SECURE OPEN ID CONNECT IMPLEMENTATION USING AZURE ACTIVE DIRECTORY AND ASP .NET FRAMEWORK

PETRO KOLOSOV

Abstract. In this manuscript secure Open ID Connect implementation using Azure is discussed.

Contents

1. Introduction

Your introduction here. Include some references [1, 2, 3, 4, 5]. Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more

Date: May 17, 2023.

Key words and phrases. Open ID Connect, OIDC, Azure Active Directory, PKCE, OAuth 2.0, XSS, CSRF, ASP .NET Core .

recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

OAuth 2.0 flow diagram

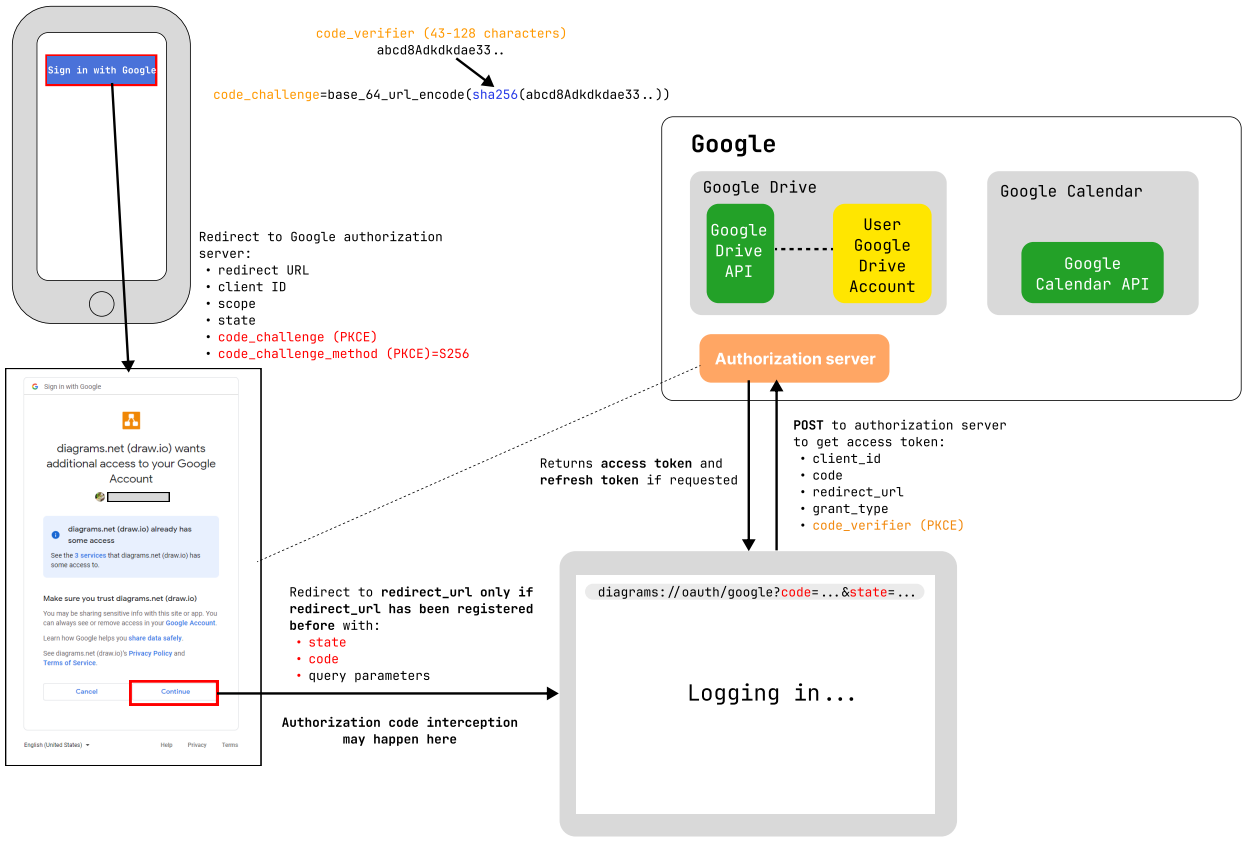

Figure 1. OAuth 2.0 with PKCE flow diagram.

