การเชื่อมต่อใช้งาน ระบบพิสูจน์ตัวตนแบบหลายปัจจัย ผ่าน OAuth2/OpenID Connect

(Authorization code flow)

By : S.Thitipol

หัวข้อหลัก

1. Multi-factor Authentication คืออะไร

2. Factor มีอะไรบ้าง

3. มาตรฐาน RFC ที่เกี่ยวข้อง

4. ผังของระบบ

5.การนำไปใช้งาน

Multi-factor Authentication คืออะไร

- Multi-Factor Authentication (MFA)
- คือการใช้ปัจจัยหลายๆ อย่างในการตรวจสอบและยืนยันตัวบุคคล เพื่ออนุญาตเข้าใช้งาน ซอฟต์แวร์, ระบบ หรือข้อมูลต่างๆ

- โดยทั่วไประบบ MFA จะใช้เครื่องมือ (Factor) ตั้งแต่ 2 อย่างขึ้นไปในการตรวจสอบและยืนยันความถูกต้อง

Multi-factor Authentication (ChatGPT)

tell me standard of MFA.

MFA stands for "Multi-Factor Authentication," which is a security measure used to protect digital accounts and devices by requiring users to provide multiple forms of authentication in order to access their accounts or devices.

The standard for MFA varies depending on the platform or service being used, but generally it involves the use of at least two of the following three factors:

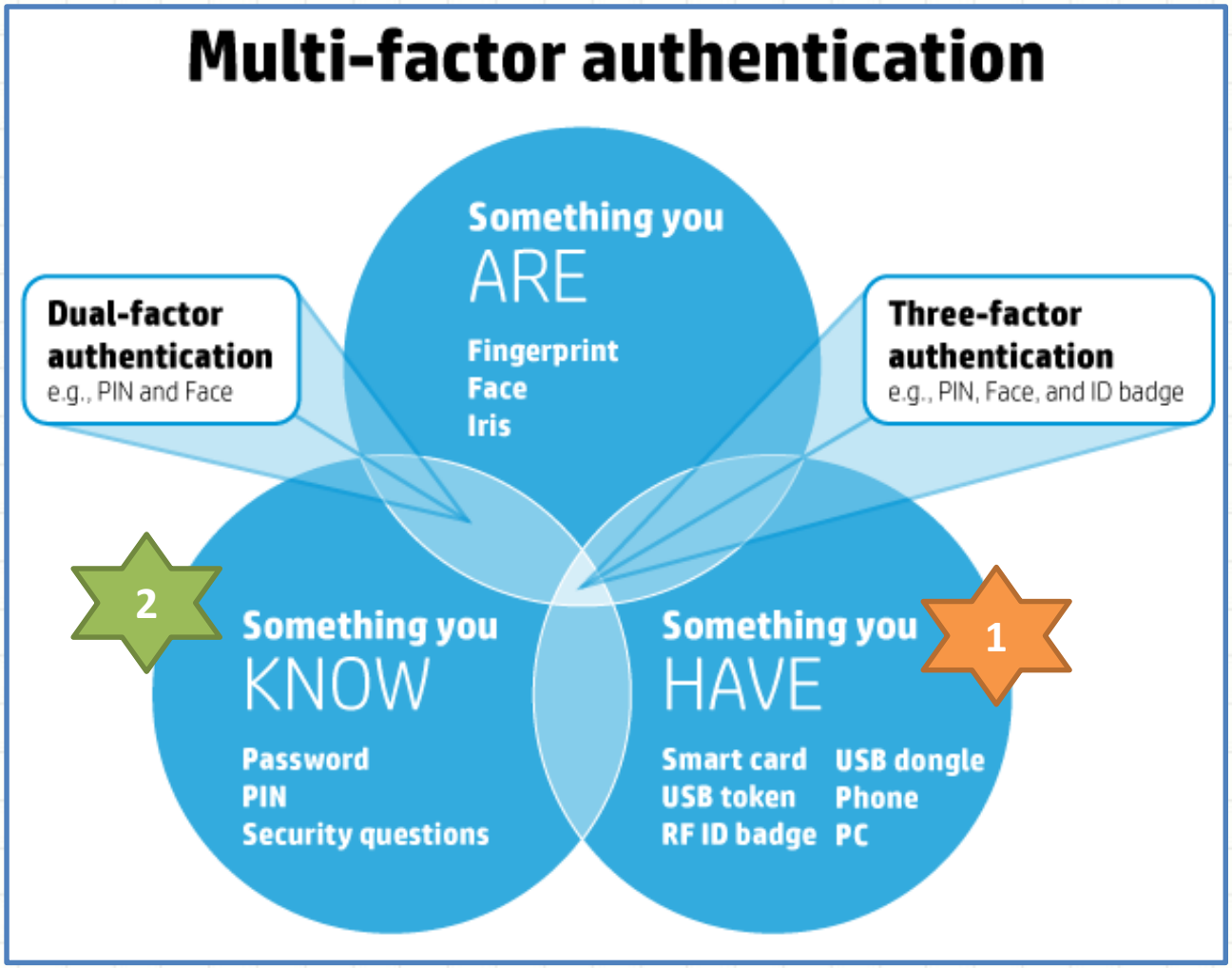
1. Something the user knows, such as a password, PIN, or security question.
2. Something the user has, such as a smart card, mobile phone, or authentication token.
3. Something the user is, such as a fingerprint, facial recognition, or other biometric factor.

