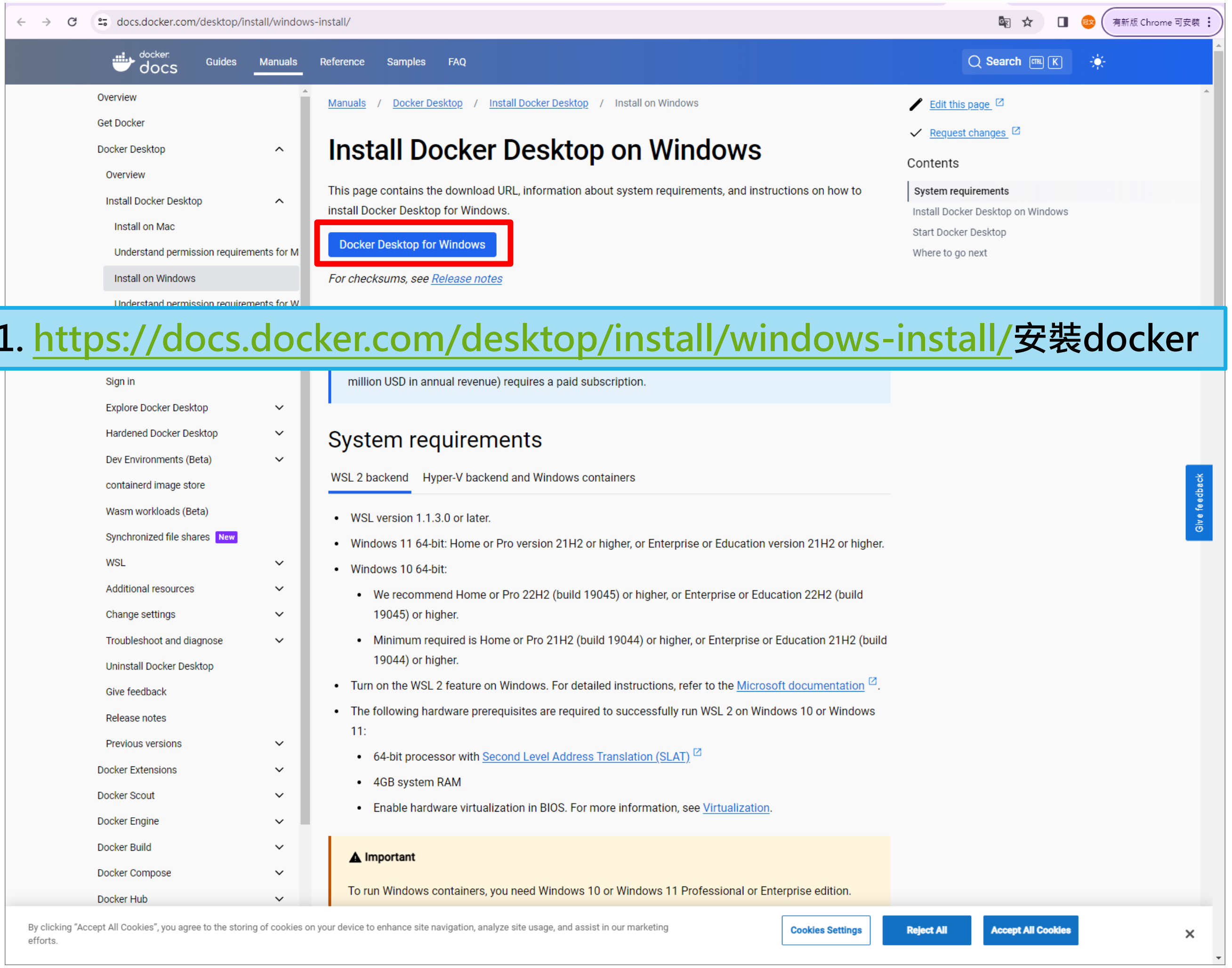
Keycloak

WINDOWS

1. 準備一台Windows環境的電腦(不能使用虛擬機或Server)

2. 安裝docker



Manuals / Docker Desktop / Install Docker Desktop / Install on Windows

Install Docker Desktop on Windows

This page contains the download URL, information about system requirements, and instructions on how to install Docker Desktop for Windows.

Docker Desktop for Windows

For checksums, see Release notes

million USD in annual revenue) requires a paid subscription.

System requirements

WSL 2 backend | Hyper-V backend and Windows containers

- WSL version 1.1.3.0 or later.
- Windows 11 64-bit: Home or Pro version 21H2 or higher, or Enterprise or Education version 21H2 or higher.
- Windows 10 64-bit:
  - We recommend Home or Pro 22H2 (build 19045) or higher, or Enterprise or Education 22H2 (build 19045) or higher.
  - Minimum required is Home or Pro 21H2 (build 19044) or higher, or Enterprise or Education 21H2 (build 19044) or higher.
- Turn on the WSL 2 feature on Windows. For detailed instructions, refer to the Microsoft documentation.
- The following hardware prerequisites are required to successfully run WSL 2 on Windows 10 or Windows 11:
  - 64-bit processor with Second Level Address Translation (SLAT)
  - 4GB system RAM
  - Enable hardware virtualization in BIOS. For more information, see Virtualization.

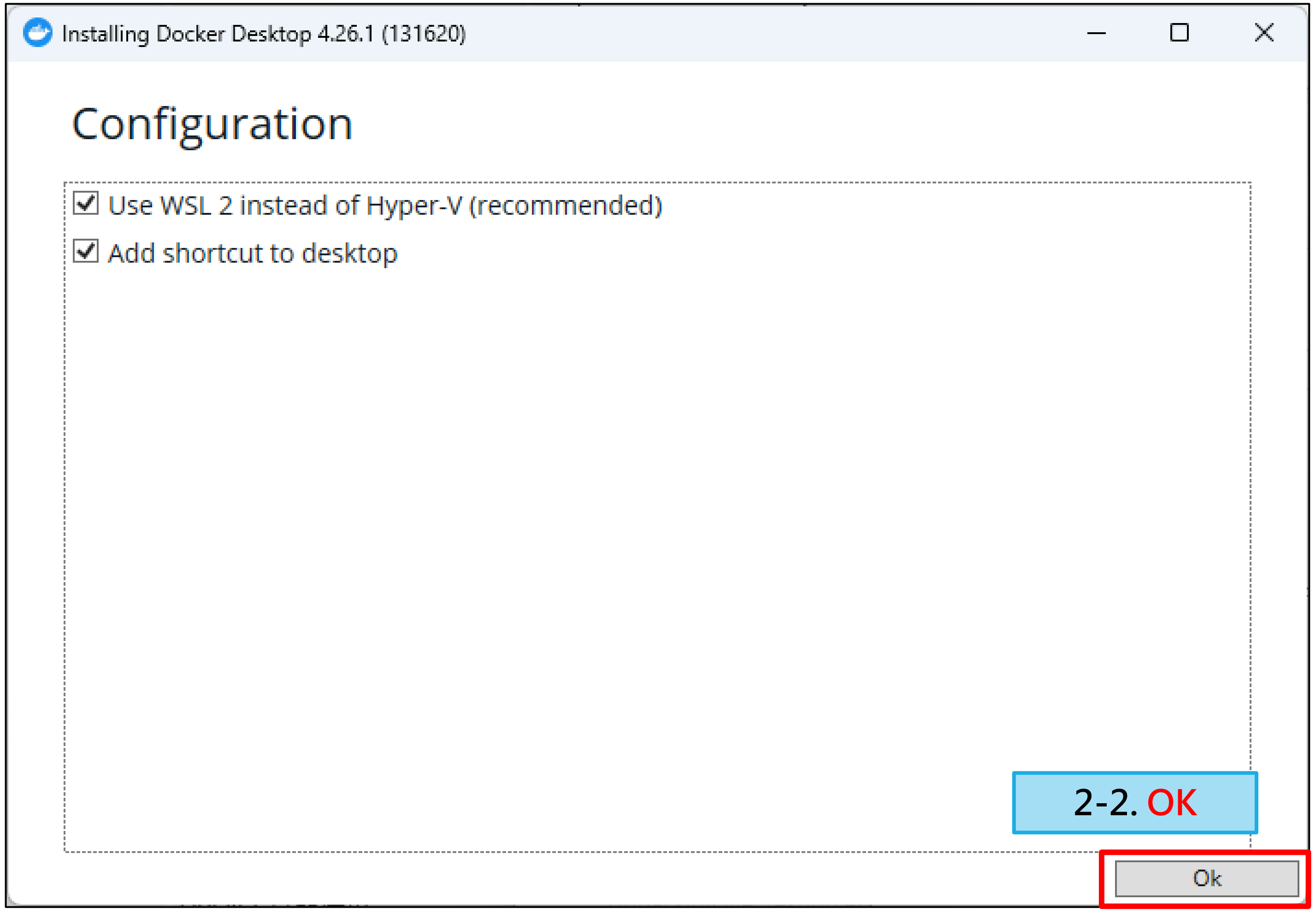
Important

To run Windows containers, you need Windows 10 or Windows 11 Professional or Enterprise edition.

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.

2-1. https://docs.docker.com/desktop/install/windows-install/安裝docker

2. 安裝docker



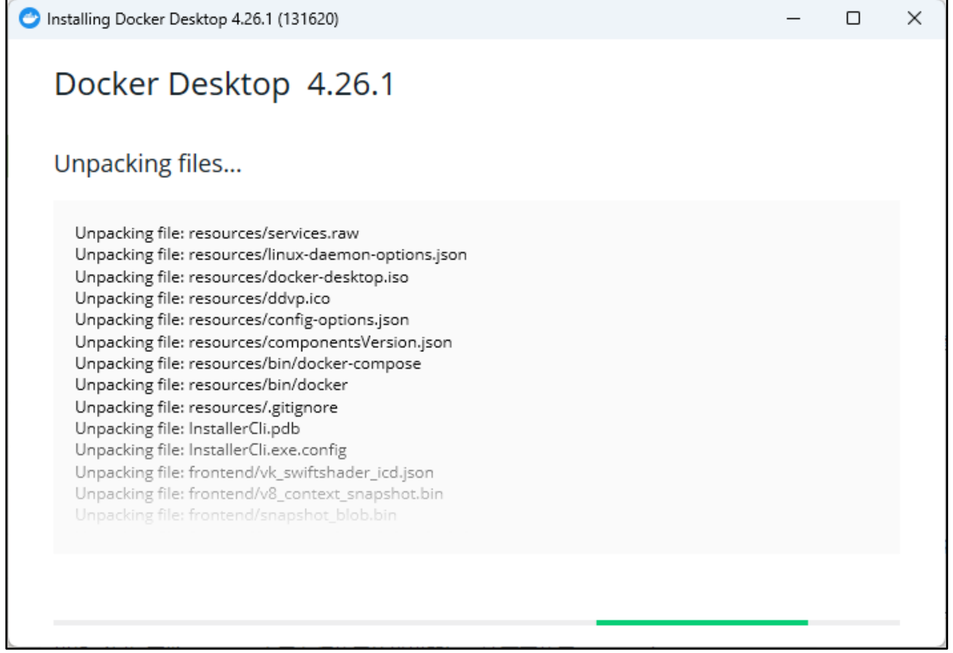
Installing Docker Desktop 4.26.1 (131620)

Configuration

☑ Use WSL 2 instead of Hyper-V (recommended)

☑ Add shortcut to desktop

2-2. OK

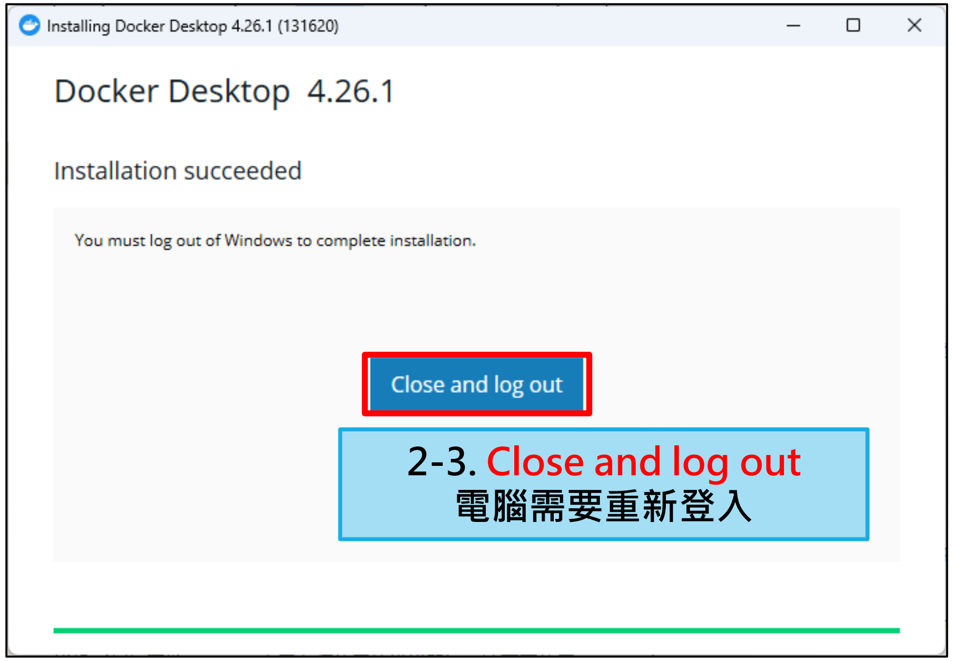


Installing Docker Desktop 4.26.1 (131620)

Docker Desktop 4.26.1

Unpacking files...

```
Unpacking file: resources/services.raw
Unpacking file: resources/linux-daemon-options.json
Unpacking file: resources/docker-desktop.iso
Unpacking file: resources/ddvp.ico
Unpacking file: resources/config-options.json
Unpacking file: resources/componentsVersion.json
Unpacking file: resources/bin/docker-compose
Unpacking file: resources/bin/docker
Unpacking file: resources/.gitignore
Unpacking file: InstallerCli.pdb
Unpacking file: InstallerCli.exe.config
Unpacking file: frontend/vk_swiftshader_icd.json
Unpacking file: frontend/v8_context_snapshot.bin
Unpacking file: frontend/snapshot_blob.bin
```



Installing Docker Desktop 4.26.1 (131620)

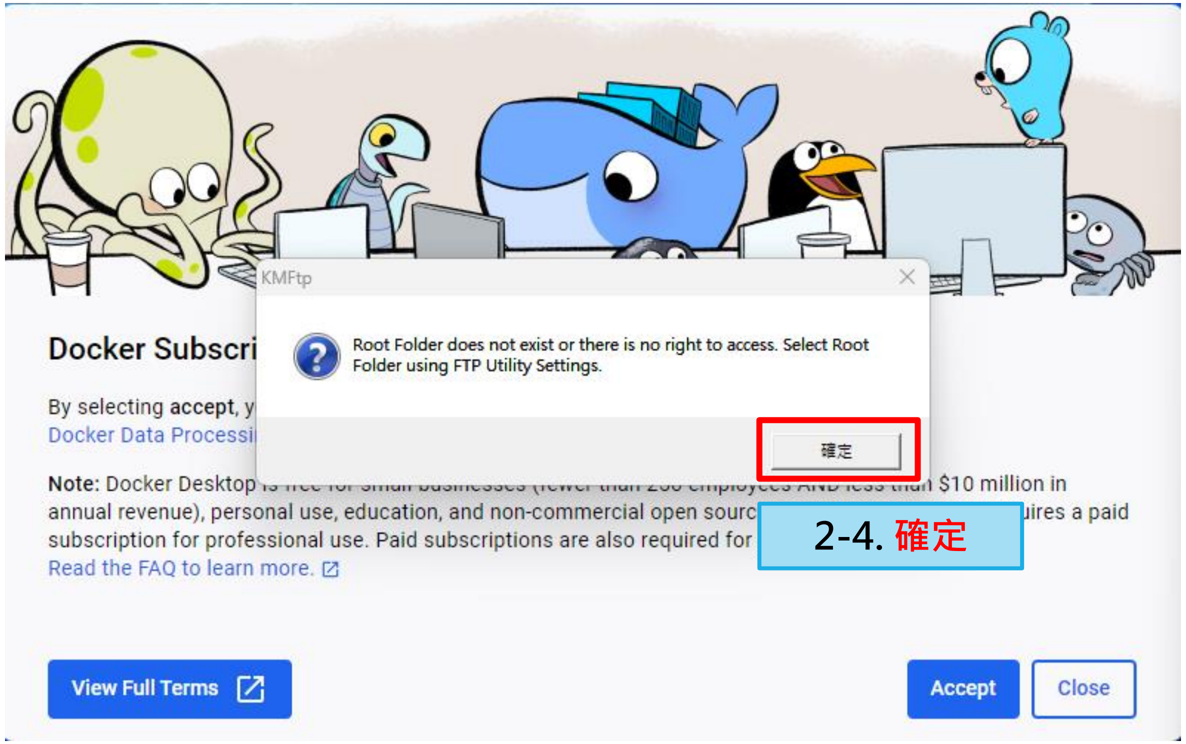
Docker Desktop 4.26.1

Installation succeeded

You must log out of Windows to complete installation.

2-3. Close and log out
電腦需要重新登入

2. 安裝docker



KMFtp

Root Folder does not exist or there is no right to access. Select Root Folder using FTP Utility Settings.

Docker Subscri

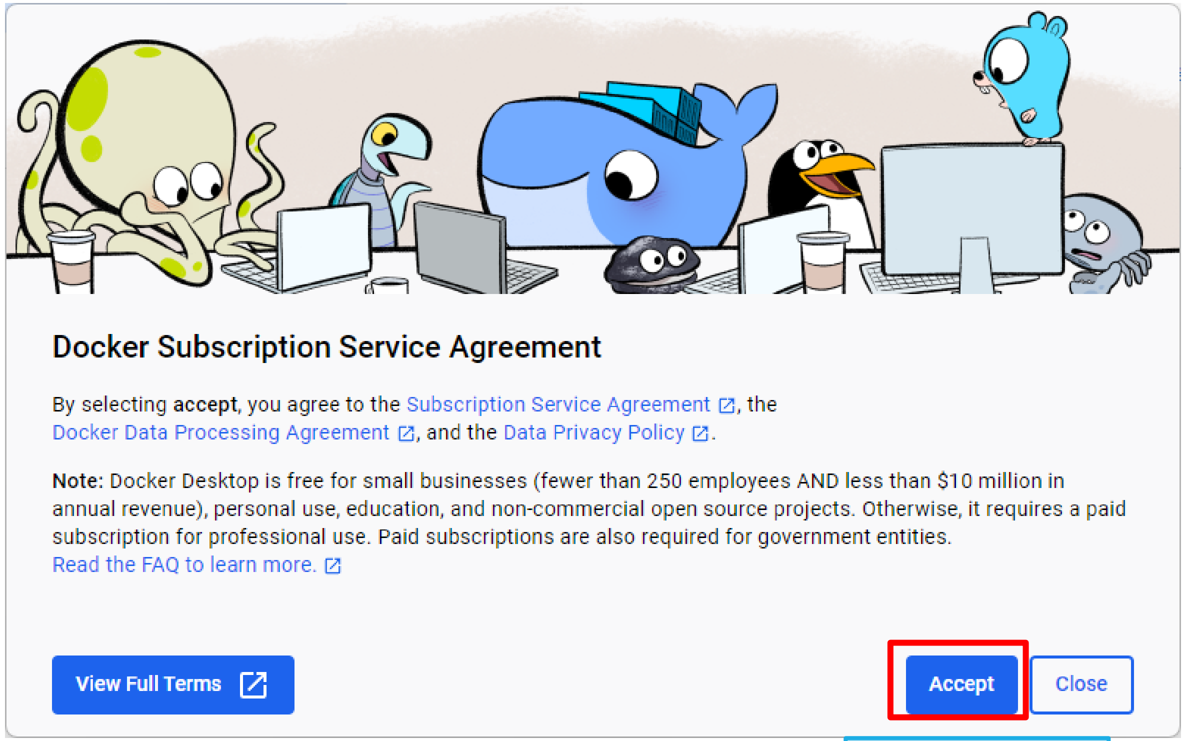
By selecting **accept**, y

Docker Data Processi

Note: Docker Desktop $10 million in annual revenue), personal use, education, and non-commercial open sourc ires a paid subscription for professional use. Paid subscriptions are also required for

Read the FAQ to learn more.

