



SMART App **Submission Briefing** & **Workshop**

A collection of various medical and healthcare icons, including a heart with a bar chart, a microscope, a DNA helix, a heart with a cross, and a heart with a pulse line, arranged in a grid.

課程設計規劃說明

- 本課程及SMART平台由衛福部資訊處李建璋處長設計規劃。
- 工研院團隊執行及教學。
- 透過SMART on FHIR的導入，加值FHIR生態系之應用。
- 感謝國內學協會對於FHIR標準的推動。

引用工具：

本網站所提供之 Sandbox 測試環境使用以下開源套件

- ✓ [SMART-ON-FHIR/SMART-LAUNCHER-V2](#)

Copyright © Copyright 2018 Boston Children's Hospital
License: [Apache-2.0 license](#)

- ✓ [HAPIFHIR/HAPI-FHIR](#)

Copyright © Copyright 2015 Smile CDR Inc.
License: [Apache-2.0 license](#)

- ✓ [SMART-ON-FHIR/GENERATED-SAMPLE-DATA](#)

Copyright © Copyright 2018 Boston Children's Hospital
License: [Apache-2.0 license](#)

線上資源：

- ✓ [SMART -- 教學課程 – JavaScript](#)
- ✓ [SMART on FHIR JavaScript 程式庫 | SMART JS 用戶端程式庫](#)
- ✓ [SMART on FHIR Python Client: SMART FHIR Client](#)

Outline

1. SMART on FHIR：可替代的醫療應用程式和可重複使用的技術

- ✓ 如何開發一個SMART App
- ✓ SMART App測試
 - ✓ Launch環境
 - ✓ EMR中啟動

2. App提案及上架流程

3. SMART App實作練習

SMART on FHIR：可替代的醫療應用程式和可重複使用的技術

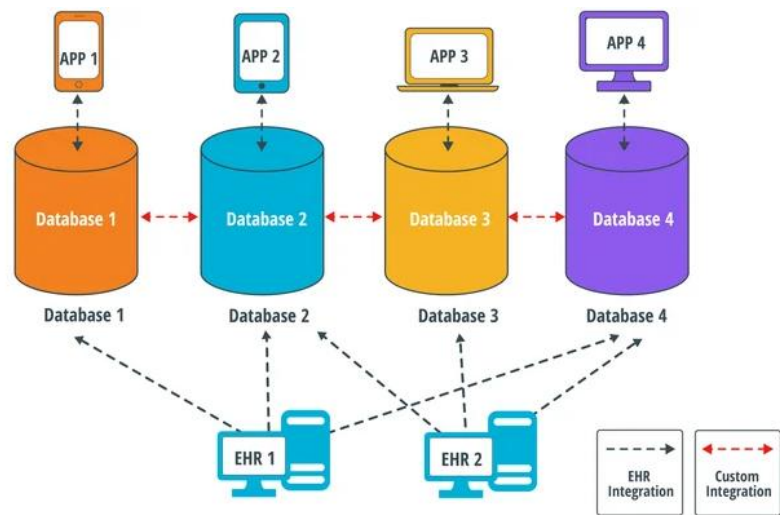
- 如何開發一個SMART App



可替代的醫療應用程式和可重複使用的技術

SMART和FHIR代表了一種開放、標準化和實用的方式

- ✓ 應用程式有各自專屬的數據庫、數據模型和介面。
。由於**缺乏標準化**，部署的每個新應用程式都須投入資源進行維護。
- ✓ 如果應用程式過時或停止維護，則會**遺留下數據孤島**。

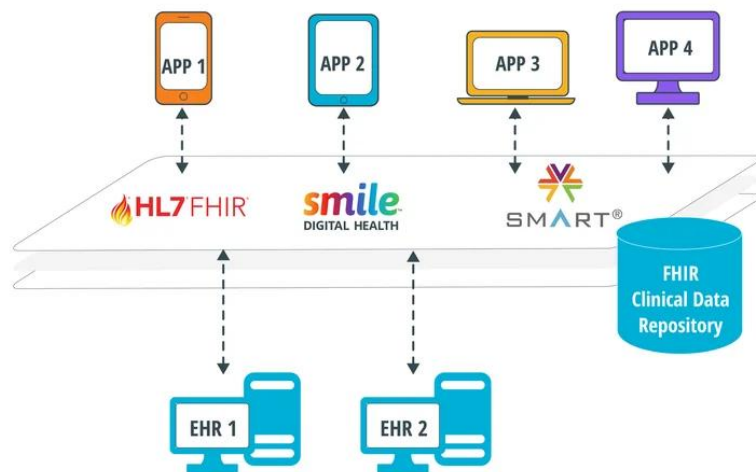


AS IS

FHIR 標準化數據框架

SMART on FHIR

SMART 標準化數據訪問



TO BE

SMART on FHIR 是在 FHIR 基礎上加入 OAuth2 + OpenID Connect 的安全框架，讓 App 可以安全地從 EHR 或 FHIR Server 存取病人資料。



臺灣50優良SMART on FHIR

應用程式徵案

SMART App [Submission Briefing](#) & [Workshop](#)

 SMART® App Gallery

[Add New Listing](#)

[Your Listings](#)



[Login](#)

Featured Apps

All Apps

APPLICATION TYPE

CATEGORIES

Care Coordination
Clinical Research
Data Visualization
Disease Management
Genomics
Medication
Patient Engagement
Population Health
Risk Calculation
FHIR Tools
COVID-19
Telehealth

OS SUPPORT

FHIR SUPPORT

SPECIALTY

PRICING

Open Source
Free
Per User
Site-Based
Other

DESIGNED FOR

EHR SUPPORT

All Apps 125

Sort: Name (A-Z) ▾



1upHealth - Aggregated Patient Data

1upHealth

Helps providers view patient data aggregated from external health systems. Patients can connect their medical data sources using FHIR.