In addition to requiring multiple factors of authentication, MFA also typically involves the use of strong encryption, secure communication protocols, and other security measures to ensure that the authentication process is secure and resistant to attacks by hackers and other malicious actors.

Factor มีอะไรบ้าง

- What you know (สิ่งที่คุณรู้) : เช่น password, รหัสประจำตัว, หรือ คำถามเฉพาะเพื่อกู้รหัสผ่าน ★
- What you have (สิ่งที่คุณมี) : เช่น บัตรสมาร์ทการ์ด, หรือ สิ่งที่เราสามารถพกติดตัวไว้ด้วยตลอด ★★
- Who you are (สิ่งที่คุณเป็น) : เช่น ลายนิ้วมือ หรือ ระบบจดจำใบหน้า ★★★
- What you do and where you are (สิ่งที่คุณทำ หรือที่ที่คุณอยู่) : เช่น การระบุที่อยู่ โดยใช้ GPS, IP Address

Multi-Factor Authentication



มาตรฐาน RFC ที่เกี่ยวข้อง (1)

การเชื่อมต่อ client - server	RFC 2818: HTTP Over TLS
RADIUS service server	RFC 2865: Remote Authentication Dial In User Service
การตรวจสอบสิทธิ์ client - server	RFC 7519: JSON Web Token (JWT)
การเข้ารหัสข้อมูล client - server	RFC 3447: Public-Key Cryptography Standards
การยืนยันตัวตน client - server	RFC 6979: Deterministic Usage of the Digital Signature

มาตรฐาน RFC ที่เกี่ยวข้อง (2)

Full Duplex Communication	RFC 6455: The WebSocket Protocol
OTP (One-Time Password)	RFC 6238: TOTP : Time-Based One-Time Password
QR code standard	RFC 9238: Loading Manufacturer Usage Description
Web Authentication	RFC 8809: Registries for Web Authentication (WebAuthn)

มาตรฐาน RFC (3)

Authorization Framework	RFC 6749: The OAuth 2.0 Authorization Framework
OpenID Connect	Identity layer on top of the OAuth 2.0 protocol https://openid.net/connect

หัวข้อย่อย

การนำไปใช้งาน

หัวข้อย่อย

1. ศัพท์ที่เกี่ยวข้อง
2. Grant Type : Authorization Code
3. API ที่เกี่ยวข้อง
4. การนำไปใช้งาน
5. ตัวอย่างการนำไปใช้งานกับ Application