2-4. 確定



Docker Subscription Service Agreement

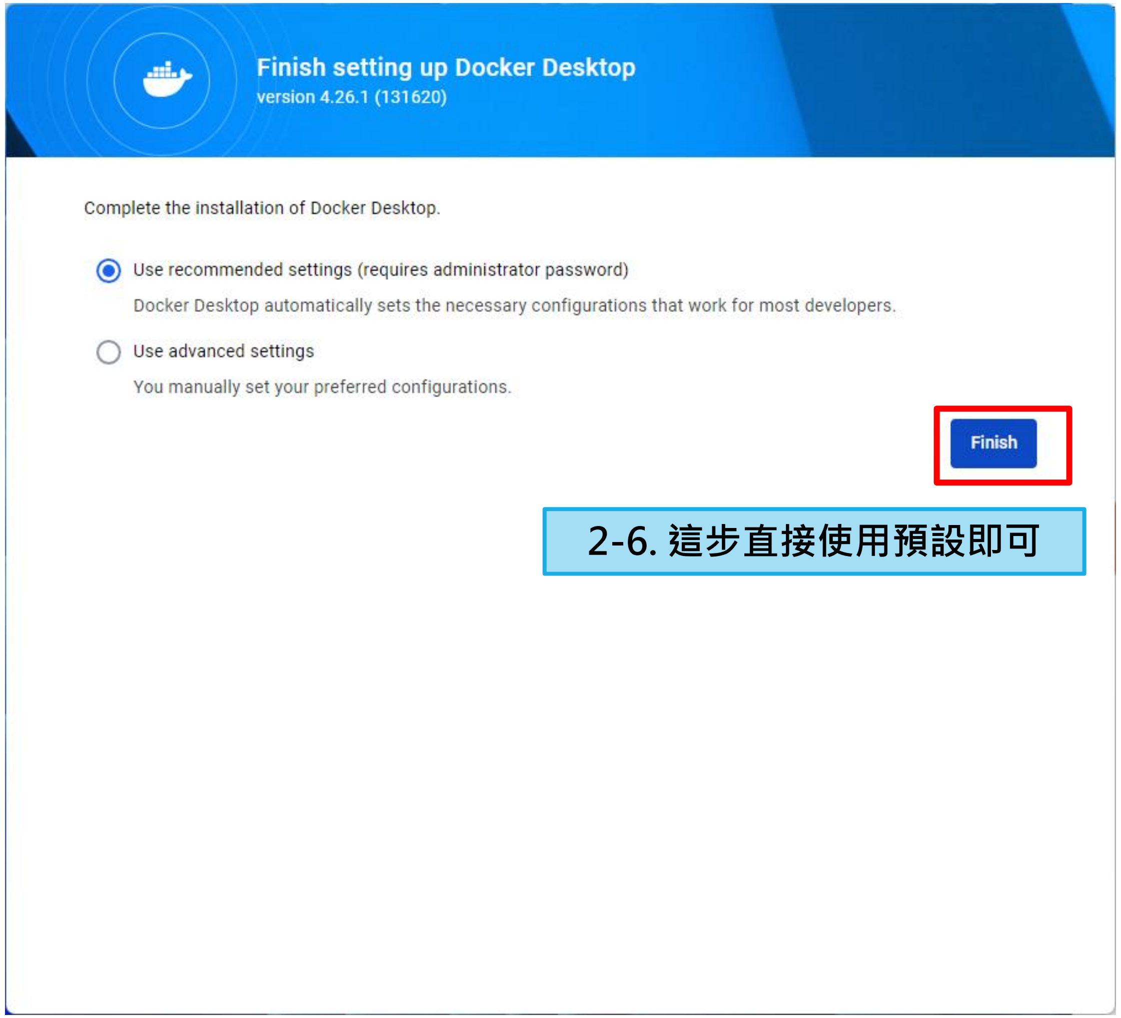
By selecting **accept**, you agree to the Subscription Service Agreement, the Docker Data Processing Agreement, and the Data Privacy Policy.

Note: Docker Desktop is free for small businesses (fewer than 250 employees AND less than $10 million in annual revenue), personal use, education, and non-commercial open source projects. Otherwise, it requires a paid subscription for professional use. Paid subscriptions are also required for government entities.

Read the FAQ to learn more.

2-5. Accept

2. 安裝docker



Finish setting up Docker Desktop

version 4.26.1 (131620)

Complete the installation of Docker Desktop.

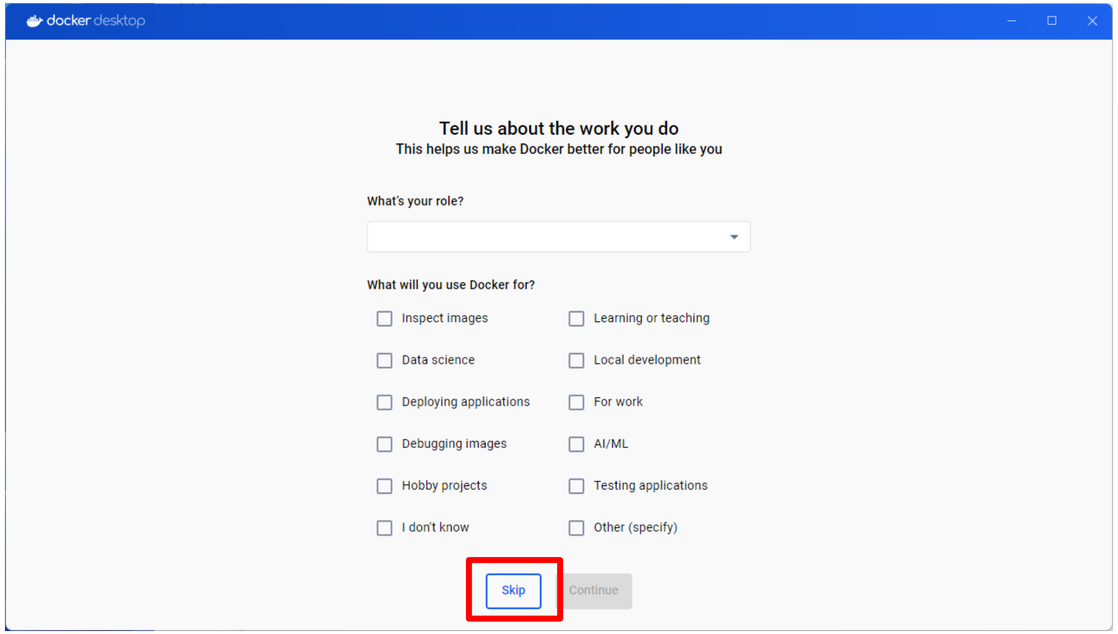
● Use recommended settings (requires administrator password)

Docker Desktop automatically sets the necessary configurations that work for most developers.

○ Use advanced settings

You manually set your preferred configurations.

2-6. 這步直接使用預設即可



Tell us about the work you do

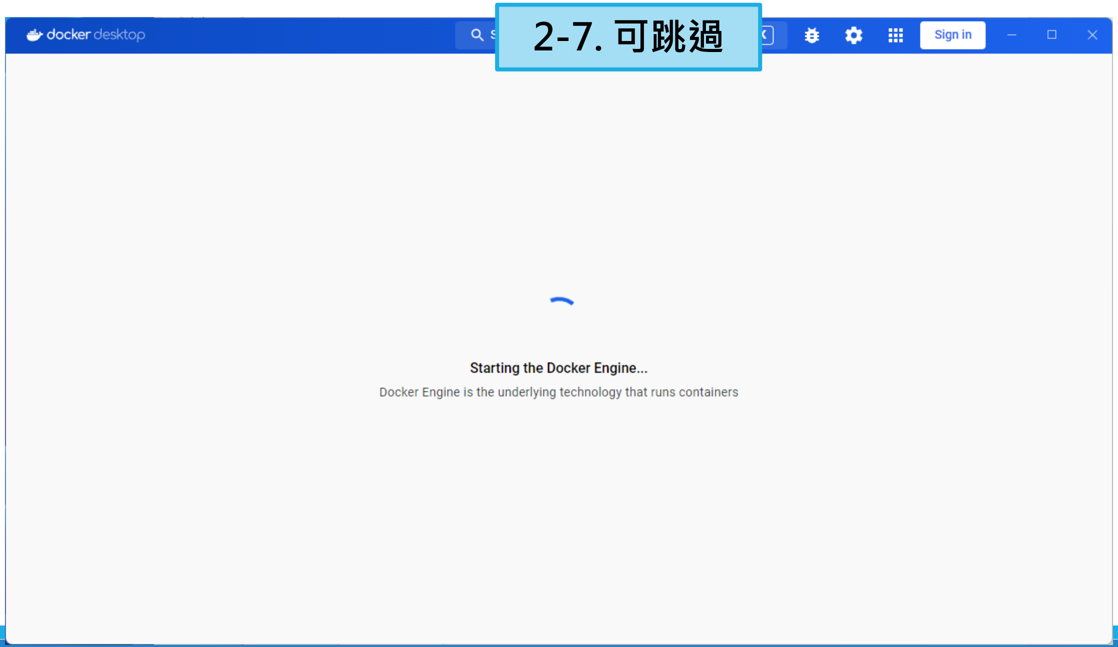
This helps us make Docker better for people like you

What's your role?

What will you use Docker for?

□ Inspect images
□ Data science
□ Deploying applications
□ Debugging images
□ Hobby projects
□ I don't know
□ Learning or teaching
□ Local development
□ For work
□ AI/ML
□ Testing applications
□ Other (specify)

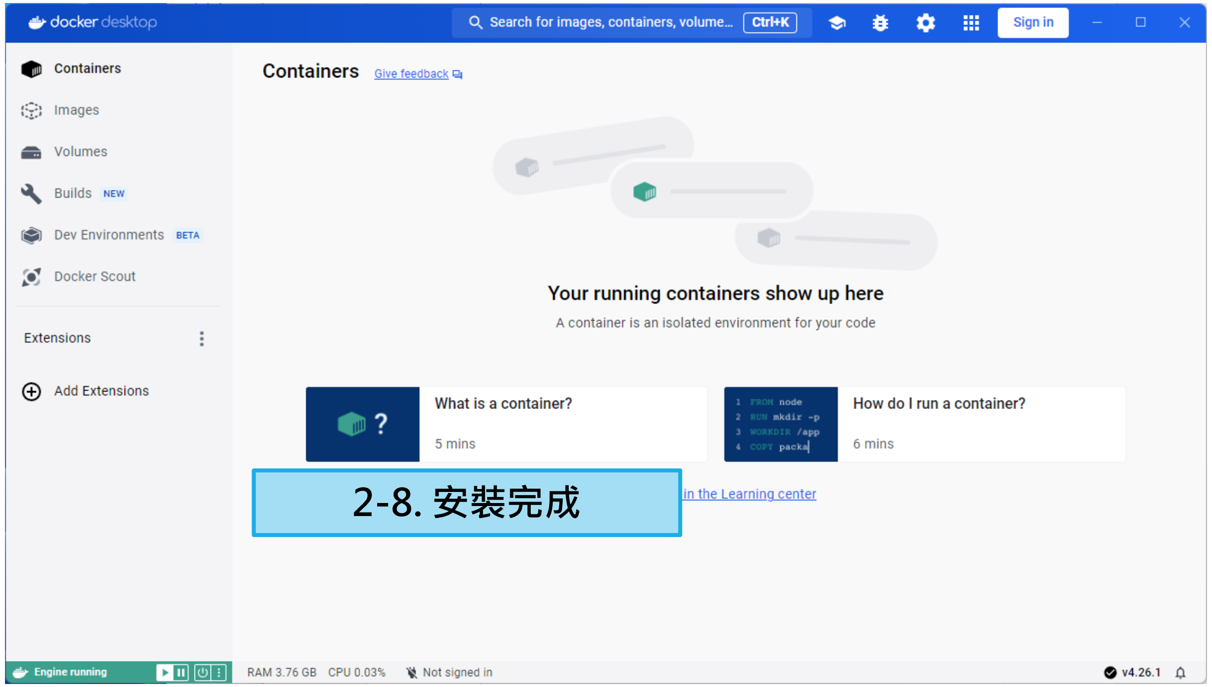
2-7. 可跳過



Starting the Docker Engine...

Docker Engine is the underlying technology that runs containers

2. 安裝docker



Containers

Your running containers show up here

A container is an isolated environment for your code

What is a container?
5 mins

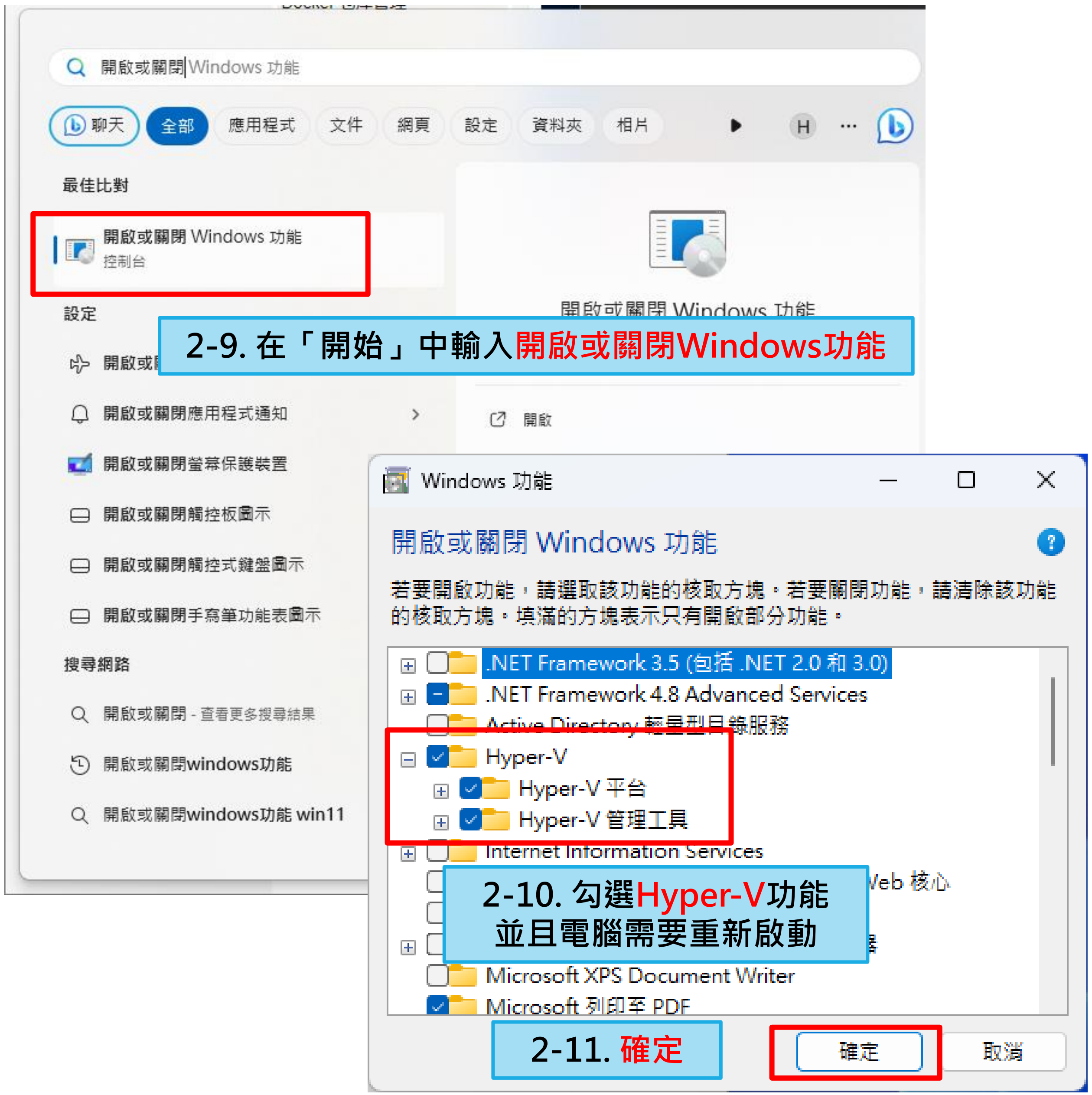
```
FROM node
RUN mkdir -p
WORKDIR /app
COPY packa
```

