the presented dialog (leave default for others):
   1. Grant type: Client Credentials
   2. Access token url: get this url by viewing the .well-known metadata in a browser (find that url in the table above). You will need to find the token endpoint.
   3. Client ID: copy the confidential client id from the above table.
   4. Client Secret: copy the client secret from the above table
   5. Enter <https://graph.microsoft.com/.default> for scope. This tells AAD to issue a token with whatever permissions are configured for this application for the Graph resource.
5. For Client Authentication select ‘Send client credentials in body’
6. In Postman, scroll to the bottom of the token request configuration dialog and select Get New Access Token. If the request succeeds, select Use Token. This will insert the token into an Authorization request for your Graph request.
7. Select Send to make the Graph request.
8. View the requests for both the token and Graph in *View->Show Postman console*. You can also use that view to investigate any errors that may have occurred. (Postman has its own Console window to view requests. However, it is not as detailed as that provided by Fiddler).

**Note: Fiddler with https encryption enabled MAY block some authentication challenges, reporting SSL errors. It typically happens when a server is doing a user credentials challenge. If that happens, stop Fiddler tracing, complete the challenge and then restart Fiddler.**

# Exercise 2. Get an access token for a user

1. Enter the following http request in Postman:

GET <https://graph.microsoft.com/v1.0/me>

1. Press Send to execute the request. You should get 401 Unauthorized response
2. Select Authorization menu option. It appears below the request url line.
3. Select the OAuth2 authorization type from the presented options.
4. Select the following values in the presented dialog:
   1. Grant type: Authorization code (with PKCE)
   2. Callback url: see above table for Public client callback url
   3. Authorization URL and Token URL: use the metadata url from the above table to determine the values for these urls. *authorization\_endpoint* will be part of that document.
   4. Uncheck the *Authorize using browser* checkbox
   5. Client ID: copy the public client id from the above table.
   6. Client Secret: make sure there is none – this is a public client!
   7. Enter “https://graph.microsoft.com/User.Read.All openid profile email offline\_access” for scope. This tells AAD to issue a token with whatever permissions are configured for this application for the Graph resource.
5. In Postman, scroll to the bottom of the token request configuration dialog and select Get New Access Token. Use user credentials provided in the above table. If the request succeeds, select Use Token. This will insert the token into an Authorization request for your Graph request.
6. Select Send to make the Graph request.
7. View the requests for both the token and Graph in Postman’s View->Show Postman Console feature. You should see the initial request for the authorization code, the subsequent POST exchanging the code for an access token and your request to Graph using that token.
8. View the tokens (id and access) in <https://jwt.ms>

|  |
| --- |
| Extra challenges:   1. What changes if you check *Authorize using browser* button? How do you resolve any issues that creates? 2. How do the contents of the token request or the contents of the token change if you drop some of scopes (in 15 g, e.g. do not use *offline\_access*). |