ศัพท์ที่เกี่ยวข้อง

1. Resource Owner (User)

คือ ตัวเรา ซึ่งเป็นเจ้าของ resources ต่าง ๆ ในระบบ

2. Application (Client)

คือ application หรือ third-party application ที่ต้องการ access ข้อมูลของเรา เช่น web server, single page application, mobile, desktop ฯลฯ

ศัพท์ที่เกี่ยวข้อง

3. Resource Server (HRIS)
คือ server ที่ทำหน้าที่ในการให้บริการข้อมูล เช่น server ที่ทำหน้าที่เป็น API server หรือ server ที่เก็บข้อมูลของเราไว้

4. Authorization Server (OpenID server)
คือ server ที่ทำหน้าที่ในการ grant authorize (ยก/อนุญาต สิทธิ์เรา) ให้ client หรือ application ใด ๆ สามารถที่จะ acccess ข้อมูลของเราได้ ตาม scope ที่เรากำหนด

ศัพท์ที่เกี่ยวข้อง

5. Authorization Code

เป็นการ grant authorize ด้วยการ return authorization_code ผ่านทาง query string url เพื่อให้ application ปลายทาง นำ authorization_code นั้นไปขอ access_token จาก authorization server ต่อไป

6. Client Credentials

ใช้ client_id และ client_secret ในการขอ access_token ตรง ๆ จาก authorization server วิธีการนี้ อยู่นอกเหนือบริบทของ user คือ user ไม่ได้เป็นคน grant authorize ด้วยตัวเอง (หลังบ้านเป็นคนทำเอง)