How do I run a container?
6 mins

in the Learning center

RAM 3.76 GB CPU 0.03% Not signed in v4.26.1

2-8. 安裝完成



開啟或關閉Windows 功能

最佳比對

開啟或關閉 Windows 功能
控制台

設定

開啟或關

開啟或關閉應用程式通知

開啟或關閉螢幕保護裝置

開啟或關閉觸控板圖示

開啟或關閉觸控式鍵盤圖示

開啟或關閉手寫筆功能表圖示

搜尋網路

開啟或關閉 - 查看更多搜尋結果

開啟或關閉windows功能

開啟或關閉windows功能 win11

開啟或關閉 Windows 功能

2-9. 在「開始」中輸入開啟或關閉Windows功能

Windows 功能

開啟或關閉 Windows 功能

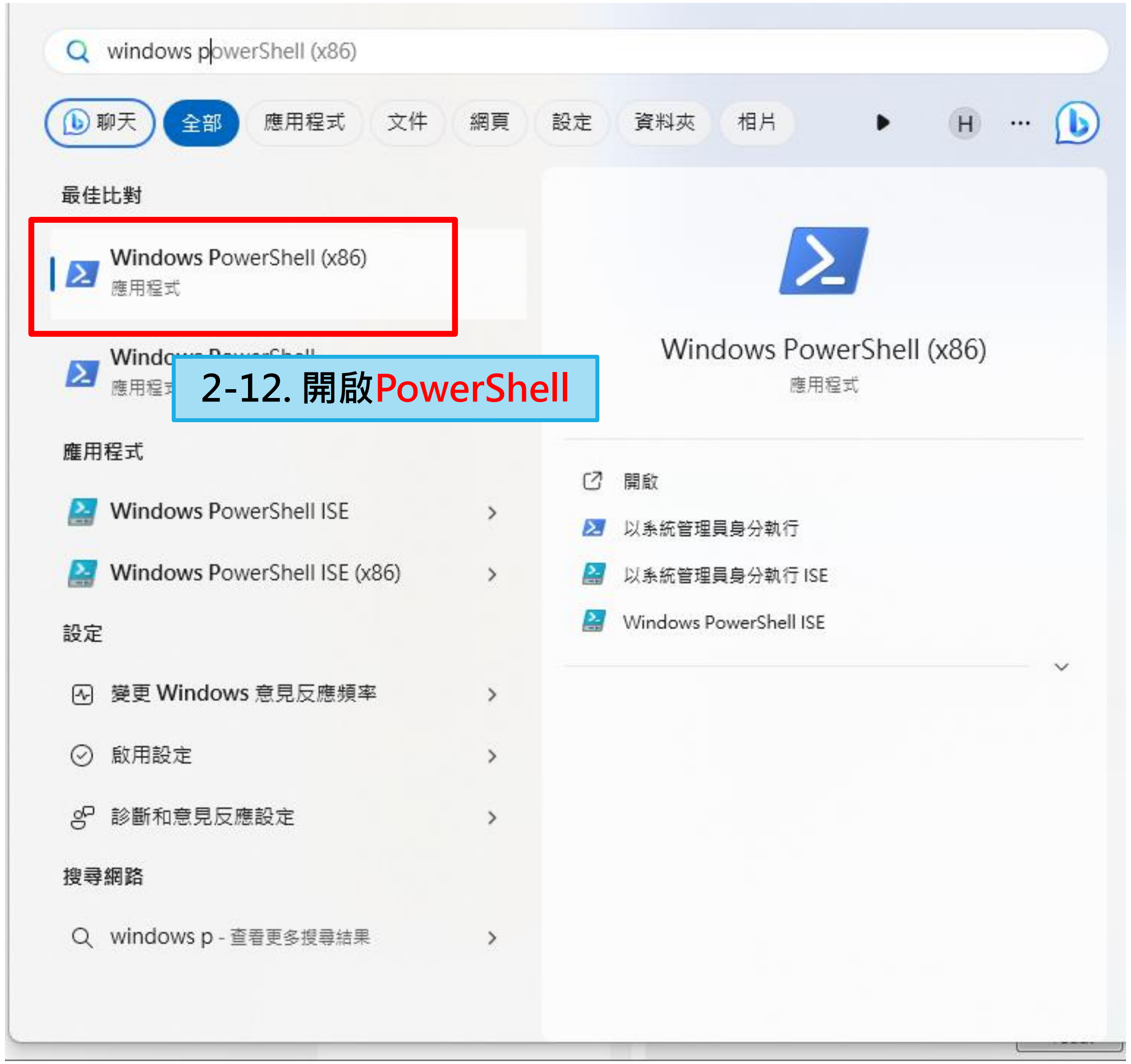
若要開啟功能，請選取該功能的核取方塊。若要關閉功能，請清除該功能的核取方塊。填滿的方塊表示只有開啟部分功能。

- ☐ .NET Framework 3.5 (包括 .NET 2.0 和 3.0)
- ■ .NET Framework 4.8 Advanced Services
- ☐ Active Directory 輕量型目錄服務
- ☑ Hyper-V
  - ☑ Hyper-V 平台
  - ☑ Hyper-V 管理工具
- ☐ Internet Information Services
- ☐ Veb 核心
- ☐ Microsoft XPS Document Writer
- ☑ Microsoft 列印至 PDF

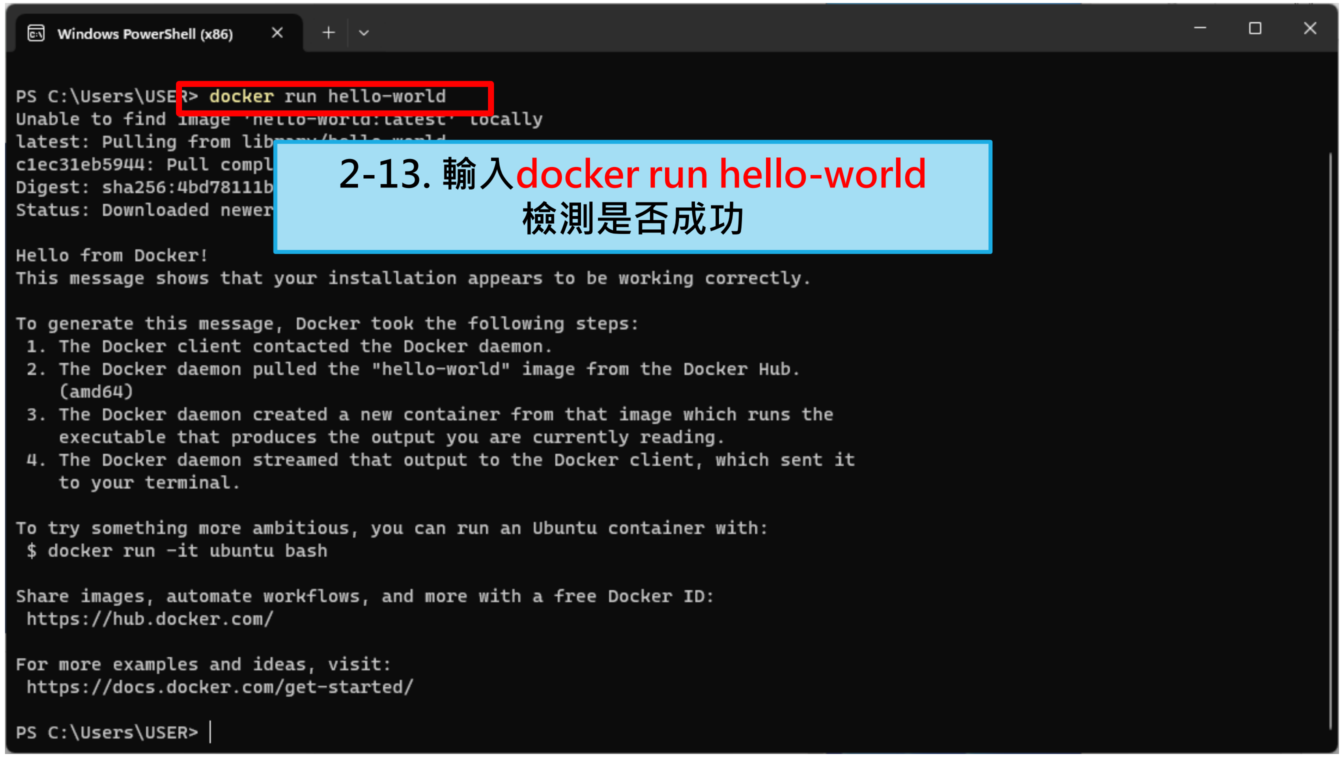
2-10. 勾選Hyper-V功能
並且電腦需要重新啟動

2-11. 確定

2. 安裝docker



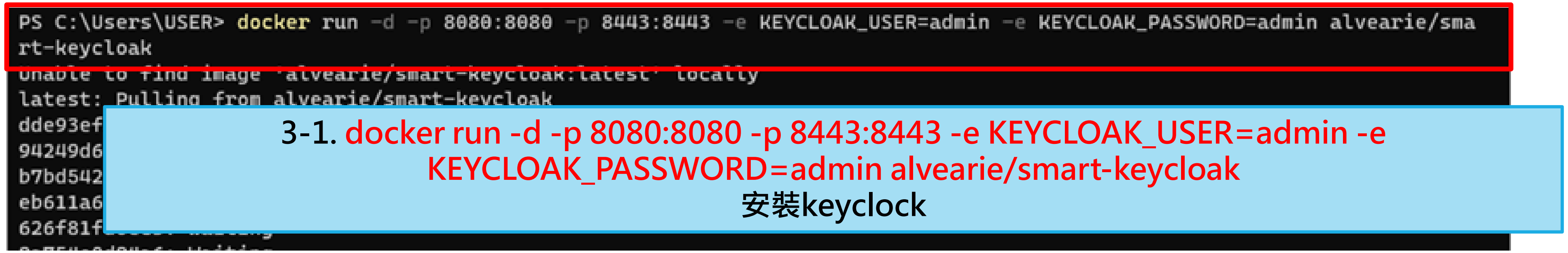
2-12. 開啟PowerShell



```
PS C:\Users\USER> docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from lib
c1ec31eb5944: Pull compl
Digest: sha256:4bd78111b
Status: Downloaded newer

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

PS C:\Users\USER>
```

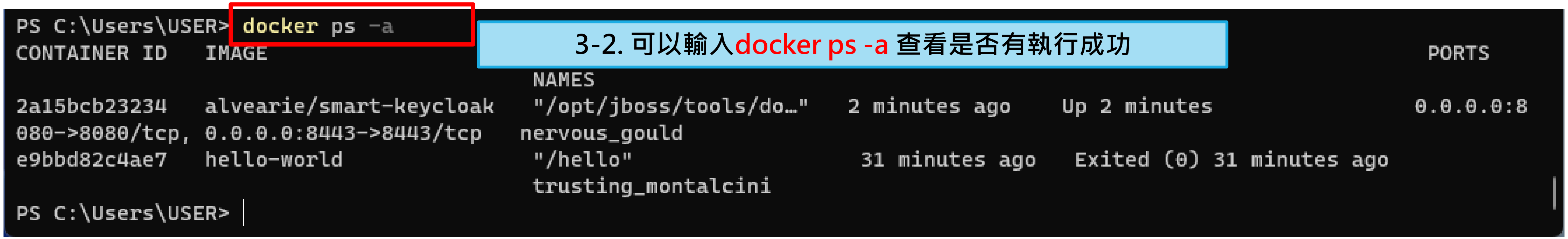
2-13. 輸入docker run hello-world
檢測是否成功

3. 安裝keycloak



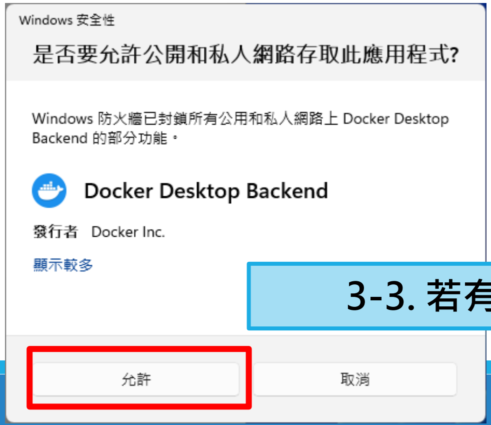
```
PS C:\Users\USER> docker run -d -p 8080:8080 -p 8443:8443 -e KEYCLOAK_USER=admin -e KEYCLOAK_PASSWORD=admin alvearie/sma
rt-keycloak
Unable to find image 'alvearie/smart-keycloak:latest' locally
latest: Pulling from alvearie/smart-keycloak
dde93ef
94249d6
b7bd542
eb611a6
626f81f
```

3-1. docker run -d -p 8080:8080 -p 8443:8443 -e KEYCLOAK_USER=admin -e KEYCLOAK_PASSWORD=admin alvearie/smart-keycloak
安裝keyclock



```
PS C:\Users\USER> docker ps -a
CONTAINER ID   IMAGE                                                                                  PORTS
                                          NAMES
2a15bcb23234   alvearie/smart-keycloak    "/opt/jboss/tools/do…"   2 minutes ago    Up 2 minutes              0.0.0.0:8
080->8080/tcp, 0.0.0.0:8443->8443/tcp    nervous_gould
e9bbd82c4ae7   hello-world                "/hello"                  31 minutes ago   Exited (0) 31 minutes ago
                                          trusting_montalcini
PS C:\Users\USER>
```

3-2. 可以輸入docker ps -a 查看是否有執行成功



Windows 安全性

是否要允許公開和私人網路存取此應用程式?

Windows 防火牆已封鎖所有公用和私人網路上 Docker Desktop Backend 的部分功能。

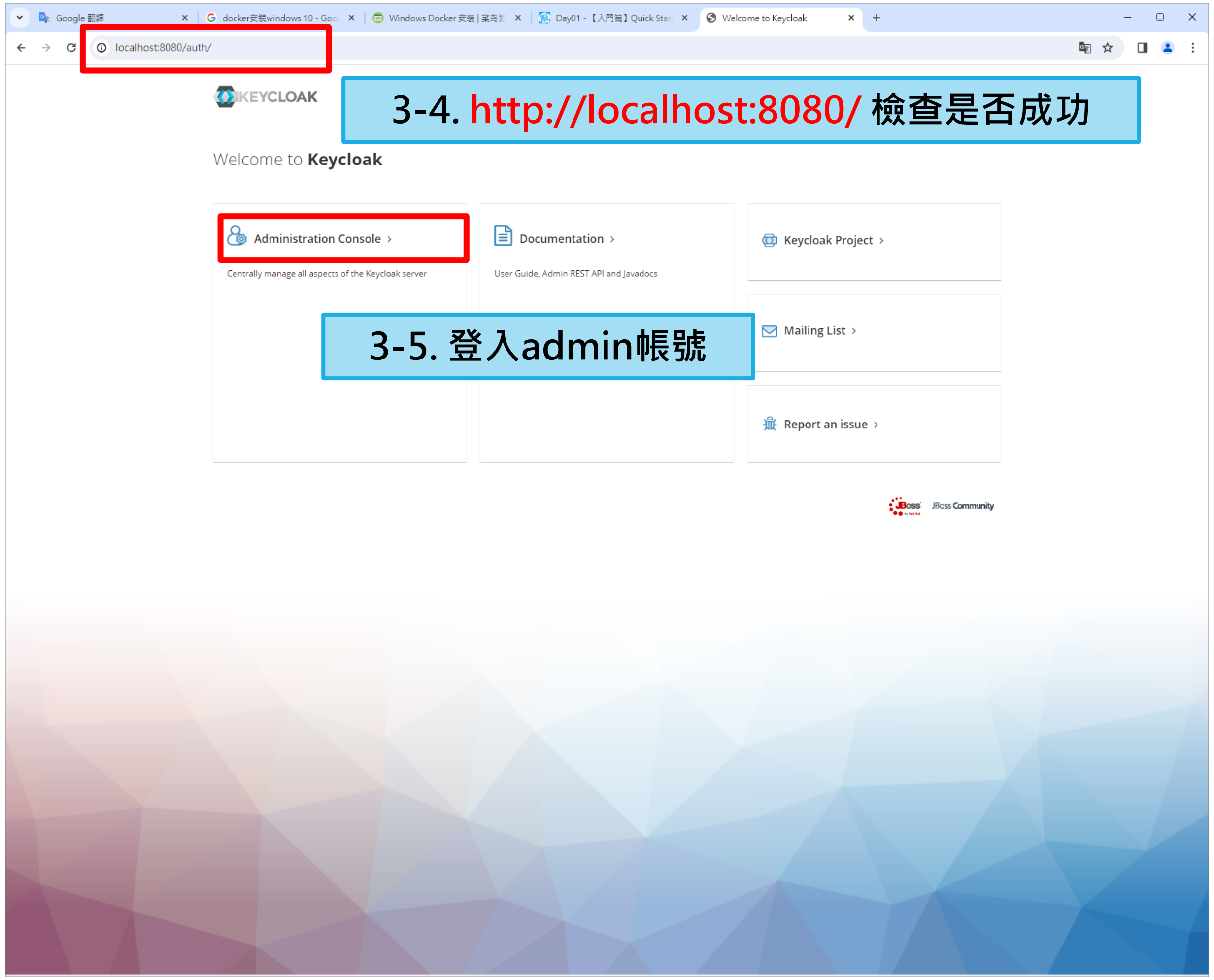
Docker Desktop Backend

發行者 Docker Inc.

顯示較多

3-3. 若有跳出，允許即可

3. 安裝keycloak


Google 翻譯
docker安裝windows 10 - Goo
Windows Docker 安裝 | 菜鸟
Day01 - 【入門篇】Quick Star
Welcome to Keycloak

localhost:8080/auth/

KEYCLOAK

Welcome to **Keycloak**

Administration Console ›

Centrally manage all aspects of the Keycloak server

Documentation ›

User Guide, Admin REST API and Javadocs

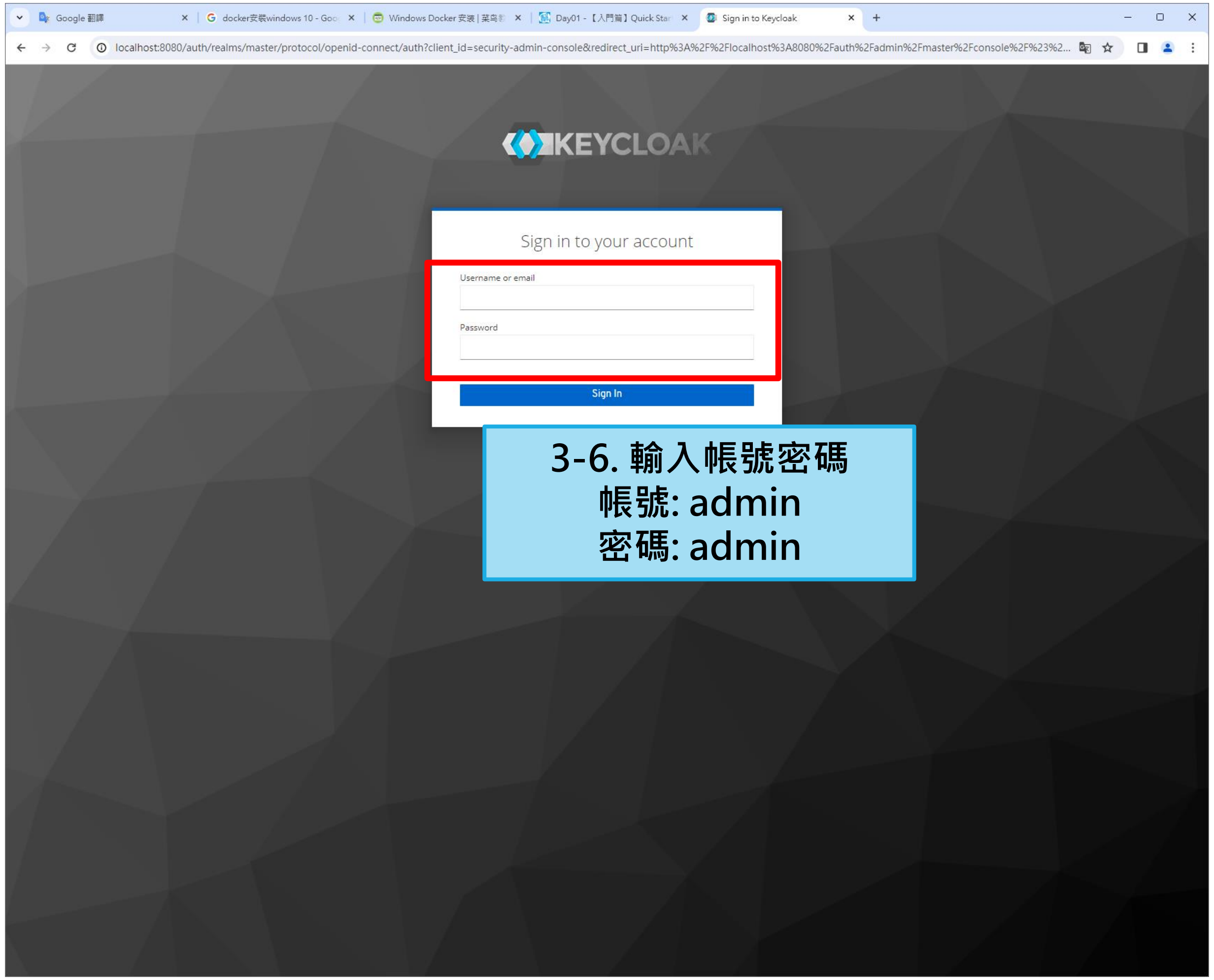
Keycloak Project ›

Mailing List ›

Report an issue ›

3-4. http://localhost:8080/ 檢查是否成功

3-5. 登入admin帳號


Google 翻譯
docker安裝windows 10 - Goo
Windows Docker 安裝 | 菜鸟
Day01 - 【入門篇】Quick Star
Sign in to Keycloak

localhost:8080/auth/realms/master/protocol/openid-connect/auth?client_id=security-admin-console&redirect_uri=http%3A%2F%2Flocalhost%3A8080%2Fauth%2Fadmin%2Fmaster%2Fconsole%2F%23%2...