[View](#)

OS: Web **Specialties:** Cardiology, Pediatrics, Trauma **Designed for:** Patients & Clinicians



Abstractive Health

Abstractive Health

Abstractive Health is a physician AI assistant that helps you curate the optimal patient record.

[View](#)

Designed for: Clinicians



ACT.md

ACT.md

ACT.md extends EMR's across the community, removing the silos that prevent you from addressing social determinants of health.

[View](#)

OS: Web, Android, iOS **Specialties:** Oncology, Pediatrics, Rheumatology **Designed for:** Patients & Clinicians

App Galleries

•SMART App Gallery:

<http://apps.smarthealthit.org/>

•Cerner App Gallery

<https://code.cerner.com/apps>

•Epic App Orchard

<https://apporchard.epic.com/Gallery>

•Allscripts Application Store

<https://allscriptsstore.cloud.prod.iapps.com/>

臺灣50優良SMART on FHIR
應用程式徵案

SMART App [Submission Briefing](#) & [Workshop](#)



[首頁](#) [SMART Marketplace](#) [Sand box](#) [SMART 提案](#) [登入](#) [註冊](#)

歡迎來到
SMART

Steps to Build a SMART on FHIR App

選擇應用程式類型

- ✓ 面向提供者或患者的應用程式
- ✓ 行動應用程式
- ✓ 網路應用程式

Software Libraries for Developers



使用者安全性驗證



確保提供者可以在整合應用程式之間切換，而毋需每次都輸入其憑證。也不需要第三方解決方案中輸入密碼，因為授權是透過他們的EHR系統進行的。

常規的符合 HIPAA 的應用程式開發程式（包括資料加密、安全連線的使用等）也適用於此。

使用 SMART 沙盒進行測試

Public Sandboxes for Testing



對於所有SMART on FHIR的開發者，建議使用 SMART 沙盒來測試應用程式的功能。Epic 等 EHR 供應商也提供單獨的沙盒，透過使用其 EHR系統測試開發者的產品。

DEVELOPER RESOURCES

SMART ON FHIR API

SMART/HL7 BULK
DATA ACCESS (FLAT
FHIR)

ELECTRONIC HEALTH
INFORMATION (EHI)
EXPORT API

SMART MARKERS – A
FRAMEWORK FOR
PATIENT GENERATED
DATA

CDS HOOKS

SYNC FOR SCIENCE

HEALTH CARDS

SMART ON FHIR
GENOMICS

CUMULUS: A
UNIVERSAL SIDECAR
FOR A SMART
LEARNING
HEALTHCARE
SYSTEM



Developer Resources

SMART on FHIR

- **Technical Documentation** <http://docs.smarthealthit.org/>
- **EHR FHIR & SMART Support – published conformance** <http://docs.smarthealthit.org/>
- **FHIR Resource Editor** <http://docs.smarthealthit.org/fred/>

FHIR Bulk Data

- **Bulk Data Reference Server** <https://bulk-data.smarthealthit.org/>
The first bulk-data server to be created. Supports the standard completely, except for the 'bulkData' parameter. This server is designed for client developers and Includes a variety of useful options. Includes a browser-based bulk-data client designed to be launched from within a web browser. As a client-side app it works only with open servers but it is useful for developers, presentations and demonstrations. It is also a great way to download some sample bulk data as JSON without having to know how the bulk-data works.
- **Bulk Data Client** <https://github.com/smart-on-fhir/sample-apps-stu3/tree/master/fhir-downloader>
A bulk-data command-line client written in NodeJS. Supports streaming and gzip. Includes a detailed graphical output. Supports JWKS authentication.
- **Bulk Data Tester** <https://bulk-data-tester.smarthealthit.org/>
Web based online tester for bulk data servers. Contains big open source test suite and test runner. Can be used online, offline as CLI or as a remote service thanks to it's public HTTP testing API.