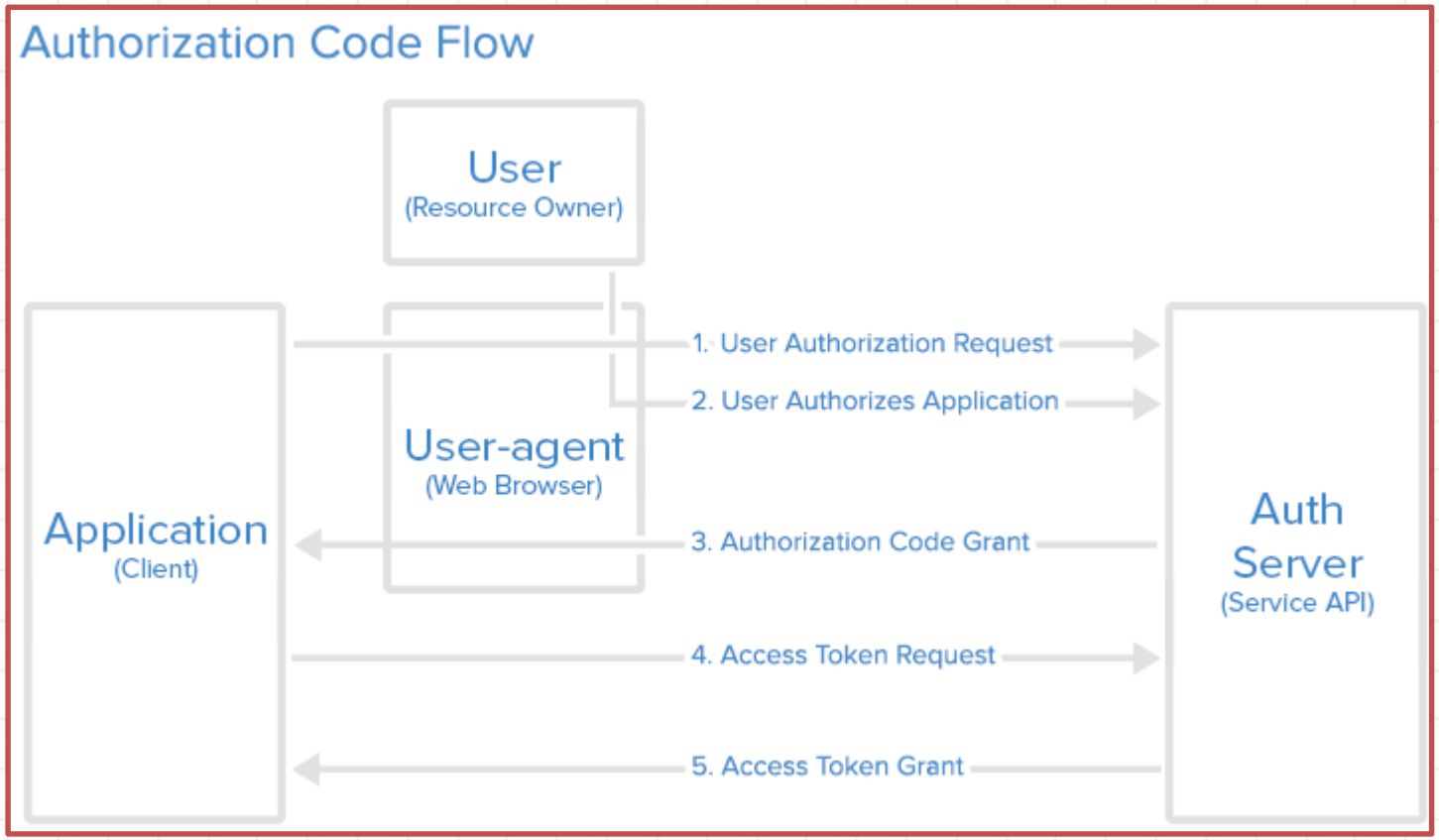
Response type : Authorization Code

Authorization code flow

The authorization server will respond with a `code`, which the client can exchange for tokens on a secure channel. This flow should be used when the application code runs on a secure server (common for MVC and server-rendered pages apps).

Response type : Authorization Code

Authorization Code ใช้สำหรับ Web Server ที่ใช้ Code ด้านหลังในการเชื่อมต่อกับ OpenID Connect Server โดยไม่ได้เปิดเผยให้สาธารณะเห็น



หัวข้อย่อย

API ที่เกี่ยวข้อง

หัวข้อย่อย

การนำไปใช้งาน

สิ่งที่ต้องรู้ก่อน

1. client_id & client_secret

client_id และ *client_secret* ออกให้โดย ศซว.ทอ. เพื่อใช้ติดต่อ OpenID connect server

2. redirect_uri

เป็น Home callback url สำหรับเมื่อ Authentication เสร็จแล้วจะมาหน้าที่กำหนดนี้

3. post_logout_redirect_uri

เป็น Logout callback url สำหรับเมื่อ logout เสร็จแล้วจะมาหน้าที่กำหนดนี้

สิ่งที่ต้องรู้ก่อน

4. เก็บ id_token ไว้ที่ฝั่ง Front-end เพื่อใช้กับ *End session-endpoint API* เพื่อที่หลังจาก logout จะได้ redirect url กลับไปหน้าที่กำหนด