KEYCLOAK

Sign in to your account

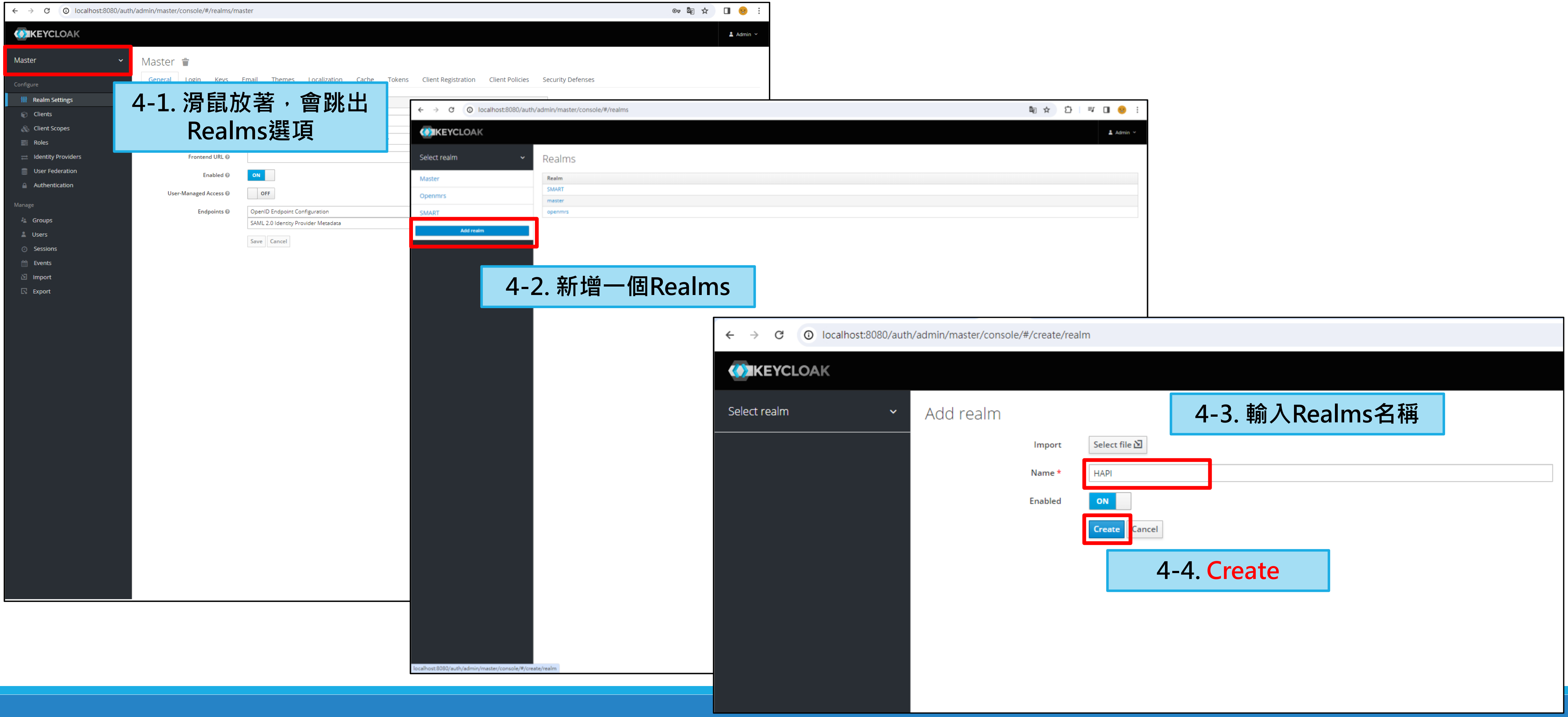
Username or email

Password

Sign In

3-6. 輸入帳號密碼
帳號: admin
密碼: admin

4. 配置keycloak



Frontend URL

Enabled: ON

User-Managed Access: OFF

Endpoints: OpenID Endpoint Configuration, SAML 2.0 Identity Provider Metadata

4-1. 滑鼠放著，會跳出Realms選項

localhost:8080/auth/admin/master/console/#/realms
KEYCLOAK
Admin
Select realm
Master
Openmrs
SMART
Add realm
Realms
localhost:8080/auth/admin/master/console/#/create/realm

Realm
SMART
master
openmrs

4-2. 新增一個Realms

Import: Select file

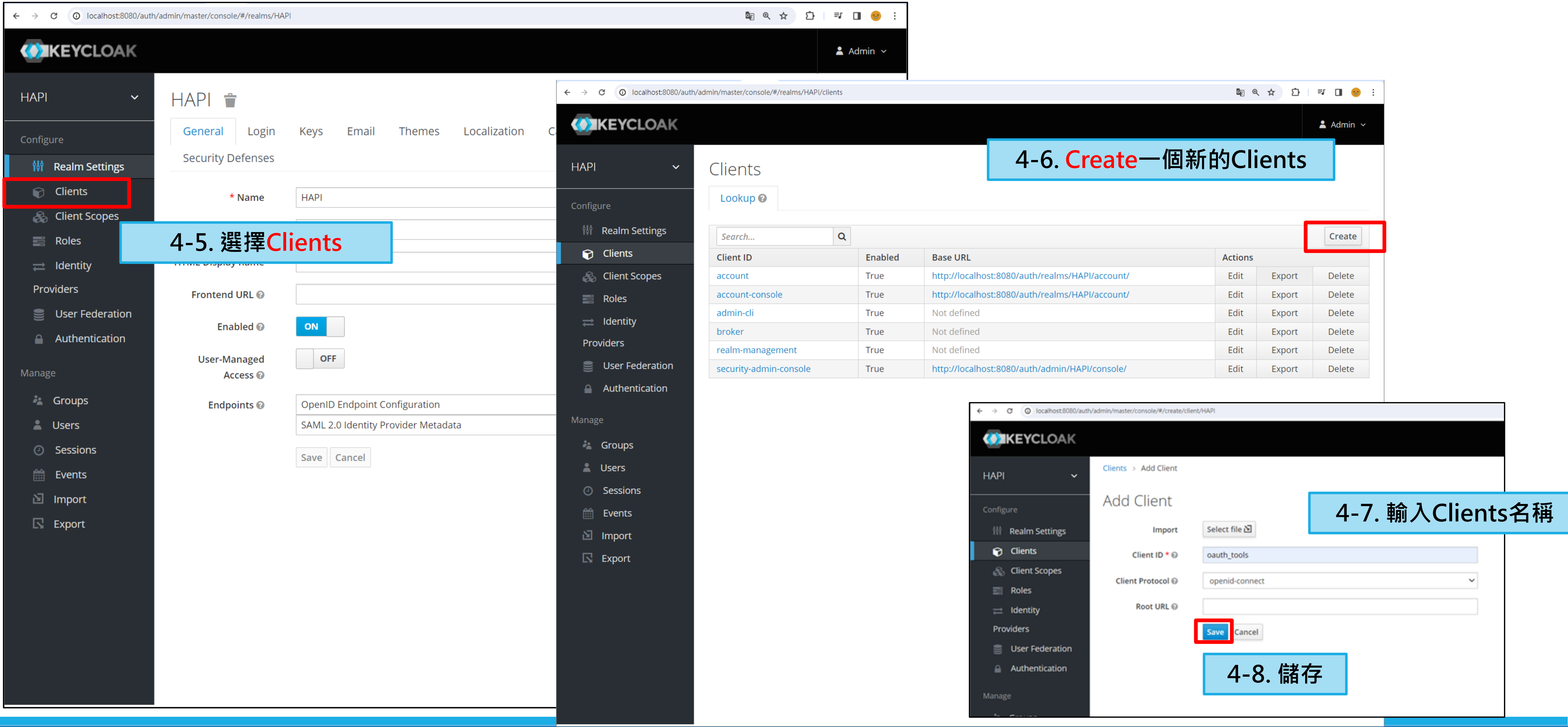
Name *: HAPI

Enabled: ON

4-3. 輸入Realms名稱

4-4. Create

4. 配置keycloak



4-5. 選擇Clients

4-6. Create一個新的Clients

4-7. 輸入Clients名稱

4-8. 儲存

localhost:8080/auth/admin/master/console/#/realms/HAPI

HAPI

* Name: HAPI

Frontend URL

Enabled: ON

User-Managed Access: OFF

Endpoints: OpenID Endpoint Configuration / SAML 2.0 Identity Provider Metadata

localhost:8080/auth/admin/master/console/#/realms/HAPI/clients

Clients

Client ID	Enabled	Base URL	Actions		
account	True	http://localhost:8080/auth/realms/HAPI/account/	Edit	Export	Delete
account-console	True	http://localhost:8080/auth/realms/HAPI/account/	Edit	Export	Delete
admin-cli	True	Not defined	Edit	Export	Delete
broker	True	Not defined	Edit	Export	Delete
realm-management	True	Not defined	Edit	Export	Delete
security-admin-console	True	http://localhost:8080/auth/admin/HAPI/console/	Edit	Export	Delete