Standards and Specifications

- Foundational
 - [FHIR](#): Fast Healthcare Interoperability Resources. Web standard for health interoperability
 - [CDS Hooks](#): Clinical Decision Support Hooks. Web standard for CDS in the EHR workflow
- Data access
 - [US Core Data Profiles](#): FHIR data profiles for health data in the US ("core data for interoperability")
 - [FHIR Bulk Data API Implementation Guide](#): FHIR export API for large-scale data access
- UI and Security Integration
 - [SMART App Launch](#): User-facing apps that connect to EHRs and health portals
 - [SMART Backend Services](#): Server-to-server FHIR connections

啟動器實施指南
(授權及身分驗證)



Tutorials

- [Getting started with Browser-based Apps](#): Tutorial to create
- [Cerner's Browser-based app tutorial](#): In-depth tutorial to build
- [Getting started with CDS Hooks](#): Tutorial to create a simple
- [Getting started for EHRs](#): Tutorial to SMART-enable a clinica

Software Libraries

- [JavaScript or TypeScript](#): Client-side and server-side library
- [Node.js from Vermonster](#): An alternative Node.js implement
- [Python](#): Server-side Python library with support for SMART
- [R](#)
- [Ruby](#)
- [Swift \(iOS\)](#)
- [Java](#)
- [.NET](#): FHIR client library from Firely

Tutorial - Building a JavaScript App



Getting started

The SMART on FHIR JavaScript client library helps you build browser-based SMART apps that interact with a FHIR REST API server. It can help your app get authorization tokens, provide information about the user and patient record in context, and issue API calls to fetch clinical data.

This tutorial describes how to create a basic client-side JavaScript SMART app with no additional libraries and frameworks.

To get started with the SMART on FHIR JavaScript client library, you'll need to:

Step 1. Include a `script` tag

Include a `script` tag referencing the library. The latest code is always available from the NPM CDN at <https://cdn.jsdelivr.net/npm/fhirclient/build/fhir-client.js>. Including this script will create a global `FHIR` object for you to work with. **Example:**

```
<script src="https://cdn.jsdelivr.net/npm/fhirclient/build/fhir-client.js"></script>
```

Step 2. Test against an open FHIR server

Now that you have the client library included, it might be a good idea to play with it a little bit and see what it is capable of. This is easy to do if we use an open FHIR server that does not require authentication.

Here is something basic to start with:

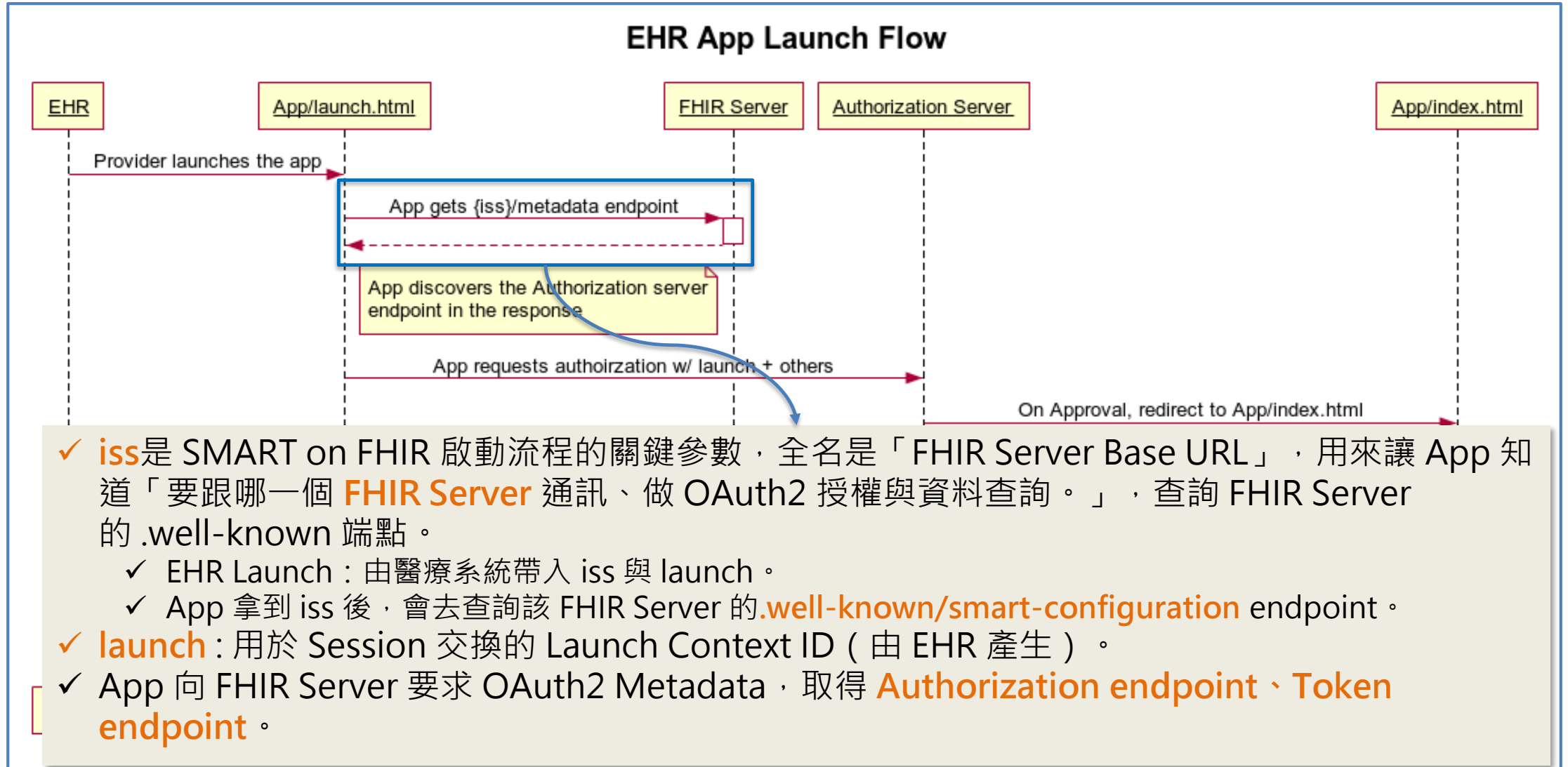
Connect to an open server and browse patients

```
const client = FHIR.client("https://r3.smarthealthit.org");  
client.request("Patient").then(console.log).catch(console.error);
```