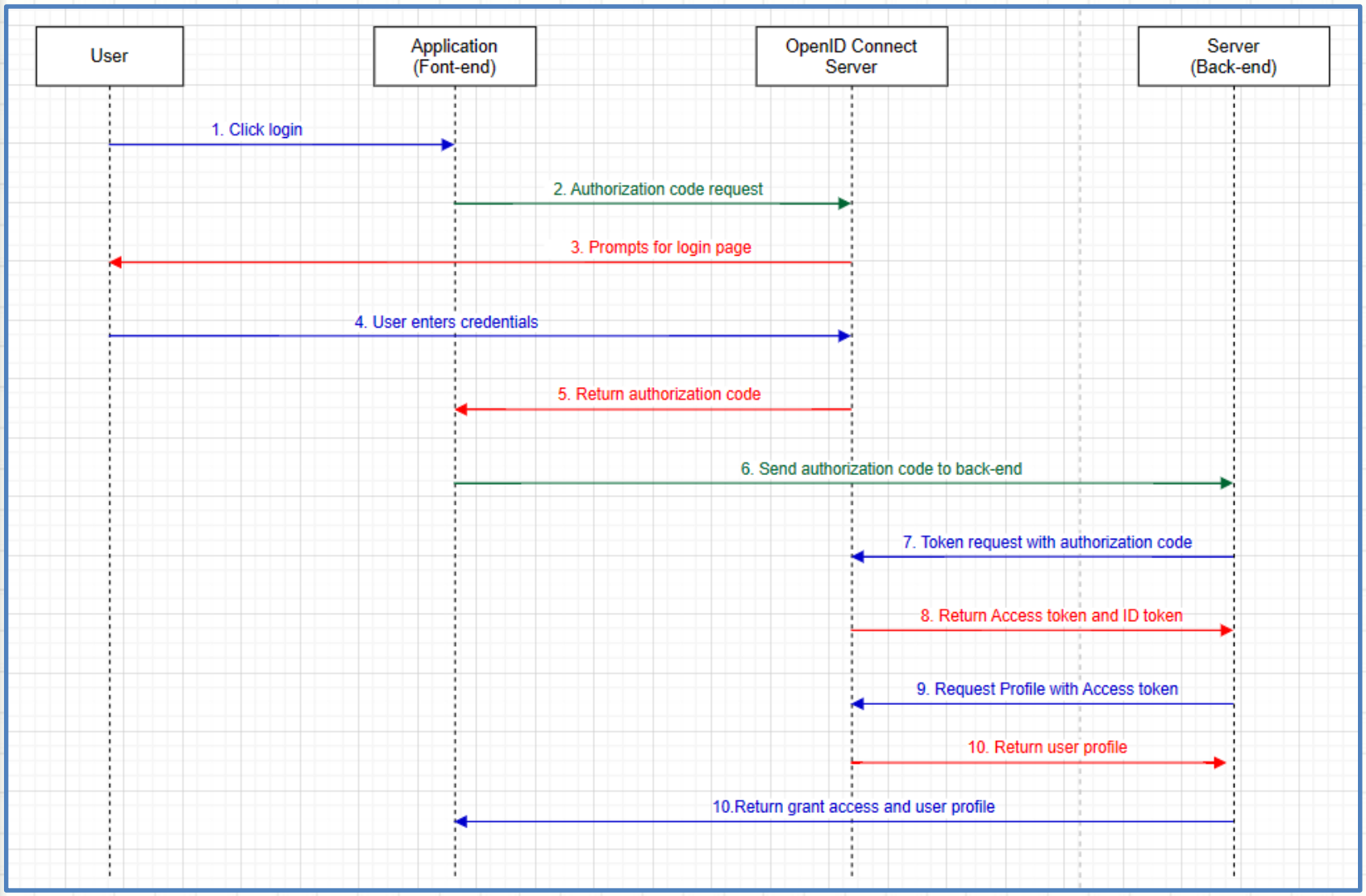
5. ใช้ *Token-endpoint API* และ *Userinfo-endpoint API* ที่ Back-end server และไม่ควรเปิดเผย *client_secret* และ *access_token* ให้ใครรู้

6. *access_token* expired ในเวลา 24 ชม.