localhost:8080/auth/admin/master/console/#/create/client/HAPI

Clients › Add Client

Add Client

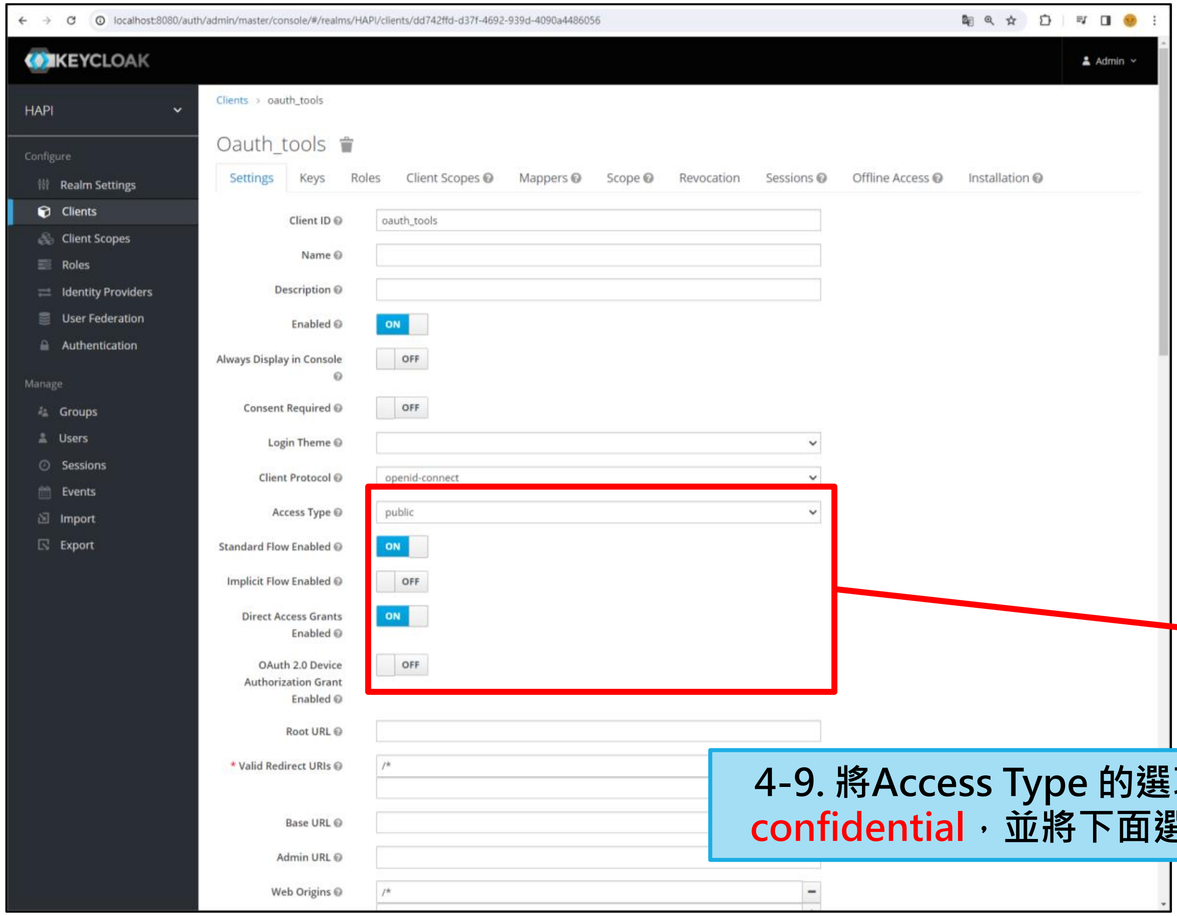
Import: Select file

Client ID *: oauth_tools

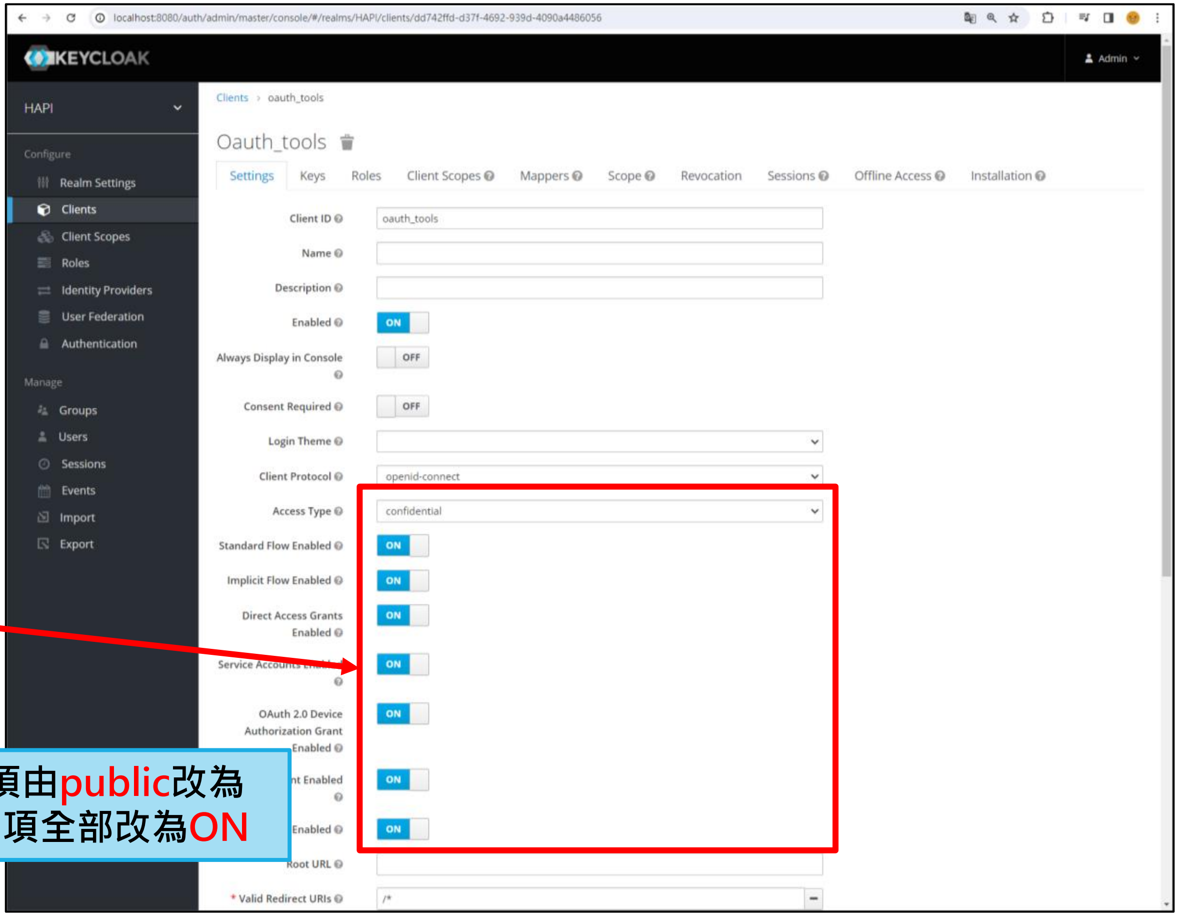
Client Protocol: openid-connect

Root URL

4. 配置keycloak



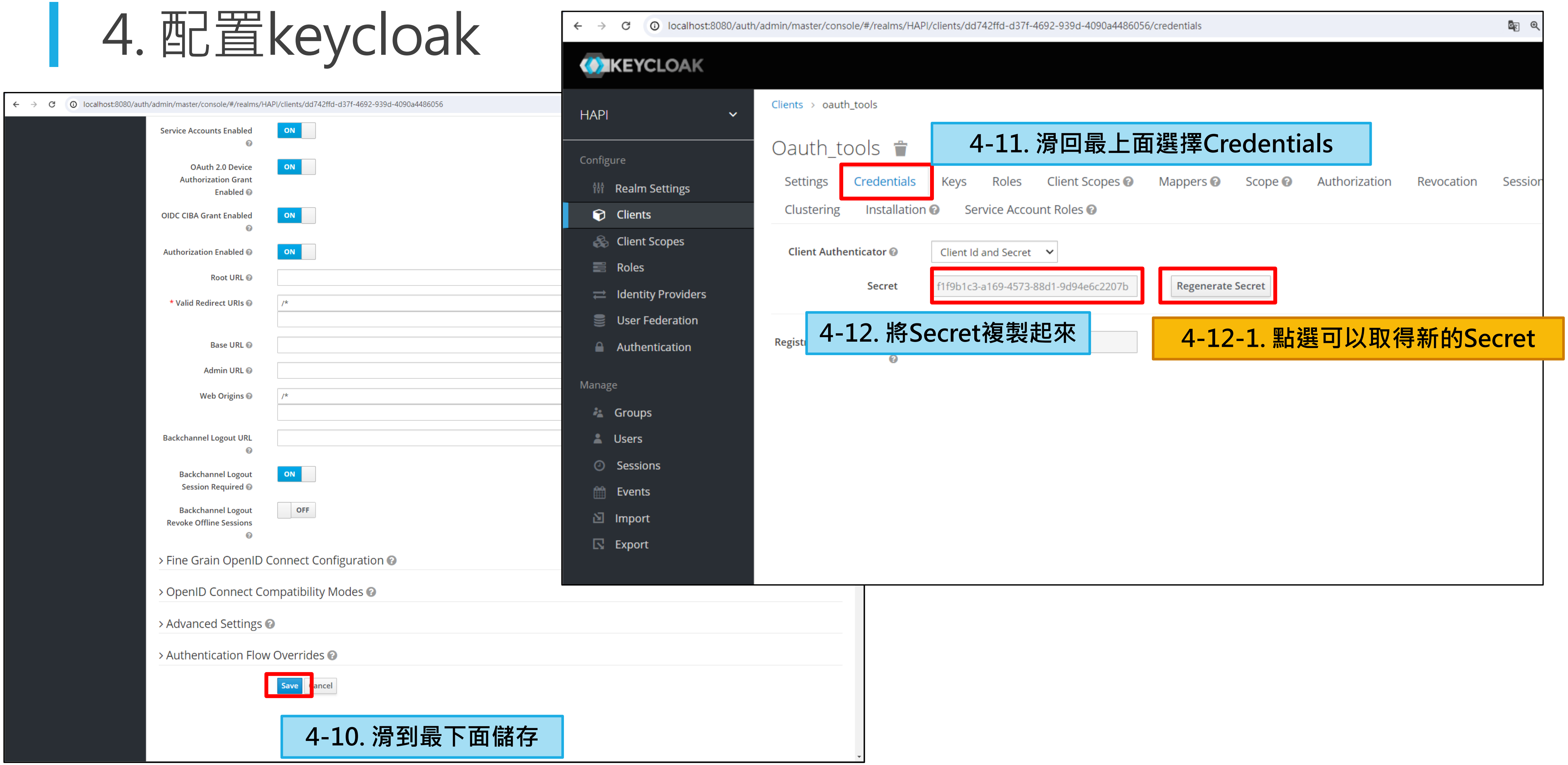
Client ID	oauth_tools
Name	
Description	
Enabled	ON
Always Display in Console	OFF
Consent Required	OFF
Login Theme	
Client Protocol	openid-connect
Access Type	public
Standard Flow Enabled	ON
Implicit Flow Enabled	OFF
Direct Access Grants Enabled	ON
OAuth 2.0 Device Authorization Grant Enabled	OFF
Root URL	
* Valid Redirect URIs	/*
Base URL	
Admin URL	
Web Origins	/*



Client ID	oauth_tools
Name	
Description	
Enabled	ON
Always Display in Console	OFF
Consent Required	OFF
Login Theme	
Client Protocol	openid-connect
Access Type	confidential
Standard Flow Enabled	ON
Implicit Flow Enabled	ON
Direct Access Grants Enabled	ON
Service Accounts Enabled	ON
OAuth 2.0 Device Authorization Grant Enabled	ON
nt Enabled	ON
Enabled	ON
Root URL	
* Valid Redirect URIs	/*

4-9. 將Access Type 的選項由public改為confidential，並將下面選項全部改為ON

4. 配置keycloak



localhost:8080/auth/admin/master/console/#/realms/HAPI/clients/dd742ffd-d37f-4692-939d-4090a4486056

Service Accounts Enabled: ON

OAuth 2.0 Device Authorization Grant Enabled: ON

OIDC CIBA Grant Enabled: ON

Authorization Enabled: ON

Root URL:

* Valid Redirect URIs: /*

Base URL:

Admin URL:

Web Origins: /*

Backchannel Logout URL:

Backchannel Logout Session Required: ON

Backchannel Logout Revoke Offline Sessions: OFF

› Fine Grain OpenID Connect Configuration
› OpenID Connect Compatibility Modes
› Advanced Settings
› Authentication Flow Overrides
Save
Cancel

4-10. 滑到最下面儲存

localhost:8080/auth/admin/master/console/#/realms/HAPI/clients/dd742ffd-d37f-4692-939d-4090a4486056/credentials

Client Authenticator: Client Id and Secret

Secret: f1f9b1c3-a169-4573-88d1-9d94e6c2207b

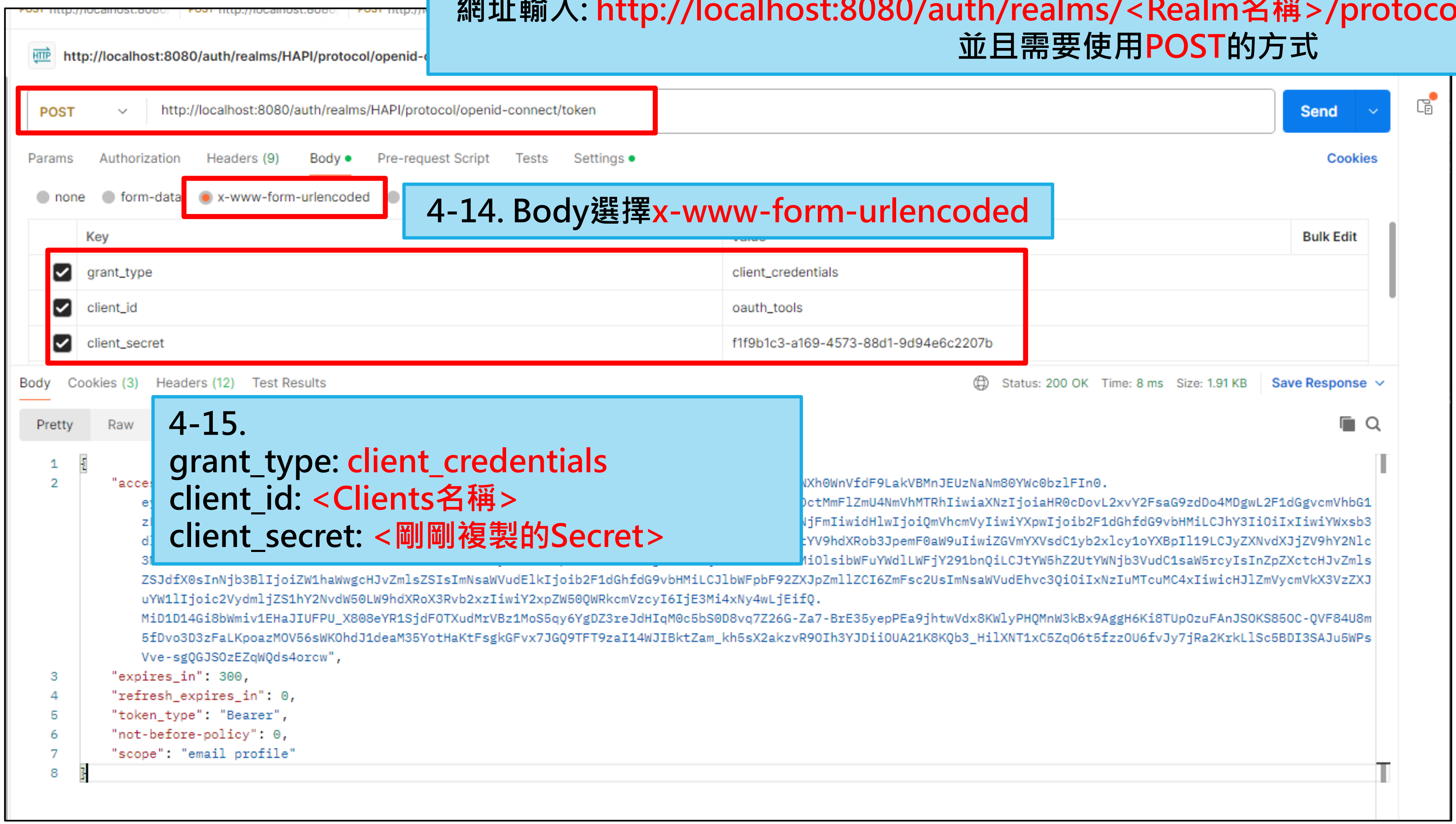
4-11. 滑回最上面選擇Credentials

4-12. 將Secret複製起來

4-12-1. 點選可以取得新的Secret

4. 配置keycloak

4-13. 可以用postman嘗試是否能夠取得access_token
網址輸入: http://localhost:8080/auth/realms/<Realm名稱>/protocol/openid-connect/token
並且需要使用POST的方式



http://localhost:8080/auth/realms/HAPI/protocol/openid-

POST http://localhost:8080/auth/realms/HAPI/protocol/openid-connect/token

4-14. Body選擇x-www-form-urlencoded

	Key	Value
☑	grant_type	client_credentials
☑	client_id	oauth_tools
☑	client_secret	f1f9b1c3-a169-4573-88d1-9d94e6c2207b

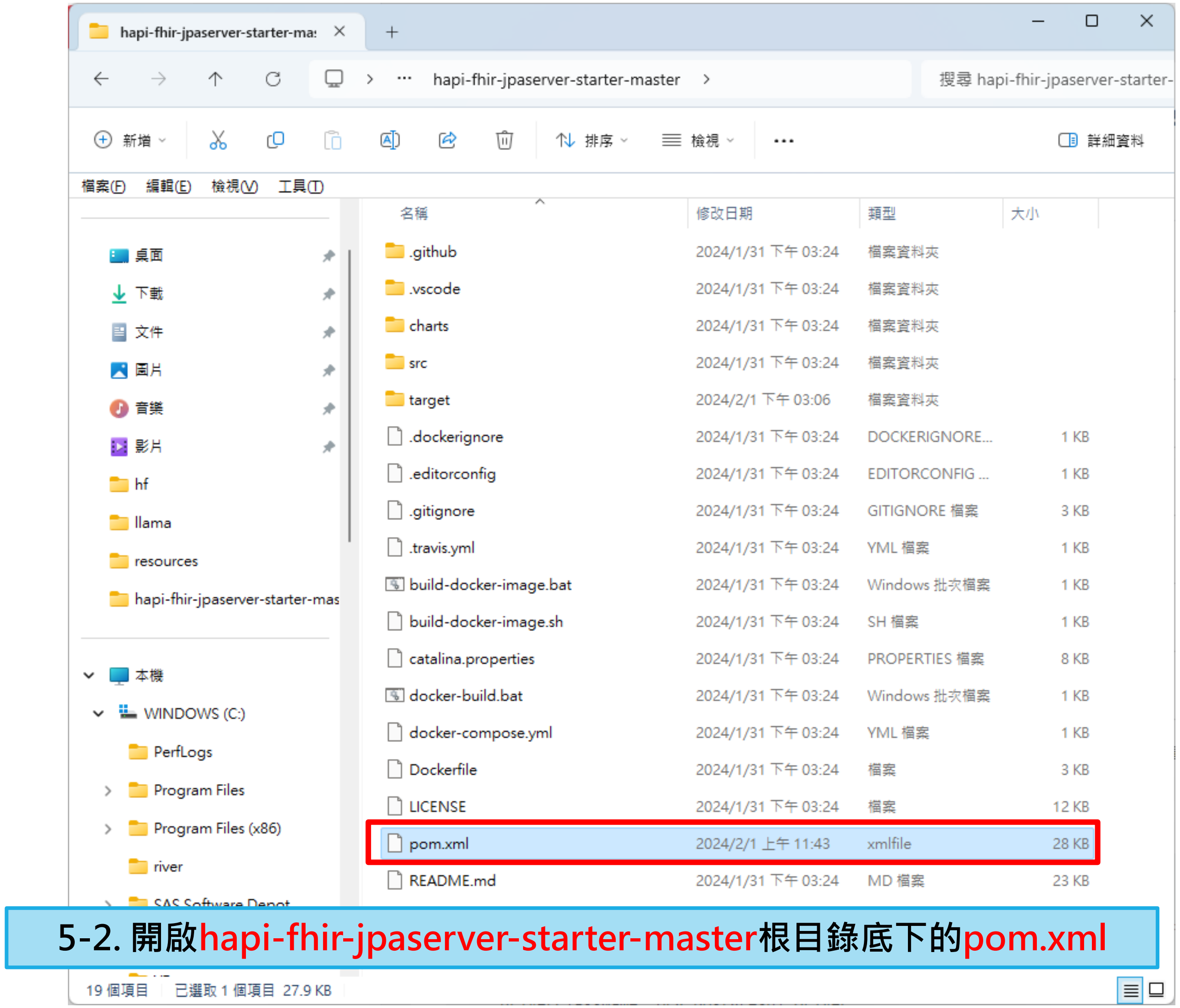
Status: 200 OK Time: 8 ms Size: 1.91 KB

4-15.
grant_type: client_credentials
client_id: <Clients名稱>
client_secret: <剛剛複製的Secret>

```
{
    "acce NXh0WnVfdF9LakVBMnJEUzNaNm80YWc0bzlFIn0.
        e 0ctMmFlZmU4NmVhMTRhIiwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdDo4MDgwL2F1dGgvcmVhbG1
        z NjFmIiwidHlwIjoiQmVhcmVyIiwiYXpwIjoib2F1dGhfdG9vbHMiLCJhY3IiOiIxIiwiYWxsb3
        d tYV9hdXRob3JpemF0aW9uIiwiZGVmYXVsdC1yb2xlcy1oYXBpIl19LCJyZXNvdXJjZV9hY2Nlc
        3 MiOlsibWFuYWdlLWFjY291bnQiLCJtYW5hZ2UtYWNjb3VudC1saW5rcyIsInZpZXctcHJvZmls
        ZSJdfX0sInNjb3BlIjoiZW1haWwgcHJvZmlsZSIsImNsaWVudElkIjoib2F1dGhfdG9vbHMiLCJlbWFpbF92ZXJpZmllZCI6ZmFsc2UsImNsaWVudEhvc3QiOiIxNzIuMTcuMC4xIiwicHJlZmVycmVkX3VzZXJ
        uYW1lIjoic2VydmljZS1hY2NvdW50LW9hdXRoX3Rvb2xzIiwiY2xpZW50QWRkcmVzcyI6IjE3Mi4xNy4wLjEifQ.
        MiD1D14Gi8bWmiv1EHaJIUFPU_X808eYR1SjdFOTXudMrVBz1MoS5qy6YgDZ3reJdHIqM0c5bS0D8vq7Z26G-Za7-BrE35yepPEa9jhtwVdx8KWlyPHQMnW3kBx9AggH6Ki8TUpOzuFAnJSOKS850C-QVF84U8m
        5fDvo3D3zFaLKpoazMOV56sWKOhdJ1deaM35YotHaKtFsgkGFvx7JGQ9TFT9zaI14WJIBktZam_kh5sX2akzvR9OIh3YJDiiOUA21K8KQb3_HilXNT1xC5ZqO6t5fzzOU6fvJy7jRa2KrkLlSc5BDI3SAJu5WPs
        Vve-sgQGJSOzEZqWQds4orcw",
    "expires_in": 300,
    "refresh_expires_in": 0,
    "token_type": "Bearer",
    "not-before-policy": 0,
    "scope": "email profile"
}
```

5. 修改HAPI FHIR的設定

5-1. 首先將HAPI FHIR 架設起來(這裡以Windows版本為例)
架設步驟可參考網頁，並先不執行HAPI FHIR
https://silcoet.ntunhs.edu.tw/FHIRSampleCode/Server/Hapi

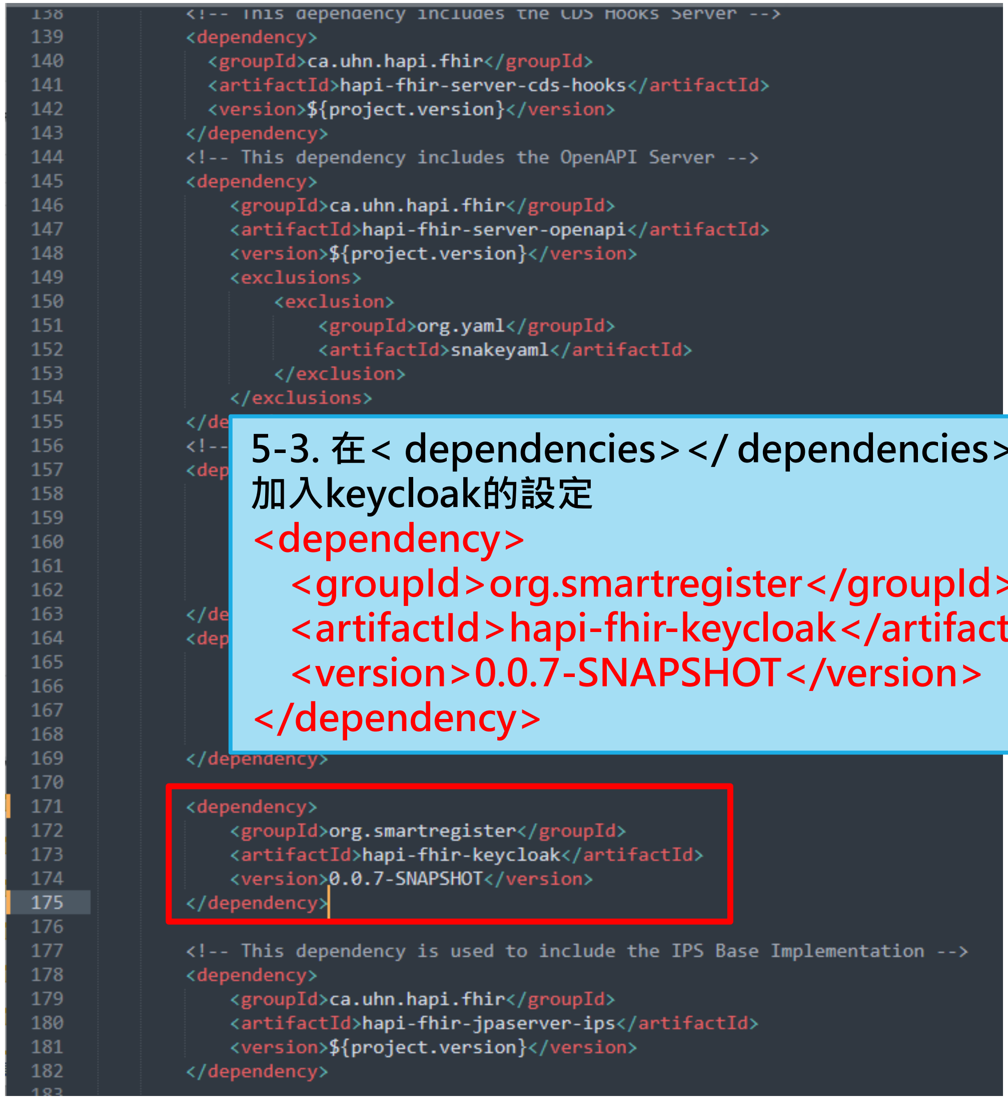


hapi-fhir-jpaserver-starter-ma:
hapi-fhir-jpaserver-starter-master
搜尋 hapi-fhir-jpaserver-starter-
新增
排序
檢視
詳細資料
檔案(F) 編輯(E) 檢視(V) 工具(T)
桌面
下載
文件
圖片
音樂
影片
hf
llama
resources
hapi-fhir-jpaserver-starter-mas
本機
WINDOWS (C:)
PerfLogs
Program Files
Program Files (x86)
river
SAS Software Depot