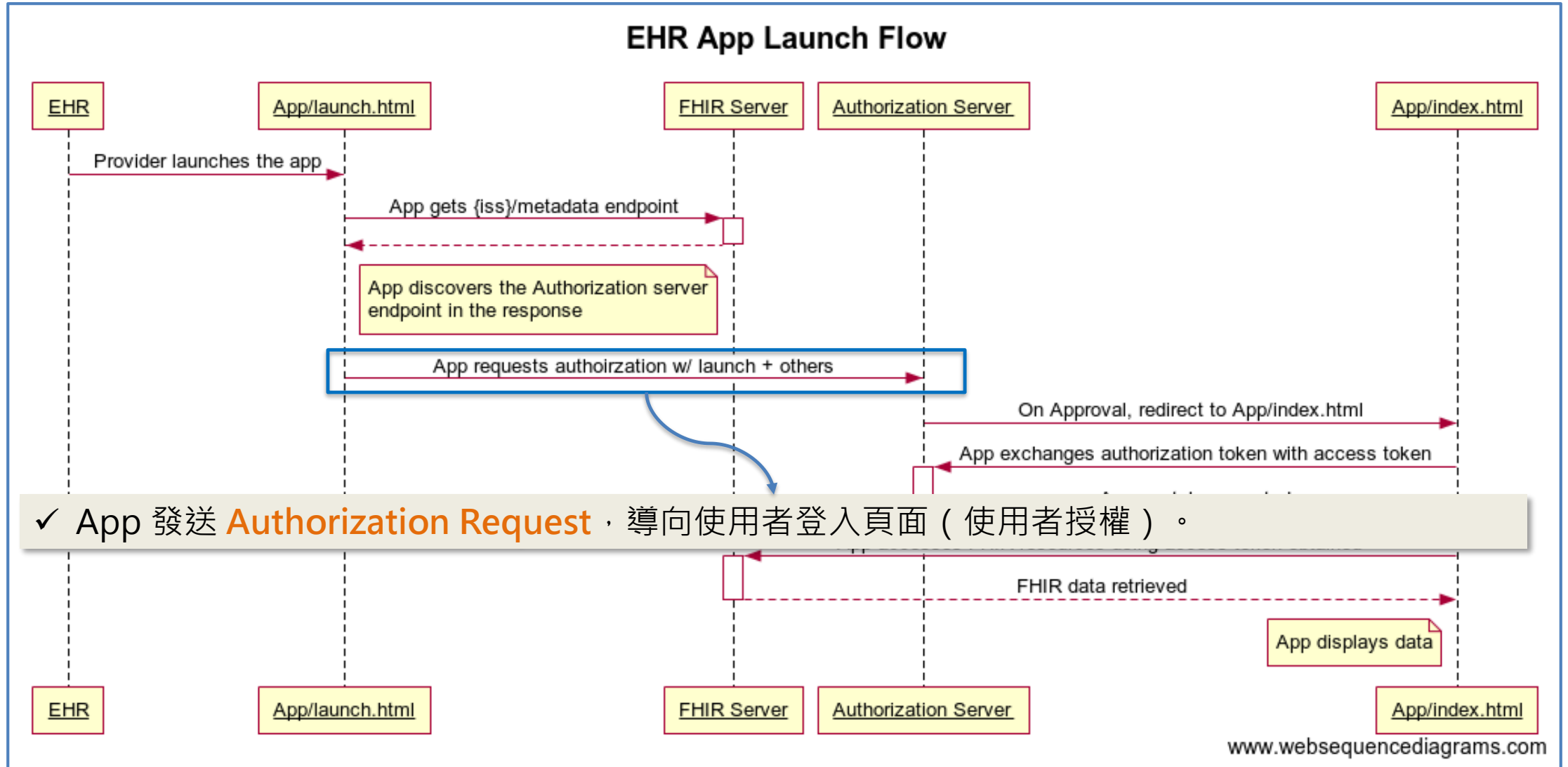
Once this is working you can try [other examples](#) and then proceed to the next step.