7. Back-end server ควรสร้าง Token ต่างหากเพื่อไว้อ้างอิงกับ *access_token*

7. ศึกษา *Flow การทำงาน* ก่อนลง code

Flow การทำงาน



หัวข้อย่อย

ตัวอย่างการนำไปใช้งานกับ Application JavaScript

ขั้นตอนฝั่ง Front-end

1. สร้าง url เพื่อใช้ redirect ไปหน้า login ของ OpenID connect

	Function : Authorization-endpoint API		
	รายการข้อมูล	รายละเอียด	
1	client_id	(ศชว.ทอ. เป็นผู้ออกให้)	Require
2	redirect_uri	Home url callback	Require
3	scope	ระบุ : “openid profile”	Require
4	response_type	ระบุ : “code”	Require
5	response_mode	ระบุ : “query”	Require
6	state	State is there to protect the end user from cross site request forgery(CSRF) attacks	ค่าอะไรก็ได้
7	nonce	String value used to associate a Client session with an ID Token	ค่าอะไรก็ได้

ขั้นตอนฝั่ง Front-end

```
this.swc_openid_login_url =
'https://openid.rtaf.mi.th/oidc/auth?client_id='+client_id+'&redirect_uri='+redirect_uri+'&scope=openid%20profile&response_type=code&response_mode=query&state='+state+'&nonce='+nonce
```

https://openid.rtaf.mi.th/oidc/auth?client_id=[redacted]1&redirect_uri=h

ขั้นตอนฝั่ง Front-end

2. เมื่อ user click button เพื่อ login ให้ redirect ไปหน้าที่กำหนด

```
window.location.href = this.swc_openid_login_url
```

JS

3-4. หลังจาก user enters credentials เสร็จ OpenID connect จะ redirect กลับมาที่หน้าที่กำหนดไว้ใน redirect_uri

ขั้นตอนฝั่ง Front-end

5. Authorization code จะถูกส่งมาใน Address bar ของ Front-end ในกรณี Credentials failed จะส่ง error มามาเช่นกัน

https://otp.rtaf.mi.th/login@/?code=UIqfbwnquYrhqdwdvTx3jaavftBasf

https://otp.rtaf.mi.th/login@/?error=access_denied&error_description=Passv

ขั้นตอนฝั่ง Front-end

6. Front-end ส่ง Authorization code กลับที่ Back-end เพื่อให้ Back-end ขอ Access token, User profile จาก OpenID connect server

```
let urlParams = new URLSearchParams(window.location.search);
let code = urlParams.get('code')

let _profile = await this.fn__swc_openid_get_profile (code ,
redirect_uri )
```

ขั้นตอนฝั่ง Back-end

7. Back-end ส่ง Authorization code ไปที่ OpenID connect server เพื่อขอ Access token และ ID token

	Function : Token-endpoint API		
	รายการข้อมูล	รายละเอียด	
1	client_id	(ศชว.ทอ. เป็นผู้ออกให้)	Require
2	client_secret	(ศชว.ทอ. เป็นผู้ออกให้)	Require
3	grant_type	ระบุ : “authorization_code”	Require
4	redirect_uri	Home url callback	Require
5	code	ได้มาหลังจาก login สำเร็จ (ในaddress bar)	Require

ขั้นตอนฝั่ง Back-end

8-9. หลังจาก OpenID connect server ส่ง Access token, ID token กลับมา ให้ Back-end ส่ง Access token กลับไปที่ OpenID connect server อีกครั้งเพื่อขอ User profile