名稱	修改日期	類型	大小
.github	2024/1/31 下午 03:24	檔案資料夾	
.vscode	2024/1/31 下午 03:24	檔案資料夾	
charts	2024/1/31 下午 03:24	檔案資料夾	
src	2024/1/31 下午 03:24	檔案資料夾	
target	2024/2/1 下午 03:06	檔案資料夾	
.dockerignore	2024/1/31 下午 03:24	DOCKERIGNORE...	1 KB
.editorconfig	2024/1/31 下午 03:24	EDITORCONFIG ...	1 KB
.gitignore	2024/1/31 下午 03:24	GITIGNORE 檔案	3 KB
.travis.yml	2024/1/31 下午 03:24	YML 檔案	1 KB
build-docker-image.bat	2024/1/31 下午 03:24	Windows 批次檔案	1 KB
build-docker-image.sh	2024/1/31 下午 03:24	SH 檔案	1 KB
catalina.properties	2024/1/31 下午 03:24	PROPERTIES 檔案	8 KB
docker-build.bat	2024/1/31 下午 03:24	Windows 批次檔案	1 KB
docker-compose.yml	2024/1/31 下午 03:24	YML 檔案	1 KB
Dockerfile	2024/1/31 下午 03:24	檔案	3 KB
LICENSE	2024/1/31 下午 03:24	檔案	12 KB
pom.xml	2024/2/1 上午 11:43	xmlfile	28 KB
README.md	2024/1/31 下午 03:24	MD 檔案	23 KB

19 個項目　已選取 1 個項目 27.9 KB

5-2. 開啟hapi-fhir-jpaserver-starter-master根目錄底下的pom.xml

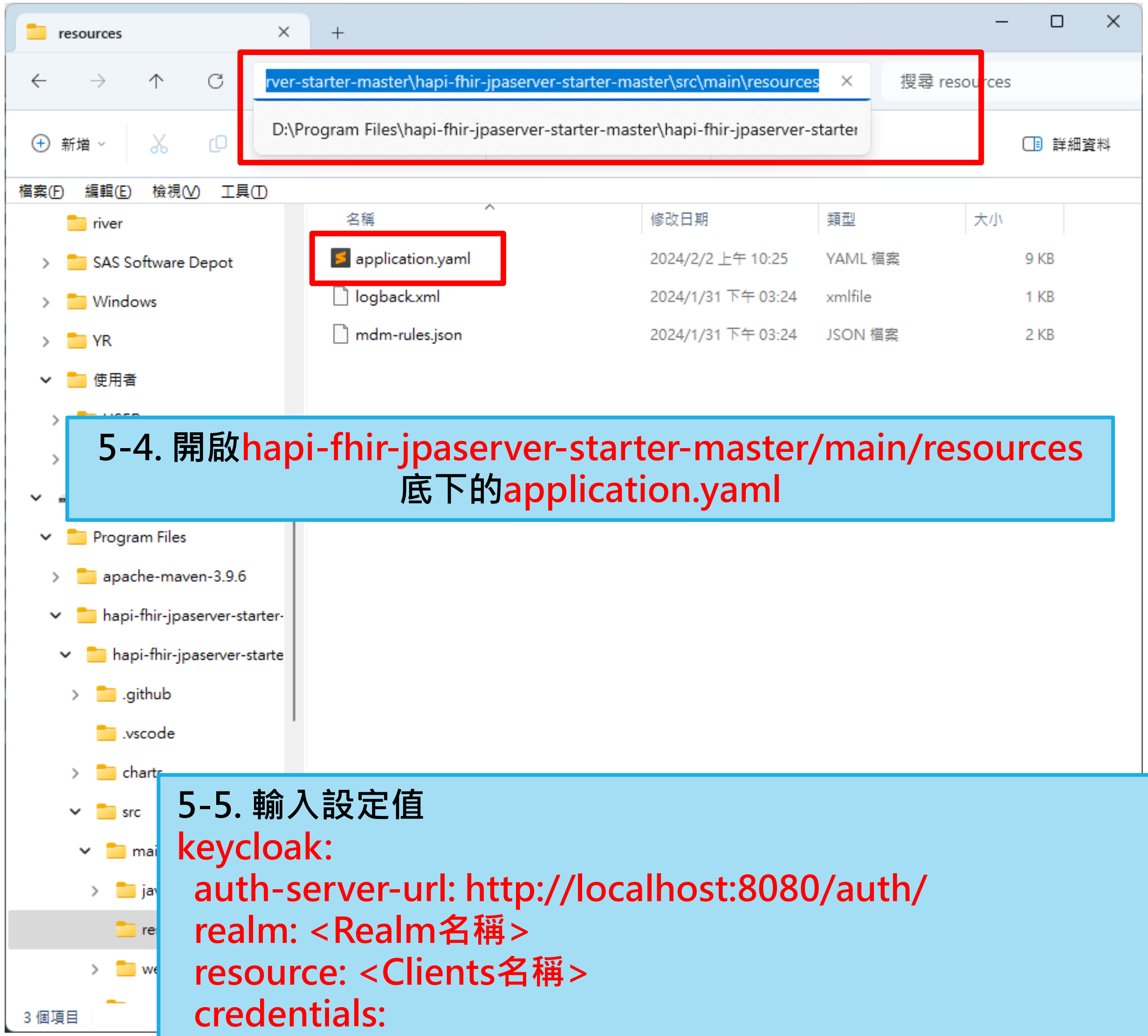


```
		<!-- This dependency includes the CDS Hooks Server -->
		<dependency>
		  <groupId>ca.uhn.hapi.fhir</groupId>
		  <artifactId>hapi-fhir-server-cds-hooks</artifactId>
		  <version>${project.version}</version>
		</dependency>
		<!-- This dependency includes the OpenAPI Server -->
		<dependency>
			<groupId>ca.uhn.hapi.fhir</groupId>
			<artifactId>hapi-fhir-server-openapi</artifactId>
			<version>${project.version}</version>
			<exclusions>
				<exclusion>
					<groupId>org.yaml</groupId>
					<artifactId>snakeyaml</artifactId>
				</exclusion>
			</exclusions>
		</de
		<!--
		<dep






		</de
		<dep




		</dependency>

		<dependency>
			<groupId>org.smartregister</groupId>
			<artifactId>hapi-fhir-keycloak</artifactId>
			<version>0.0.7-SNAPSHOT</version>
		</dependency>

		<!-- This dependency is used to include the IPS Base Implementation -->
		<dependency>
			<groupId>ca.uhn.hapi.fhir</groupId>
			<artifactId>hapi-fhir-jpaserver-ips</artifactId>
			<version>${project.version}</version>
		</dependency>
```

5-3. 在< dependencies></ dependencies>中加入keycloak的設定

```
<dependency>
   <groupId>org.smartregister</groupId>
   <artifactId>hapi-fhir-keycloak</artifactId>
   <version>0.0.7-SNAPSHOT</version>
</dependency>
```

5. 修改HAPI FHIR的設定



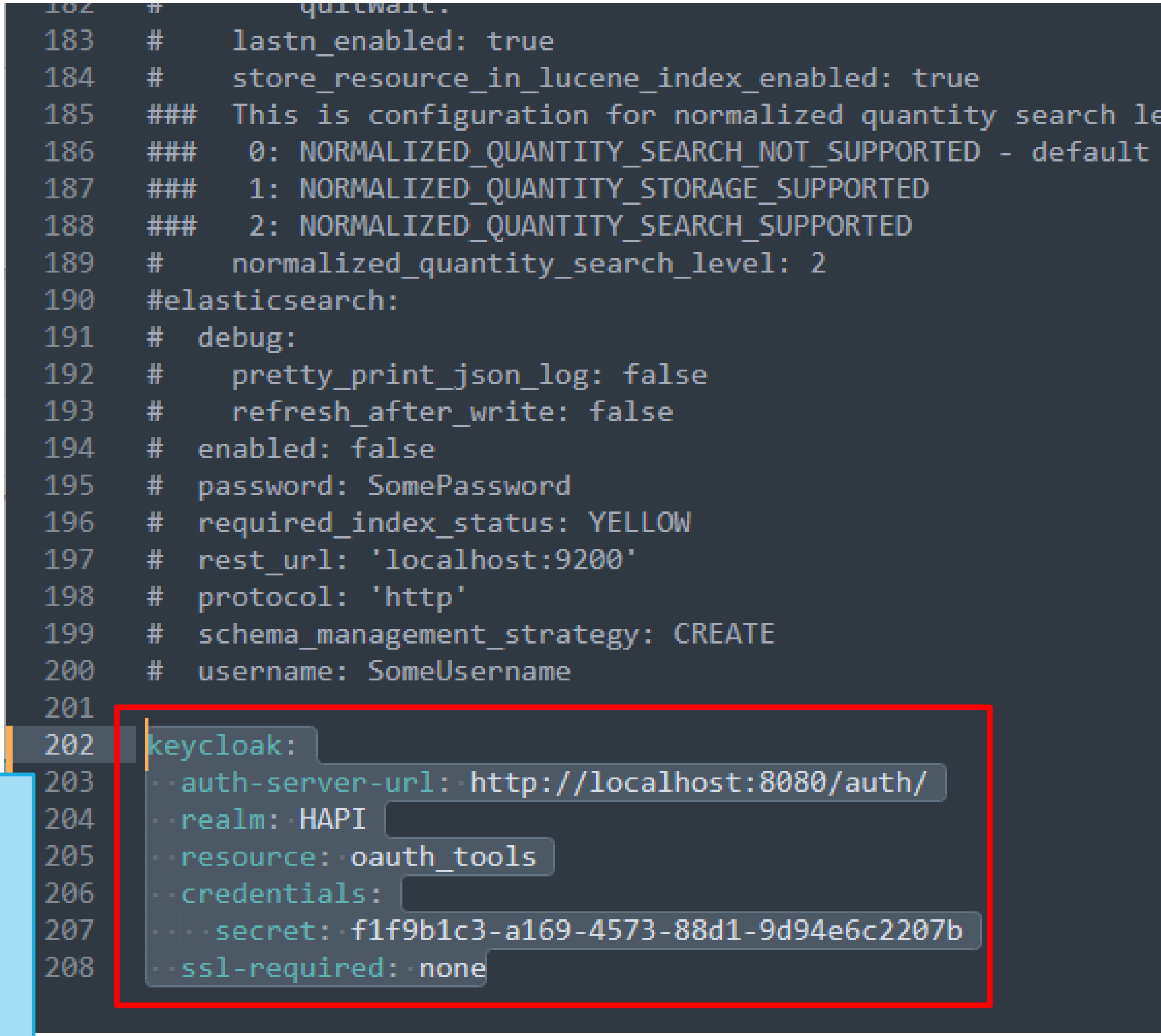
rver-starter-master\hapi-fhir-jpaserver-starter-master\src\main\resources

D:\Program Files\hapi-fhir-jpaserver-starter-master\hapi-fhir-jpaserver-starter

搜尋 resources

名稱	修改日期	類型	大小
application.yaml	2024/2/2 上午 10:25	YAML 檔案	9 KB
logback.xml	2024/1/31 下午 03:24	xmlfile	1 KB
mdm-rules.json	2024/1/31 下午 03:24	JSON 檔案	2 KB

5-4. 開啟hapi-fhir-jpaserver-starter-master/main/resources底下的application.yaml



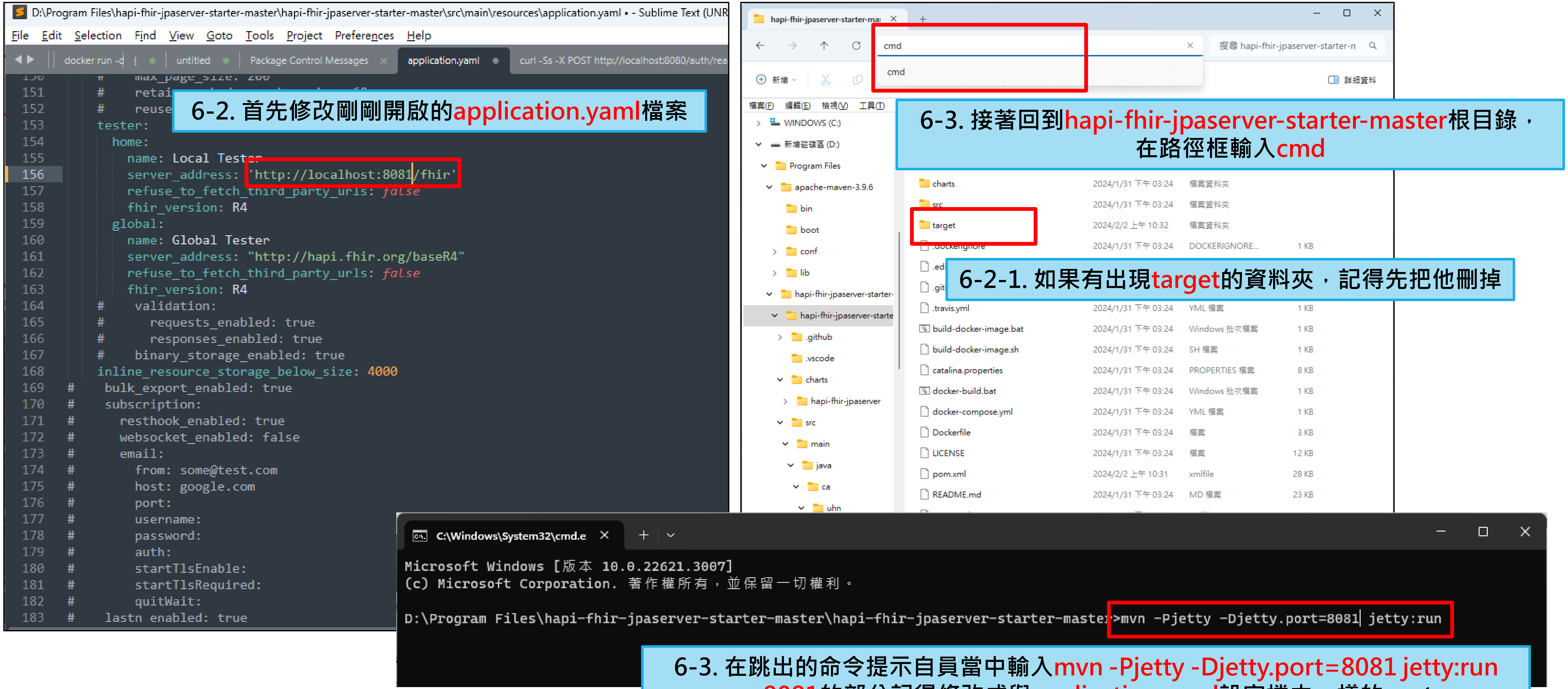
```
#       quitwait:
#     lastn_enabled: true
#     store_resource_in_lucene_index_enabled: true
###  This is configuration for normalized quantity search le
###    0: NORMALIZED_QUANTITY_SEARCH_NOT_SUPPORTED - default
###    1: NORMALIZED_QUANTITY_STORAGE_SUPPORTED
###    2: NORMALIZED_QUANTITY_SEARCH_SUPPORTED
#     normalized_quantity_search_level: 2
#elasticsearch:
#   debug:
#     pretty_print_json_log: false
#     refresh_after_write: false
#   enabled: false
#   password: SomePassword
#   required_index_status: YELLOW
#   rest_url: 'localhost:9200'
#   protocol: 'http'
#   schema_management_strategy: CREATE
#   username: SomeUsername

keycloak:
  auth-server-url: http://localhost:8080/auth/
  realm: HAPI
  resource: oauth_tools
  credentials:
    secret: f1f9b1c3-a169-4573-88d1-9d94e6c2207b
  ssl-required: none
```

5-5. 輸入設定值

```
keycloak:
  auth-server-url: http://localhost:8080/auth/
  realm: <Realm名稱>
  resource: <Clients名稱>
  credentials:
    secret: <剛剛複製的Secret>
  ssl-required: none
```

6. 重新執行HAPI FHIR Server

6-1. 因為直接執行HAPI FHIR Server會跟keycloak的port衝突，因此這裡將HAPI FHIR Server改為以8081port執行



D:\Program Files\hapi-fhir-jpaserver-starter-master\hapi-fhir-jpaserver-starter-master\src\main\resources\application.yaml • - Sublime Text (UNR
File Edit Selection Find View Goto Tools Project Preferences Help
docker run -d
untitled
Package Control Messages
application.yaml
curl -Ss -X POST http://localhost:8080/auth/rea
hapi-fhir-jpaserver-starter-ma
新增
詳細資料
檔案(F) 編輯(E) 檢視(V) 工具(T)
搜尋 hapi-fhir-jpaserver-starter-n
WINDOWS (C:)
新增磁碟區 (D:)
Program Files
apache-maven-3.9.6
bin
boot
conf
lib
hapi-fhir-jpaserver-starter-
hapi-fhir-jpaserver-starte
.github
.vscode
charts
hapi-fhir-jpaserver
src
main
java
ca
uhn
C:\Windows\System32\cmd.e

6-2. 首先修改剛剛開啟的application.yaml檔案

```
    #     max_page_size: 200
    #     retai
    #     reuse
    tester:
      home:
        name: Local Tester
        server_address: 'http://localhost:8081/fhir'
        refuse_to_fetch_third_party_urls: false
        fhir_version: R4
      global:
        name: Global Tester
        server_address: "http://hapi.fhir.org/baseR4"
        refuse_to_fetch_third_party_urls: false
        fhir_version: R4
    #    validation:
    #      requests_enabled: true
    #      responses_enabled: true
    #    binary_storage_enabled: true
    inline_resource_storage_below_size: 4000
#    bulk_export_enabled: true
#    subscription:
#      resthook_enabled: true
#      websocket_enabled: false
#      email:
#        from: some@test.com
#        host: google.com
#        port:
#        username:
#        password:
#        auth:
#        startTlsEnable:
#        startTlsRequired:
#        quitWait:
#    lastn_enabled: true
```