Step 3. SMART Authorization

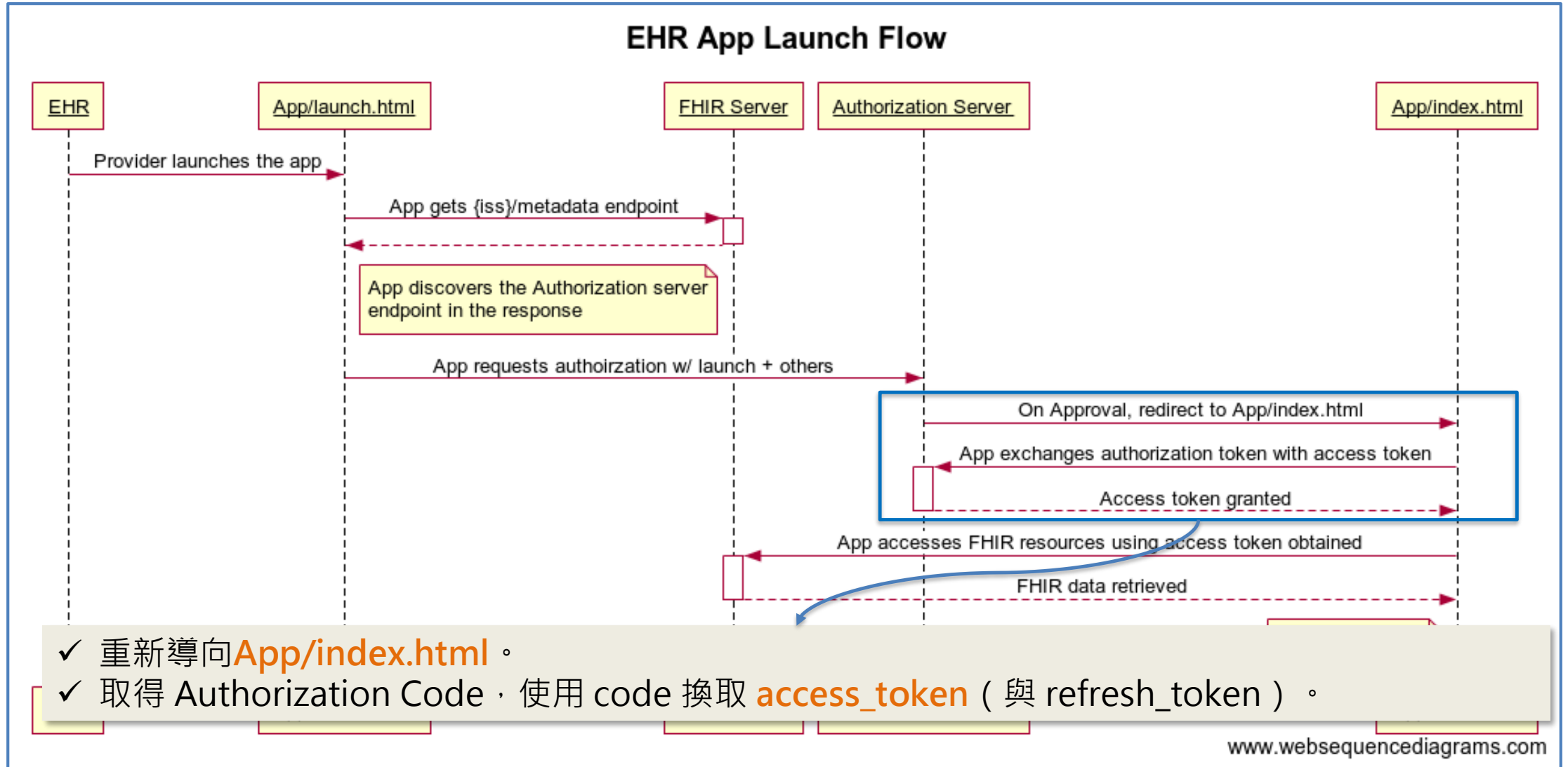
SMART on FHIR – 認證流程(I)