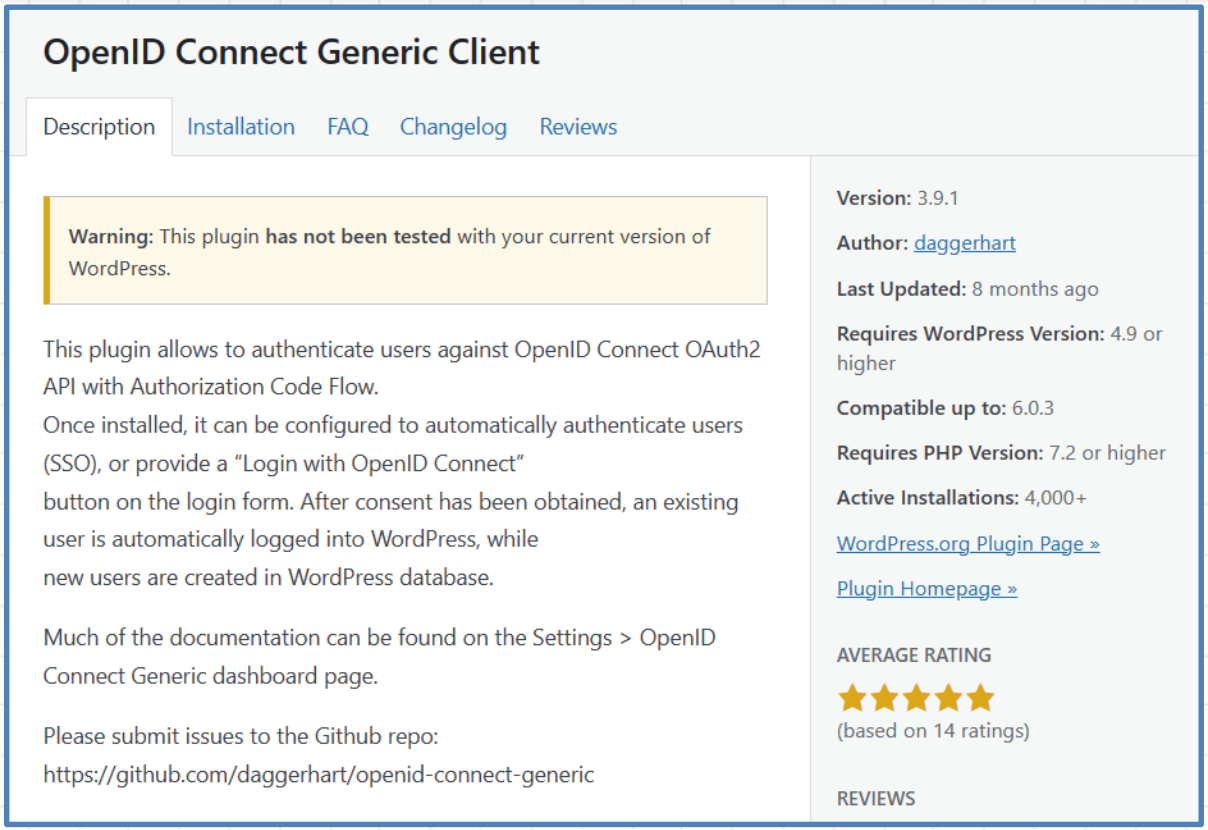
	Function : Userinfo-endpoint API		
	รายการข้อมูล	รายละเอียด	
1	access_token	ได้จากการใช้ Token-endpoint API	Require

หัวข้อย่อย

ตัวอย่างการนำไปใช้งานกับ 3rd WordPress

การนำไปใช้งานกับ 3rd WordPress

1. login เข้า WordPress ในสิทธิ์ของ root , install plugin ที่ชื่อว่า OpenID Connect Generic Client



OpenID Connect Generic Client

Description | Installation | FAQ | Changelog | Reviews

Warning: This plugin **has not been tested** with your current version of WordPress.

This plugin allows to authenticate users against OpenID Connect OAuth2 API with Authorization Code Flow.
Once installed, it can be configured to automatically authenticate users (SSO), or provide a "Login with OpenID Connect" button on the login form. After consent has been obtained, an existing user is automatically logged into WordPress, while new users are created in WordPress database.

Much of the documentation can be found on the Settings > OpenID Connect Generic dashboard page.