6-3. 接著回到hapi-fhir-jpaserver-starter-master根目錄，在路徑框輸入cmd

```
cmd
cmd
```

charts	2024/1/31 下午 03:24	檔案資料夾	
src	2024/1/31 下午 03:24	檔案資料夾	
target	2024/2/2 上午 10:32	檔案資料夾	
.dockerignore	2024/1/31 下午 03:24	DOCKERIGNORE...	1 KB
.ed			
.git			
.travis.yml	2024/1/31 下午 03:24	YML 檔案	1 KB
build-docker-image.bat	2024/1/31 下午 03:24	Windows 批次檔案	1 KB
build-docker-image.sh	2024/1/31 下午 03:24	SH 檔案	1 KB
catalina.properties	2024/1/31 下午 03:24	PROPERTIES 檔案	8 KB
docker-build.bat	2024/1/31 下午 03:24	Windows 批次檔案	1 KB
docker-compose.yml	2024/1/31 下午 03:24	YML 檔案	1 KB
Dockerfile	2024/1/31 下午 03:24	檔案	3 KB
LICENSE	2024/1/31 下午 03:24	檔案	12 KB
pom.xml	2024/2/2 上午 10:31	xmlfile	28 KB
README.md	2024/1/31 下午 03:24	MD 檔案	23 KB

6-2-1. 如果有出現target的資料夾，記得先把他刪掉

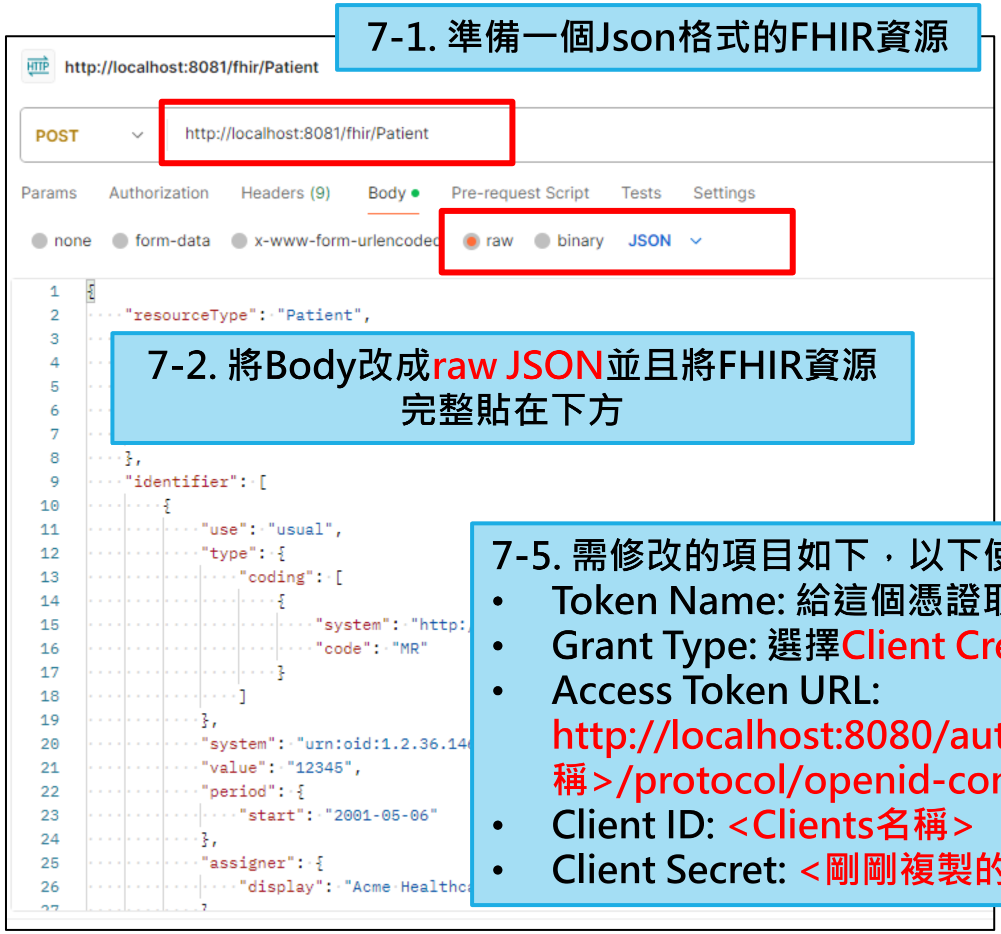
```
Microsoft Windows [版本 10.0.22621.3007]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

D:\Program Files\hapi-fhir-jpaserver-starter-master\hapi-fhir-jpaserver-starter-master>mvn -Pjetty -Djetty.port=8081 jetty:run
```

6-3. 在跳出的命令提示自員當中輸入mvn -Pjetty -Djetty.port=8081 jetty:run，8081的部分記得修改成與application.yaml設定檔中一樣的port

7. 測試HAPI FHIR Server

7-1. 準備一個Json格式的FHIR資源

7-2. 將Body改成raw JSON並且將FHIR資源完整貼在下方



http://localhost:8081/fhir/Patient

POST http://localhost:8081/fhir/Patient

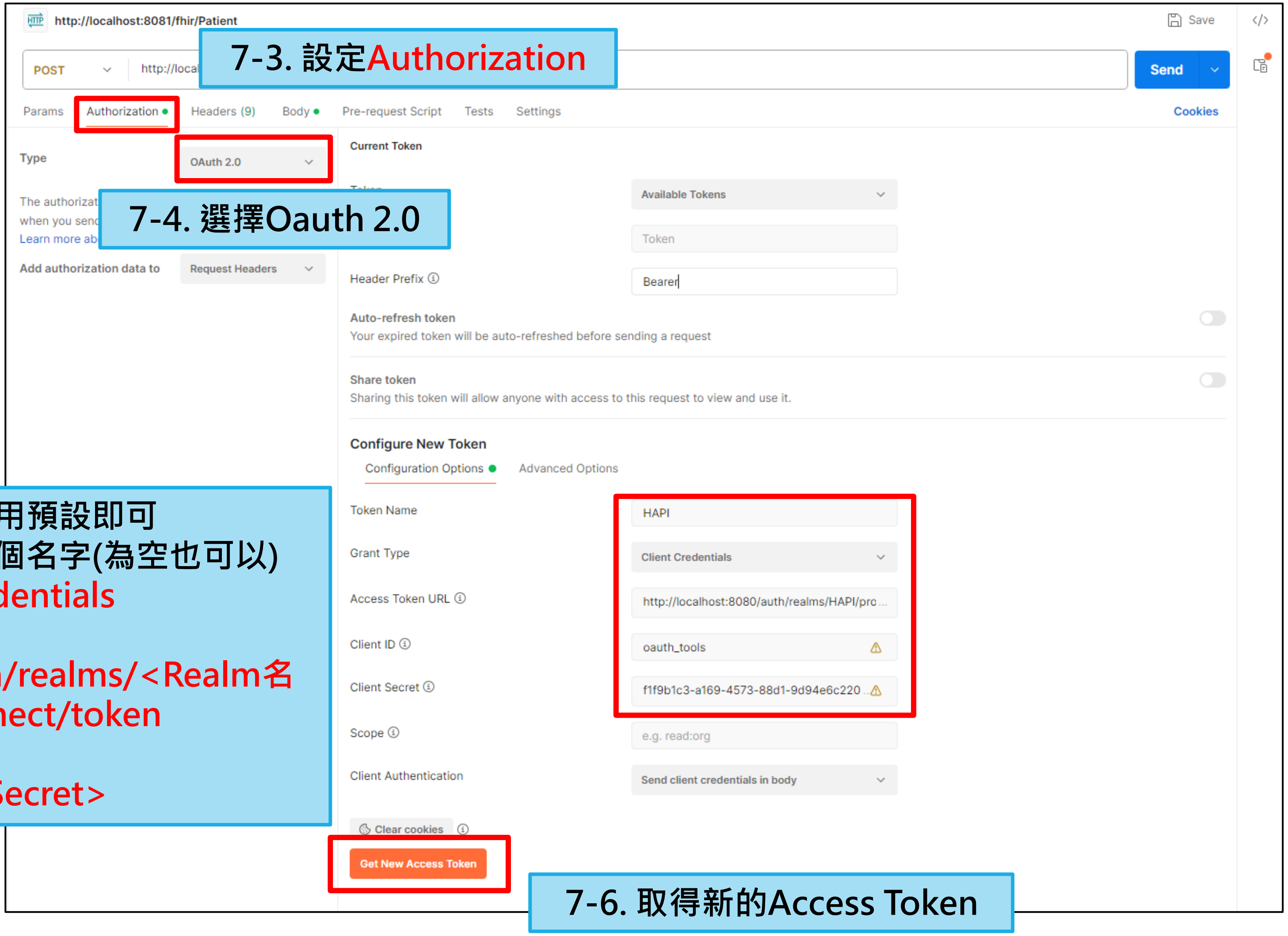
Params Authorization Headers (9) Body Pre-request Script Tests Settings

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary JSON

```
{
    "resourceType": "Patient",
    },
    "identifier": [
        {
            "use": "usual",
            "type": {
                "coding": [
                    {
                        "system": "http:
                        "code": "MR"
                    }
                ]
            },
            "system": "urn:oid:1.2.36.14
            "value": "12345",
            "period": {
                "start": "2001-05-06"
            },
            "assigner": {
                "display": "Acme Healthc
```

7-3. 設定Authorization

7-4. 選擇Oauth 2.0



http://localhost:8081/fhir/Patient

POST http://local

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Type: OAuth 2.0

The authorizat when you send Learn more ab

Add authorization data to: Request Headers

Current Token

Token: Available Tokens

Token

Header Prefix ⓘ: Bearer

Auto-refresh token
Your expired token will be auto-refreshed before sending a request

Share token
Sharing this token will allow anyone with access to this request to view and use it.

Configure New Token

Configuration Options ● Advanced Options

Token Name: HAPI

Grant Type: Client Credentials

Access Token URL ⓘ: http://localhost:8080/auth/realms/HAPI/pro...

Client ID ⓘ: oauth_tools

Client Secret ⓘ: f1f9b1c3-a169-4573-88d1-9d94e6c220...

Scope ⓘ: e.g. read:org

Client Authentication: Send client credentials in body

Clear cookies ⓘ

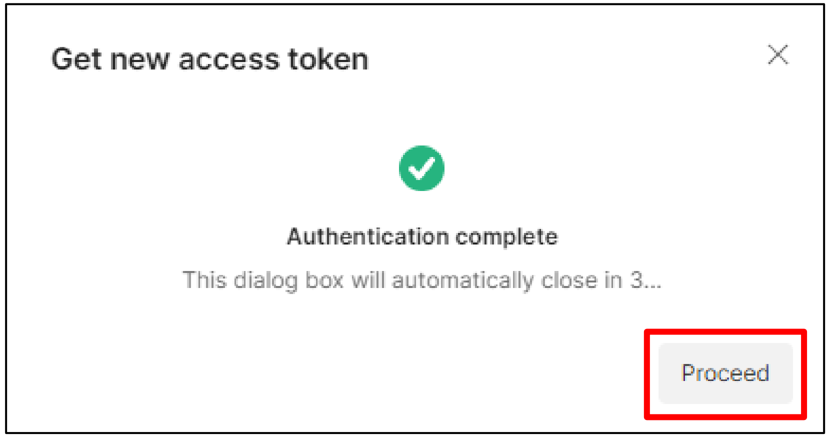
Get New Access Token

7-5. 需修改的項目如下，以下使用預設即可

- Token Name: 給這個憑證取個名字(為空也可以)
- Grant Type: 選擇Client Credentials
- Access Token URL: http://localhost:8080/auth/realms/<Realm名稱>/protocol/openid-connect/token
- Client ID: <Clients名稱>
- Client Secret: <剛剛複製的Secret>

7-6. 取得新的Access Token

7. 測試HAPI FHIR Server

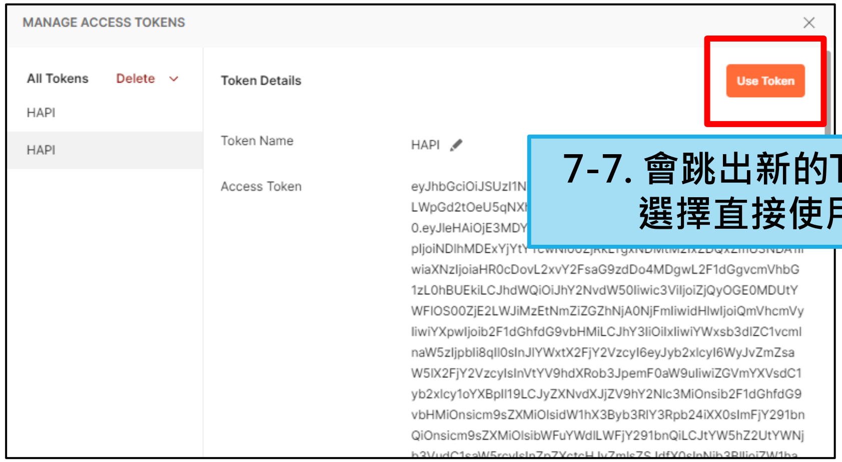


Get new access token

Authentication complete

This dialog box will automatically close in 3...

7-7. 可以等他倒數完，也可以直接按Proceed



MANAGE ACCESS TOKENS

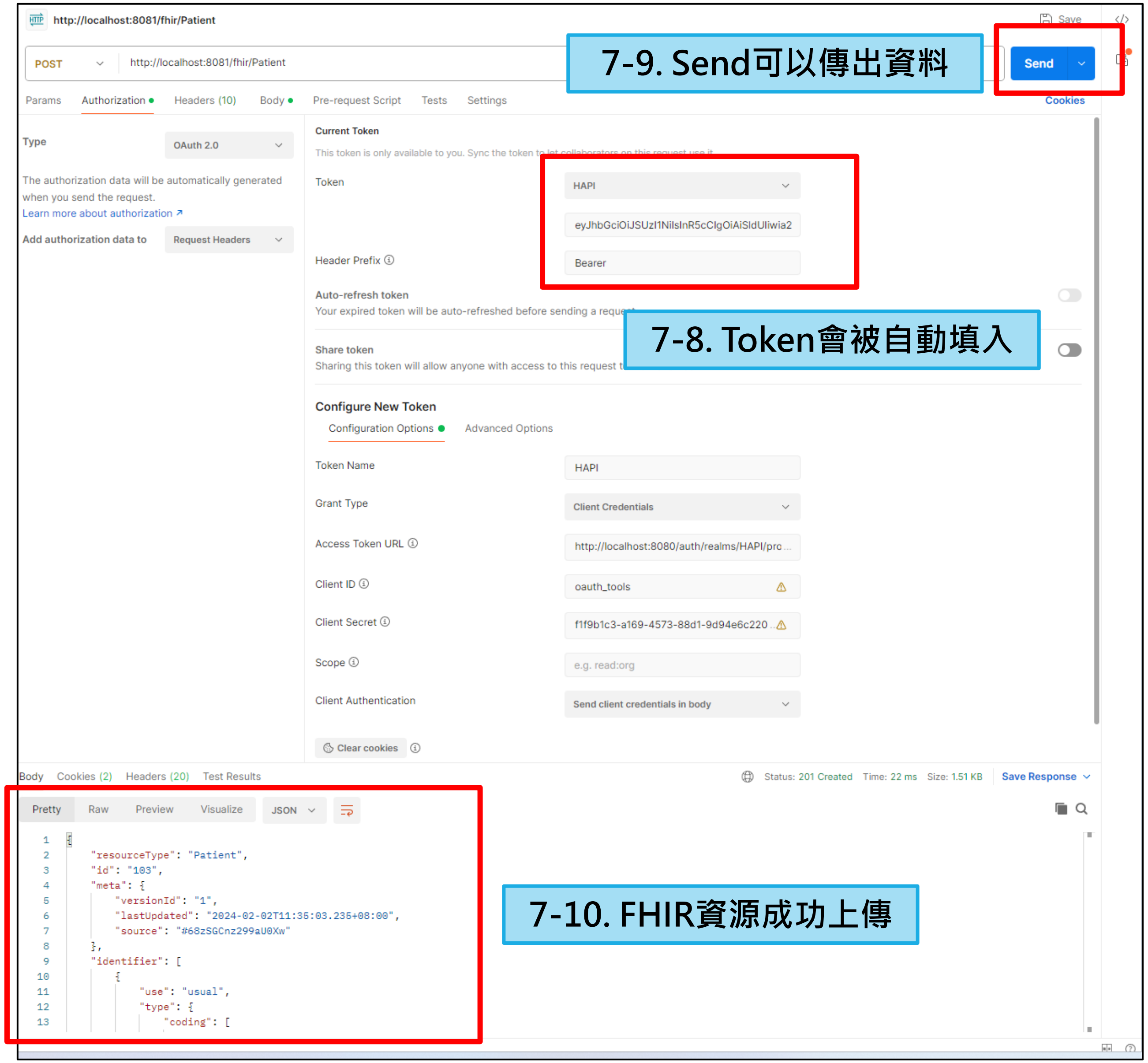
Token Details

Token Name: HAPI

Access Token:

```
eyJhbGciOiJSUzI1N
LWpGd2tOeU5qNX
0.eyJleHAiOjE3MDY
pljoiNDlhMDExYjYtY
wiaXNzIjoiaHR0cDovL2xvY2FsaG9zdDo4MDgwL2F1dGgvcmVhbG
1zL0hBUEkiLCJhdWQiOiJhY2NvdW50Iiwic3ViIjoiZjQyOGE0MDUtY
WFlOS00ZjE2LWJiMzEtNmZiZGZhNjA0NjFmIiwidHlwIjoiQmVhcmVy
IiwiYXpwIjoib2F1dGhfdG9vbHMiLCJhY3IiOiIxIiwiYWxsb3dlZC1vcml
naW5zIjpbIi8qIl0sInJlYWxtX2FjY2VzcyI6eyJyb2xlcyI6WyJvZmZsa
W5lX2FjY2VzcyIsInVtYV9hdXRob3JpemF0aW9uIiwiZGVmYXVsdC1
yb2xlcy1oYXBpIl19LCJyZXNvdXJjZV9hY2Nlc3MiOnsib2F1dGhfdG9
vbHMiOnsicm9sZXMiOlsidW1hX3Byb3RlY3Rpb24iXX0sImFjY291bn
QiOnsicm9sZXMiOlsibWFuYWdlLWFjY291bnQiLCJtYW5hZ2UtYWNj
```

7-7. 會跳出新的Token
選擇直接使用



http://localhost:8081/fhir/Patient

URL: http://localhost:8081/fhir/Patient

Type: OAuth 2.0

The authorization data will be automatically generated when you send the request.
Learn more about authorization ↗

Add authorization data to: Request Headers

Current Token

This token is only available to you. Sync the token to let collaborators on this request use it.

Token: HAPI
eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2

Header Prefix ⓘ: Bearer

Auto-refresh token
Your expired token will be auto-refreshed before sending a request.