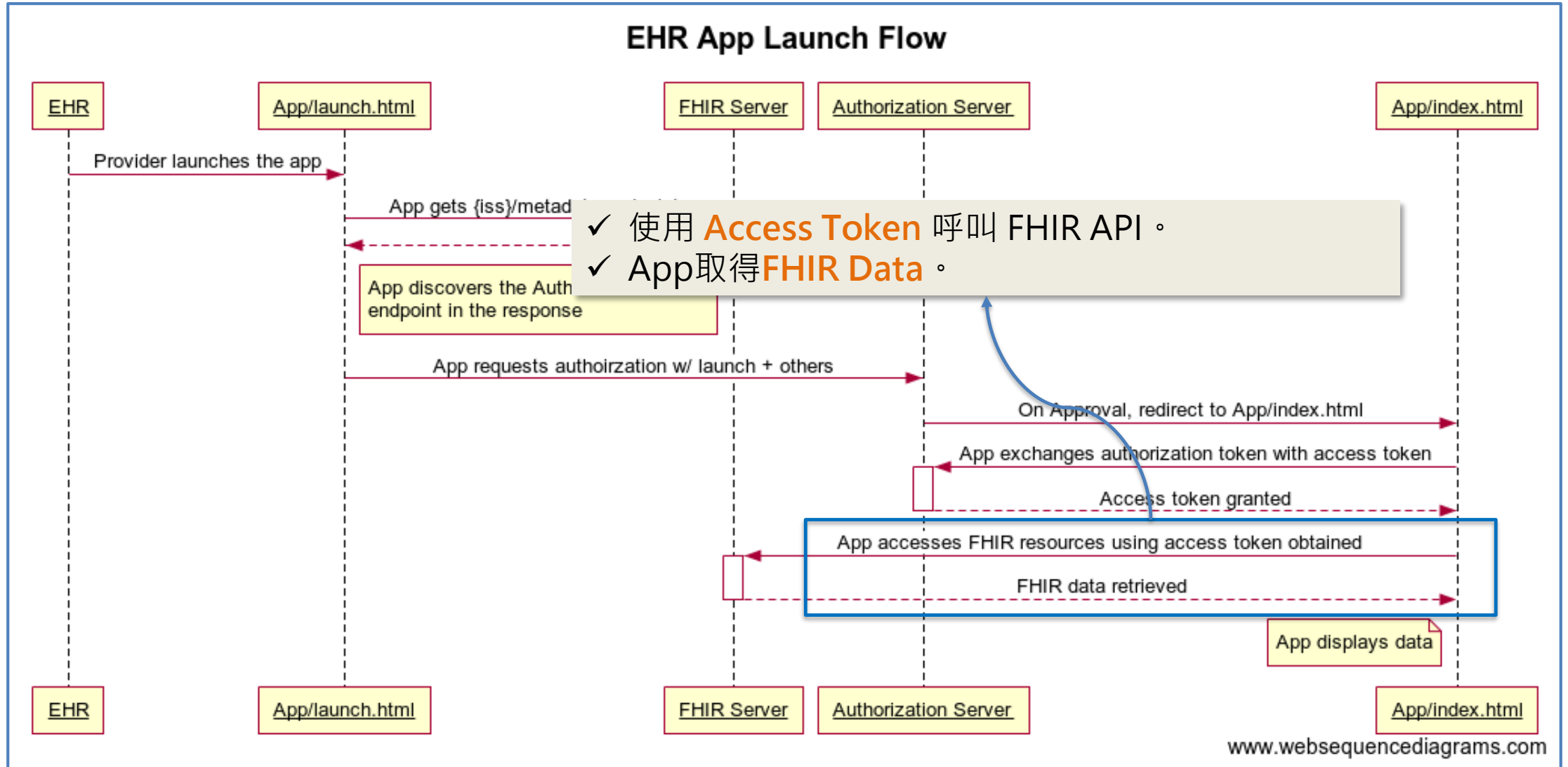
SMART on FHIR – 認證流程(II)



SMART on FHIR – 認證流程(III)



SMART on FHIR – 認證流程(IV)



支援 SMART App Launch 的應用程式庫



client-js
用戶端和伺服器端



client-py
伺服器端

SMART on FHIR JavaScript Library

JavaScript client for FHIR

[View on GitHub](#)

SMART on FHIR JavaScript Library

This is a JavaScript library for connecting SMART apps to FHIR servers. It works both in browsers (IE 10+) and on the server (Node 18+).

This is the documentation for version 2+. If you want to migrate from older versions, please check out [what's new in v2](#) and [migration instructions](#).

SMART on FHIR Python Client 4.3.1

SMART on FHIR Python Client

- ▶ SMART FHIR Client
- ▶ Credits
- ▶ How To Contribute
- ▶ Namespaces
- ▶ Classes
- ▶ Files

SMART FHIR Client

This is *fhirclient*, a flexible Python client for FHIR servers supporting the [SMART on FHIR](#) protocol.

Client versioning is not identical to FHIR versioning. The main branch is usually on the latest version bugfix releases. The develop branch should be on recent freezes, and the feature/latest-ci continuous integration builds.

Version	FHIR	
4.2.0	4.0.1	(R4)
4.0.0	4.0.0	(R4)
3.0.0	3.0.0	(STU-3)
x.x	1.8.0	(STU-3 Ballot, Jan 2017)
x.x	1.6.0	(STU-3 Ballot, Sep 2016)
1.0.3	1.0.2	(DSTU 2)
1.0	1.0.1	(DSTU 2)

建置 JavaScript 應用程式

SMART 應用程式在 EHR 中運行，須支援 EHR 啟動流程。執行邏輯分成兩部份：

- [launch.htm](#) - 由 EHR 呼叫以啟動授權流程。
- [index.html](#) - 授權成功後，EHR 會將使用者重新定向到實際應用程式執行的地方。



用戶端：

```
<!-- Launch.html -->
<script src="./node_module/fhirclient/build/fhir-client.js"></script> <script>
FHIR.oauth2.authorize({
  "client_id": "my_web_app",
  "scope": "patient/*.read" });
</script>
```

由EHR提供

設定所需使用的FHIR Resources

```
<!-- index.html -->
<script src="./node_module/fhirclient/build/fhir-client.js"></script>
<script>
FHIR.oauth2.ready()
  .then(client => client.request("Patient"))
  .then(console.log)
  .catch(console.error);
</script>
```

建置 JavaScript 應用程式

`client.request(requestOptions, fhirOptions) :`

`npm i fhirclient`

[Ref: 請求範例](#)



無須驗證的情況下擷取FHIR Resources :

```
const client = new FHIR.client("https://r3.smarthealthit.org");
client.request("Patient/2e27c71e-30c8-4ceb-8c1c-5641e066c0a4");
```



The screenshot shows a CodePen live editor with the following content:

- HTML Tab:** Contains a single `output` element.
- JS Tab:** Contains the following JavaScript code:

```
function display(data) {
  const output = document.getElementById("output");
  output.innerText = data instanceof Error ?
    String(data) :
    JSON.stringify(data, null, 4);
}

const client = new FHIR.client("https://r3.smarthealthit.org");
client.request("Patient/2e27c71e-30c8-4ceb-8c1c-5641e066c0a4")
  .then(display)
  .catch(display);
```
- Result Tab:** Displays the JSON response from the FHIR server:

```
{
  "resourceType": "Patient",
  "id": "2e27c71e-30c8-4ceb-8c1c-5641e066c0a4",
  "meta": {
    "versionId": "15",
    "lastUpdated": "2025-09-30T02:47:48.532-04:00",
    "profile": [
      "http://standardhealthrecord.org/fhir/StructureDefinition/shr-der"
    ],
    "tag": [
      {
        "system": "https://smarthealthit.org/tags",
        "code": "synthea-7-2017"
      }
    ]
  }
}
```
- Buttons:** Includes a "LIVE" button, a "CodePen Embed" button, and a "Rerun" button.

建置 JavaScript 應用程式

驗證過的用戶端擷取FHIR Resources：

```
const client = new FHIR.client({
  serverUrl: "https://r3.smarthealthit.org",
  tokenResponse: {
    patient: "2e27c71e-30c8-4ceb-8c1c-5641e066c0a4"
  }
});

client.request(`Patient/${client.patient.id}`);
```

The screenshot shows a live JavaScript editor interface with three tabs: HTML, JS, and Result. The JS tab is active, displaying the following code:

```
function display(data) {
  const output = document.getElementById("output");
  output.innerText = data instanceof Error ?
    String(data) :
    JSON.stringify(data, null, 4);
}

const client = new FHIR.client({
  serverUrl: "https://r3.smarthealthit.org",
  tokenResponse: {
    patient: "2e27c71e-30c8-4ceb-8c1c-5641e066c0a4"
  }
});

client.request(`Patient/${client.patient.id}`)
  .then(display)
  .catch(display);
```

The Result tab shows the JSON response from the FHIR server:

```
{
  "resourceType": "Patient",
  "id": "2e27c71e-30c8-4ceb-8c1c-5641e066c0a4",
  "meta": {
    "versionId": "15",
    "lastUpdated": "2025-09-30T02:47:48.532-04:00",
    "profile": [
      "http://standardhealthrecord.org/fhir/StructureDefinition/shr-der"
    ],
    "tag": [
      {
        "system": "https://smarthealthit.org/tags",
        "code": "synthea-7-2017"
      }
    ]
  },
  "text": {
    "status": "generated",
    "div": "<div xmlns=\\"http://www.w3.org/1999/xhtml\\">Generated by <a href=\\"https://smarthealthit.org\\">SMART on FHIR</a>"
  }
}
```

At the bottom of the editor, there are zoom controls (1x, 0.5x, 0.25x) and a 'Rerun' button.

建置 JavaScript 應用程式

正確啟動並經過身份驗證的用戶端將擁有一個 user 屬性；藉由要求 openid 和 fhirUser取得：

```
const id_token = "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9." +
  "eyJwcm9maWxlIjoiaUJhY3RpdGlubmVyL3NtYXJ0LVByYWN0aXRpb2" +
  "5lci03MjA4MDQxNiIsImZoaXJvc2VyIjoiaUJhY3RpdGlubmVyL3Nt" +
  "YXJ0LVByYWN0aXRpb25lci03MjA4MDQxNiIsInN1YiI6IjM2YTEwYm" +
  "M0ZDJhNmM1OGI0YWZkYWZhZjZjYmFhYmFhYmFhYmFhYmFhYmFhYm" +
  "YTgwMjg0MmFkNWZhZGQxNzgiLCJpc3MiOiJodHRwOi8vbGF1bmNoLn" +
  "NtYXJ0aGVhbnRoXQub3JnIiwiaWF0IjoxNTU5MzkyMjk1LCJleHAi" +
  "OjE1NTkzOTU0TV9.eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9." +
  "ZgwwYasxVU8x3gJiX3jqONttqPhkh7418EFssCKnnaBlUDwsbhp7xd" +
  "WN4o1L1NvH4bp_R_zJ25F1s6jLmNm2Qp9LqU133PEdcRIqQPgBMyZB" +
  "WUTyxQ9ihKY1RAjlztAULQ3wKea-rfe0BXJZeUJBsQPzYCnbKY1dON" +
  "_NRd8N9pTImqf41MpIbEe7YEOHuirIb6HBpurhAHjTLDv1IuHpEA0x" +
  "pmtxVVHivf-FYXzTFmn4cGe2PsNJfBl8R_zow2n6qaSANDvSxJDE4D" +
  "UgIJ6H18wiSJH6Plf_bapccAwxbx-zZCw";

const client = new FHIR.client({
  serverUrl: "https://r3.smarthealthit.org",
  tokenResponse: { id_token }
});

client.request(client.user.fhirUser);
```