Please submit issues to the Github repo:
https://github.com/daggerhart/openid-connect-generic

Version: 3.9.1

Author: daggerhart

Last Updated: 8 months ago

Requires WordPress Version: 4.9 or higher

Compatible up to: 6.0.3

Requires PHP Version: 7.2 or higher

Active Installations: 4,000+

WordPress.org Plugin Page »

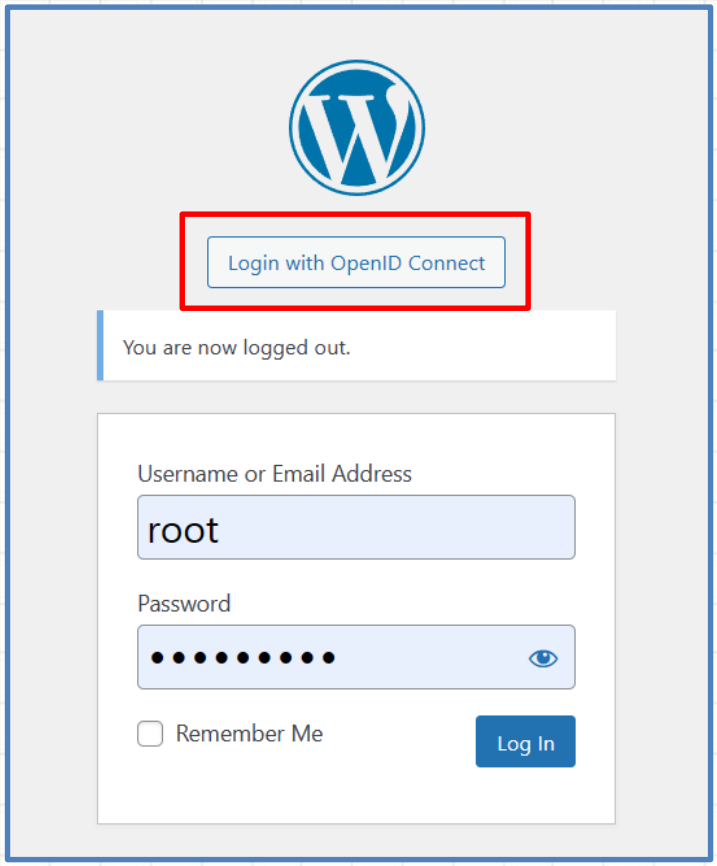
Plugin Homepage »

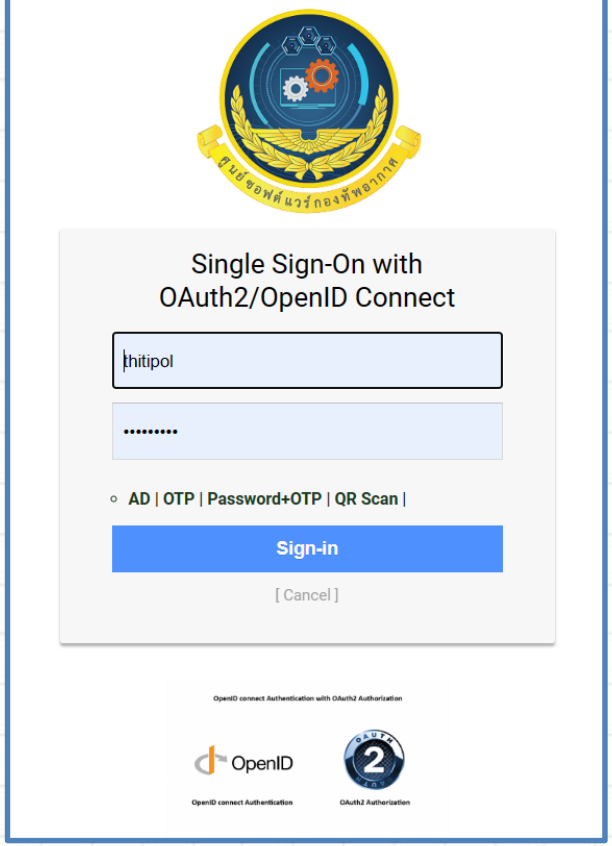
AVERAGE RATING

★★★★★
(based on 14 ratings)

REVIEWS

การนำไปใช้งานกับ 3rd WordPress





การเชื่อมต่อใช้งาน ระบบพิสูจน์ตัวตนแบบหลายปัจจัย ผ่าน OAuth2/OpenID Connect

(Authorization code flow)

By : S.Thitipol