2. Statement of the problem

3. Authentication flow

Authentication flow diagram

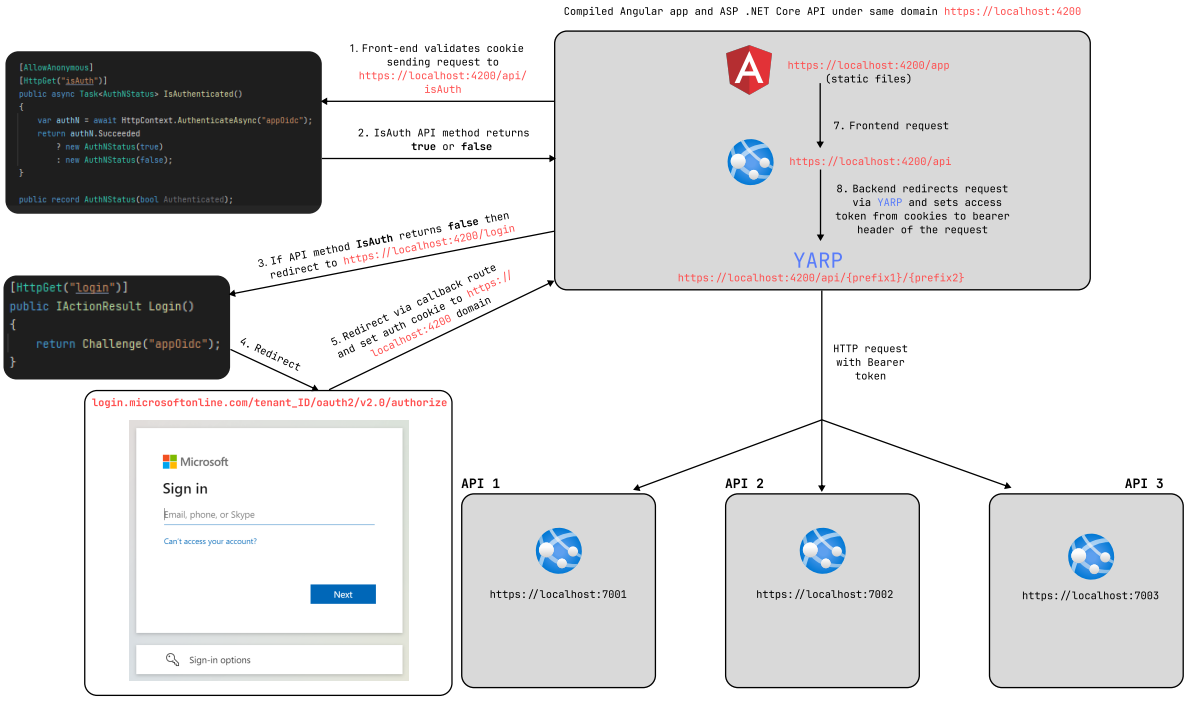

Figure 2. Authentication flow diagram.

(1) Frontend application sends request to `https://localhost:4200/api/isAuth` API method to validate authentication

(2) API method `https://localhost:4200/api/isAuth` returns `true` if authentication valid, otherwise `false`

(3) If authentication invalid then browser redirects to `https://localhost:4200/login`, otherwise no action taken

(4) Login redirects browser to authorize url `login.microsoftonline.com/tenant/oauth2/v2.0/authorize`

(5) If user is logged in then browser is redirected to fallback url with cookies already set for `https://localhost:4200` domain

(6) Request is sent to validate authentication `https://localhost:4200/api/isAuth`, now it returns `true`

(7) Frontend at `https://localhost:4200/app` sends request to the `https://localhost:4200/api/OtherApi1/products`

(8) Cookie exists for `https://localhost:4200/app` so that `https://localhost:4200/api` attaches it as header to request via YARP and sends request with token to the external resource `OtherApi1/products`

(9) If response code is 401 at step (8) then repeat step (1)

4. Refresh token flow

5. Conclusions

Conclusions of your manuscript.

6. Acknowledgements

Thanks someone and somebody for help and useful comments.

References

[1] Mohd Shadab Siddiqui and Deepanker Verma. Cross site request forgery: A common web application weakness. In *2011 IEEE 3rd International Conference on Communication Software and Networks*, pages 538–543. IEEE, 2011.

[2] Kevin Spett. Cross-site scripting. *SPI Labs*, 1(1):20, 2005.

[3] J Bradley and N Agarwal. Rfc 7636: Proof key for code exchange by oauth public clients, 2015.

[4] Dick Hardt. The oauth 2.0 authorization framework. Technical report, 2012.

[5] Roy Fielding and Julian Reschke. Rfc 7231: Hypertext transfer protocol (http/1.1): semantics and content, 2014.

Email address: `kolosovp94@gmail.com`

URL: `https://kolosovpetro.github.io`