Share token
Sharing this token will allow anyone with access to this request t

Configure New Token

Token Name: HAPI

Grant Type: Client Credentials

Access Token URL ⓘ: http://localhost:8080/auth/realms/HAPI/prc

Client ID ⓘ: oauth_tools

Client Secret ⓘ: f1f9b1c3-a169-4573-88d1-9d94e6c220

Scope ⓘ: e.g. read:org

Client Authentication: Send client credentials in body

Status: 201 Created   Time: 22 ms   Size: 1.51 KB

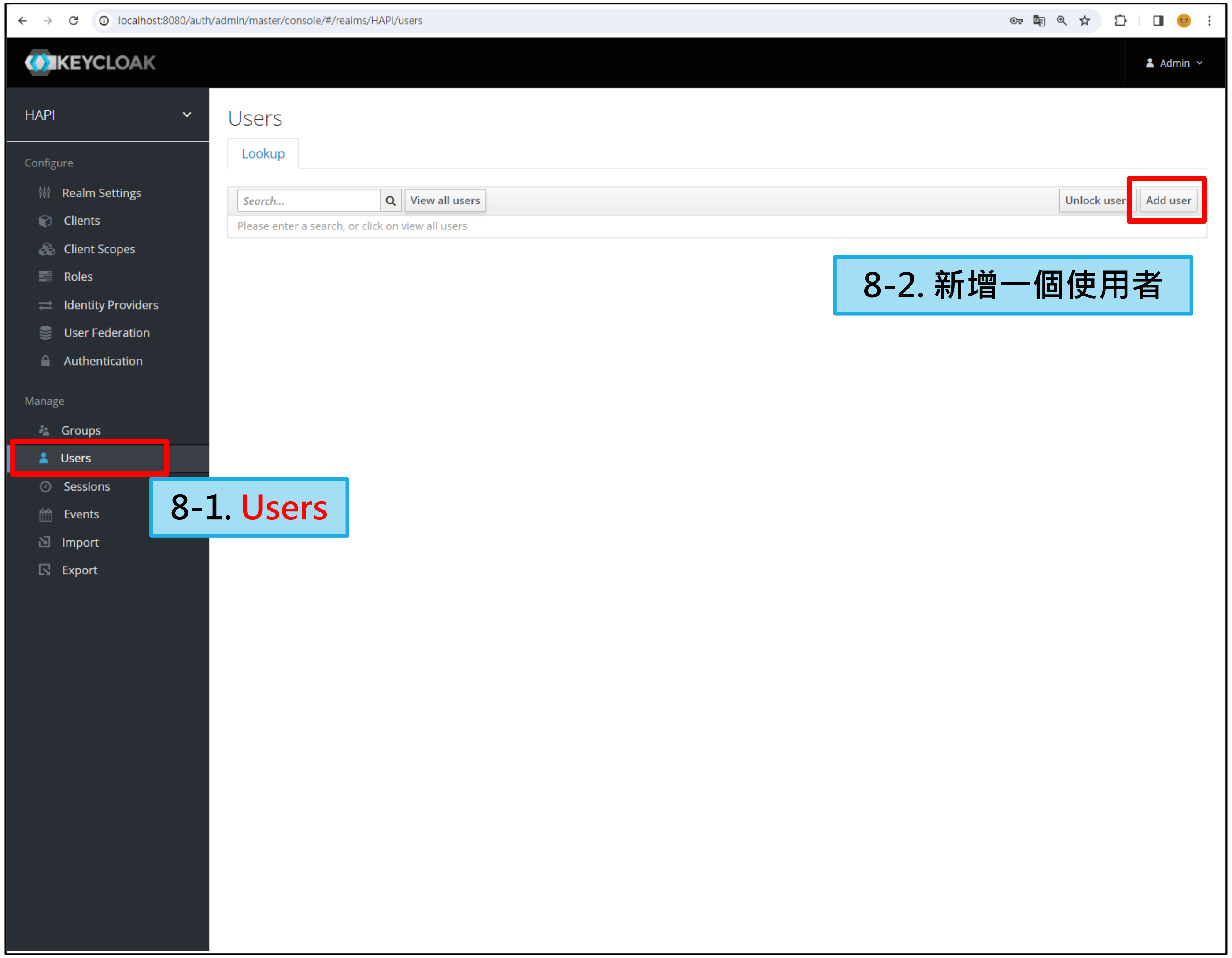
```
{
    "resourceType": "Patient",
    "id": "103",
    "meta": {
        "versionId": "1",
        "lastUpdated": "2024-02-02T11:35:03.235+08:00",
        "source": "#68zSGCnz299aU0Xw"
    },
    "identifier": [
        {
            "use": "usual",
            "type": {
                "coding": [
```

7-9. Send可以傳出資料

7-8. Token會被自動填入

7-10. FHIR資源成功上傳

8. 新增使用者



localhost:8080/auth/admin/master/console/#/realms/HAPI/users

Users

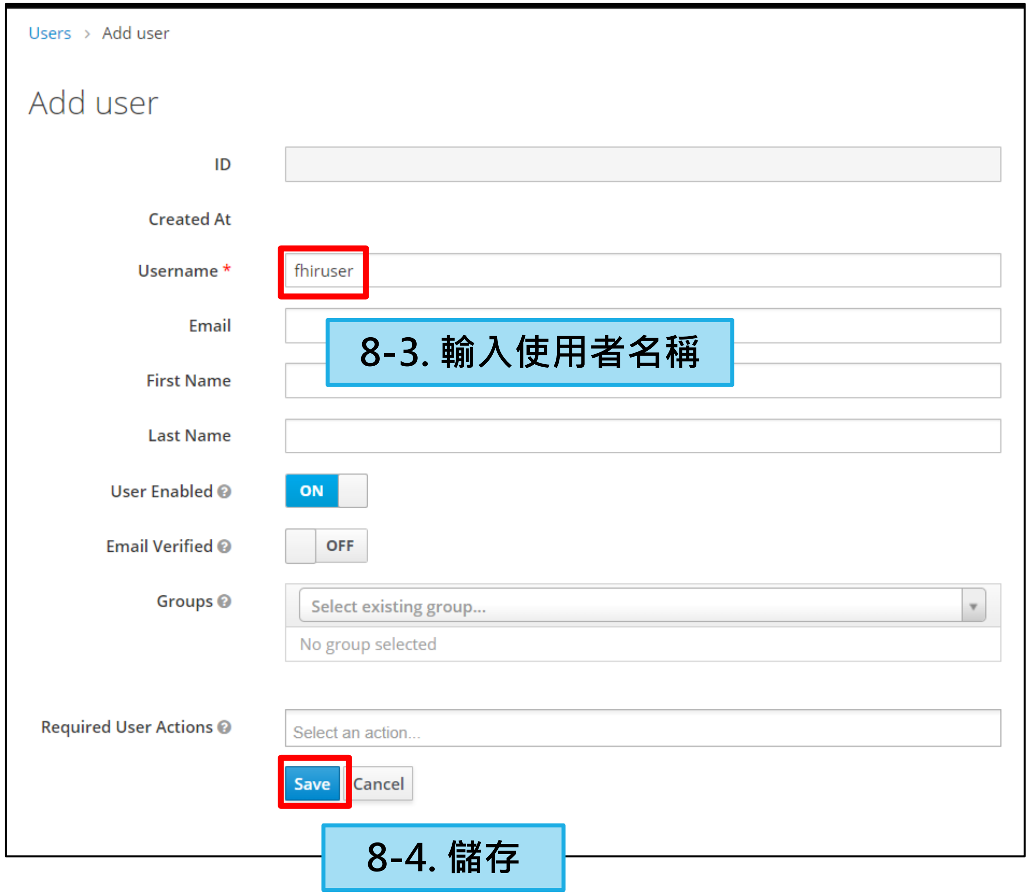
Lookup

Search... | View all users | Unlock users | Add user

Please enter a search, or click on view all users

8-1. Users

8-2. 新增一個使用者



Users › Add user

Add user

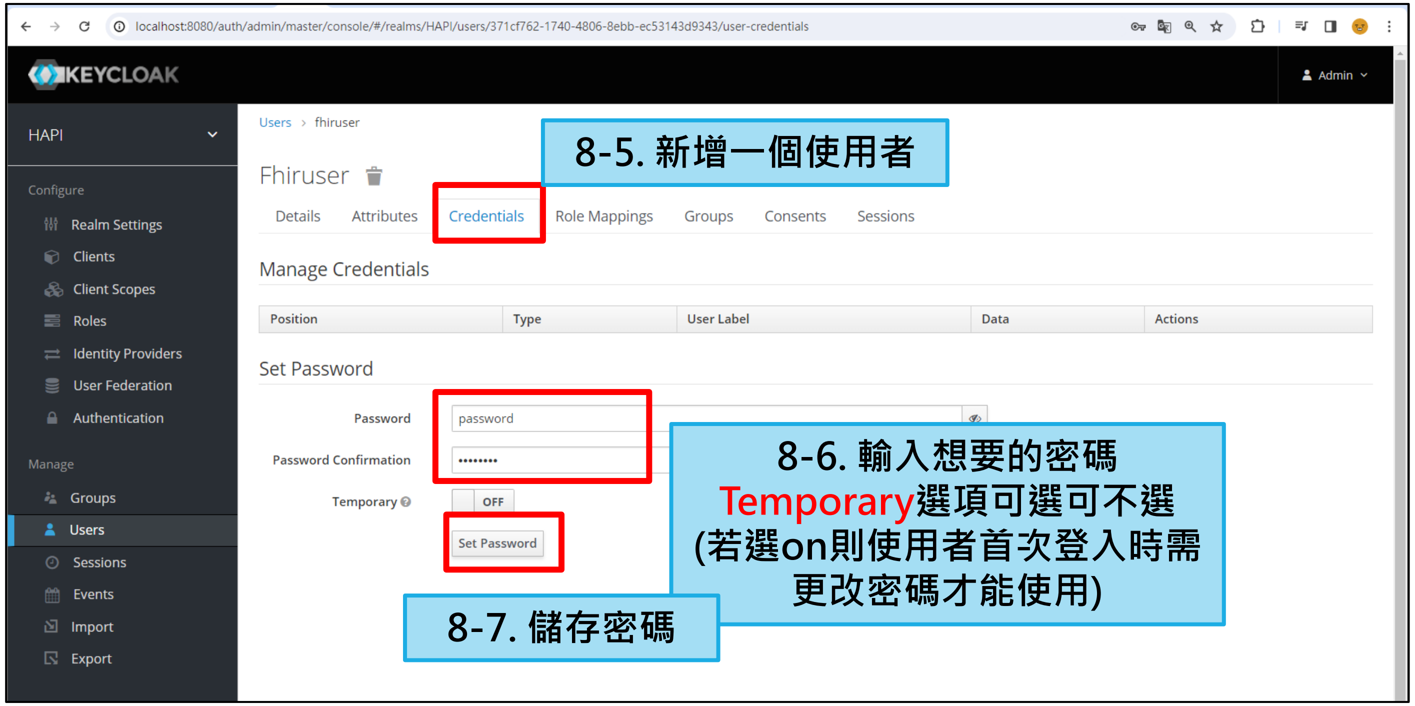
ID	
Created At	
Username *	fhiruser
Email	
First Name	
Last Name	
User Enabled	ON
Email Verified	OFF
Groups	Select existing group... No group selected
Required User Actions	Select an action...

Save Cancel

8-3. 輸入使用者名稱

8-4. 儲存

8. 新增使用者



localhost:8080/auth/admin/master/console/#/realms/HAPI/users/371cf762-1740-4806-8ebb-ec53143d9343/user-credentials

Users › fhiruser

Fhiruser

Manage Credentials

Position	Type	User Label	Data	Actions

Set Password

Password: password

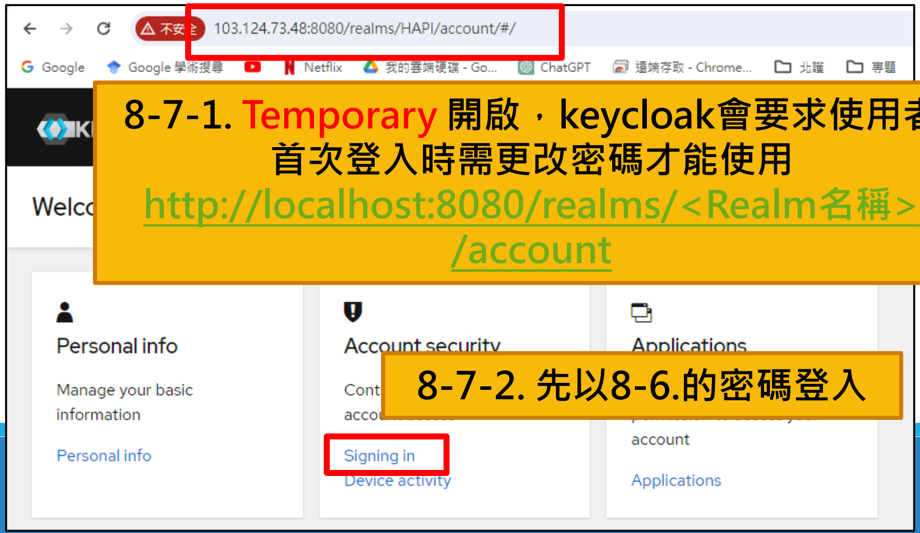
Password Confirmation: ••••••••

Temporary: OFF

8-5. 新增一個使用者

8-6. 輸入想要的密碼
Temporary選項可選可不選
(若選on則使用者首次登入時需
更改密碼才能使用)

8-7. 儲存密碼



103.124.73.48:8080/realms/HAPI/account/#/

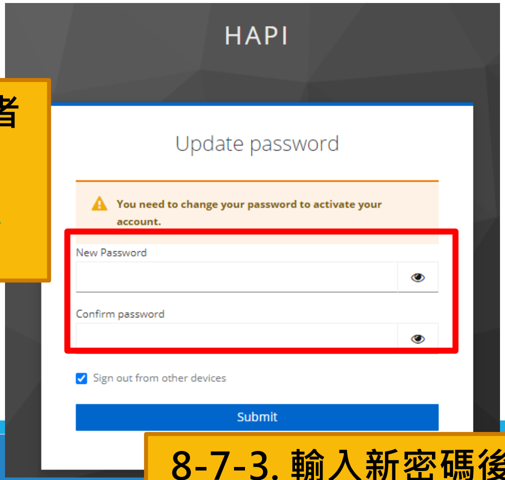
Manage your basic information

Cont... acco...

...account

8-7-1. Temporary 開啟，keycloak會要求使用者
首次登入時需更改密碼才能使用
http://localhost:8080/realms/<Realm名稱>/account

8-7-2. 先以8-6.的密碼登入



Update password

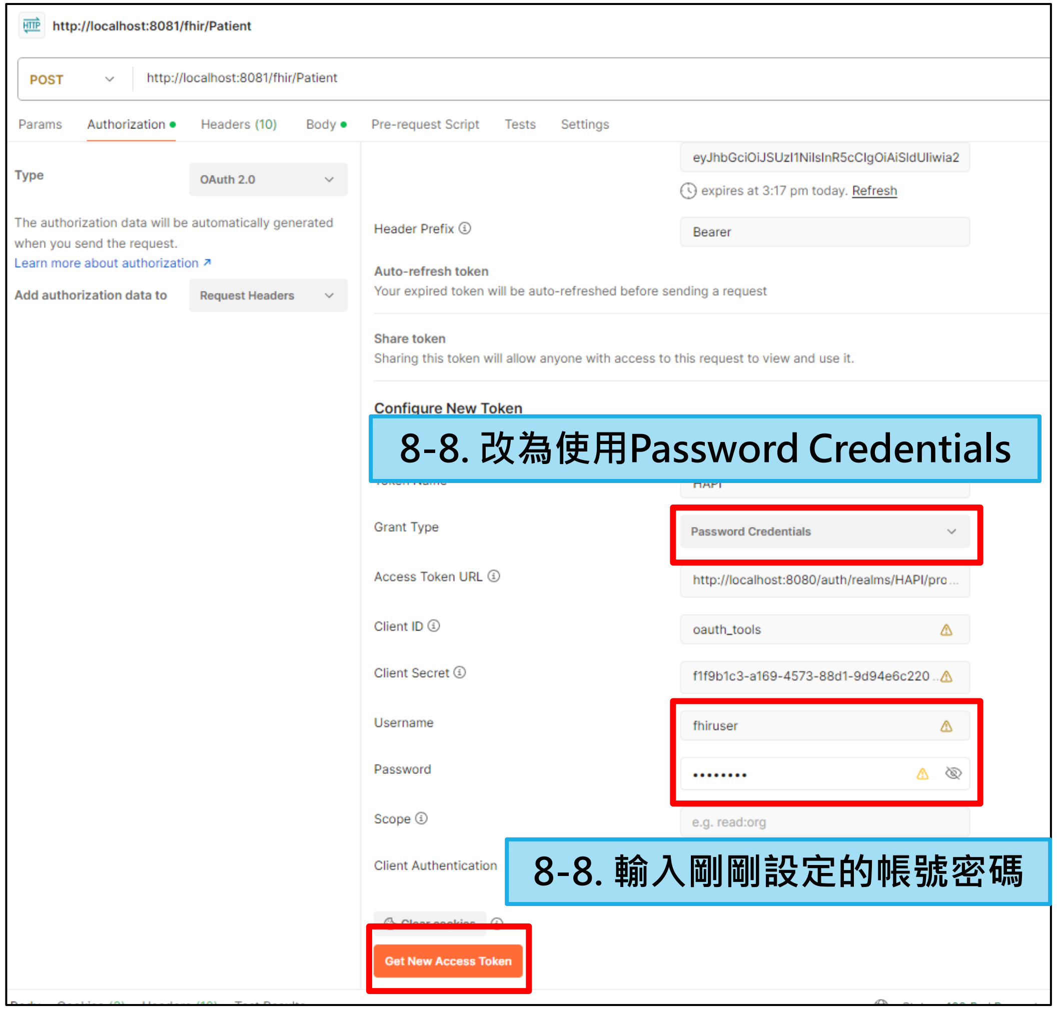
You need to change your password to activate your account.

New Password

Confirm password

☑ Sign out from other devices

8-7-3. 輸入新密碼後送出



http://localhost:8081/fhir/Patient

POST http://localhost:8081/fhir/Patient

Type: OAuth 2.0

The authorization data will be automatically generated when you send the request.
Learn more about authorization ↗

Add authorization data to: Request Headers

eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2

expires at 3:17 pm today.

Header Prefix: Bearer

Auto-refresh token
Your expired token will be auto-refreshed before sending a request

Share token
Sharing this token will allow anyone with access to this request to view and use it.

Configure New Token

Token Name: HAPI

Grant Type: Password Credentials

Access Token URL: http://localhost:8080/auth/realms/HAPI/pro...

Client ID: oauth_tools

Client Secret: f1f9b1c3-a169-4573-88d1-9d94e6c220...

Username: fhiruser

Password: •••••••••

Scope: e.g. read:org

Client Authentication

8-8. 改為使用Password Credentials

8-8. 輸入剛剛設定的帳號密碼

8-9. 一樣去取得Token後就
可以上傳FHIR資源了