建置 JavaScript 應用程式

顯示某位患者的用藥：

```
const client = new FHIR.client("https://r3.smarthealthit.org");
const getPath = client.getPath;

client.request(`/MedicationRequest?patient=smart-1642068`, {
  resolveReferences: "medicationReference"
}).then(data => data.entry.map(item => getMedicationName(
  getPath(item, "resource.medicationCodeableConcept.coding") ||
  getPath(item, "resource.medicationReference.code.coding")
))));
```

HTML

Babel

Result

EDIT ON
CODEPEN

```
2 const client = new FHIR.client("https://r3.smarthealthit.org");
3 const getPath = client.getPath;
4
5 function display(data) {
6   const output = document.getElementById("output");
7   output.innerText = data instanceof Error ?
8     String(data) :
9     JSON.stringify(data, null, 4);
10 }
11
12 function getMedicationName(medCodings = []) {
13   var coding = medCodings.find(c => c.system === rxnorm);
14   return coding && coding.display || "Unnamed Medication(TM)";
15 }
16
17 client.request(`/MedicationRequest?patient=smart-1642068`, {
18   resolveReferences: "medicationReference"
19 }).then(data => data.entry.map(item => getMedicationName(
20   getPath(item, "resource.medicationCodeableConcept.coding")
21   ||
22   getPath(item, "resource.medicationReference.code.coding")
23   )).then(display, display);
```

```
[
  "Lisinopril 20 MG Oral Tablet",
  "120 ACTUAT mometasone furoate 0.05 MG/ACTUAT Nasal Inhaler [Nasonex]",
  "Ranitidine 150 MG Oral Tablet",
  "Hydrochlorothiazide 12.5 MG / irbesartan 150 MG Oral Tablet [Avalide 150/1",
  "Hydrochlorothiazide 12.5 MG / Lisinopril 20 MG Oral Tablet",
  "pantoprazole 40 MG Enteric Coated Tablet [Protonix]",
  "Diclofenac 75 MG / Misoprostol 0.2 MG Enteric Coated Tablet [Arthrotec 75",
  "Alendronate 70 MG Oral Tablet",
  "Thyroxine 0.112 MG Oral Tablet [Levoxyl]",
  "pantoprazole 40 MG Enteric Coated Tablet",
  "Simvastatin 20 MG Oral Tablet"
]
```

Resources

1×

0.5×

0.25×

Rerun

建置 Python 應用程式

無須驗證的情況下擷取FHIR Resources：

pip install fhirclient

[Ref : SMART on FHIR Python Client: SMART FHIR Client](#)



```
from fhirclient import client
from fhirclient.models.patient import Patient

settings = {
    'app_id': 'my_web_app',
    'api_base': 'https://r4.smarthealthit.org'
}
smart = client.FHIRClient(settings=settings)

patient = Patient.read('2cda5aad-e409-4070-9a15-e1c35c46ed5a', smart.server)
print(patient.birthDate.isostring)
# '1992-07-03'
print(smart.human_name(patient.name[0]))
# 'Mr. Geoffrey Abbott'
```

建置 Python 應用程式

從伺服器中搜尋Resource：

```
from fhirclient import client
from fhirclient.models.encounter import Encounter
from fhirclient.models.procedure import Procedure

settings = {
    'app_id': 'my_web_app',
    'api_base': 'https://r4.smarthealthit.org'
}
smart = client.FHIRClient(settings=settings)

search = Encounter.where(struct={'subject': '2cda5aad-e409-4070-9a15-e1c35c46ed5a', 'status':
'finished'})
print({res.type[0].text for res in search.perform_resources_iter(smart.server)})
# {'Encounter for symptom', 'Encounter for check up (procedure)'}
```

```
# to include the resources referred to by the encounter via `subject` in the results
search = search.include('subject')
print({res.resource_type for res in search.perform_resources_iter(smart.server)})
# {'Encounter', 'Patient'}

# to include the Procedure resources which refer to the encounter via `encounter`
search = search.include('encounter', Procedure, reverse=True)
print({res.resource_type for res in search.perform_resources_iter(smart.server)})
# {'Encounter', 'Patient', 'Procedure'}

# to get the raw Bundles instead of resources only, you can use:
bundles = search.perform_iter(smart.server)
print({entry.resource.resource_type for bundle in bundles for entry in bundle.entry})
# {'Encounter', 'Patient', 'Procedure'}
```

建置 Python 應用程式



Flask App : https://github.com/smart-on-fhir/client-py/blob/main/demos/flask/flask_app.py

應用程式將啟動一個 Web 伺服器，監聽 localhost:8000，並提示使用者登入沙盒伺服器並選擇患者。然後，它會檢索所選患者的人口統計和藥物處方，並將其列在一個簡單的 HTML 頁面上。

```
git clone https://github.com/smart-on-fhir/client-py.git
cd client-py/demos/flask
python3 -m venv env
. env/bin/activate
pip install -r requirements.txt
# Edit flask_app.py and put your own server's URL as api_base.
./flask_app.py
```

開始你的第一個SMART App



建立可發布網站的環境(Web-Based App)

GitHub Pages 是 GitHub 提供的靜態網站託管服務，可以直接把你的程式碼或 HTML 網頁部署成網站。

Step 1：建立 Repository

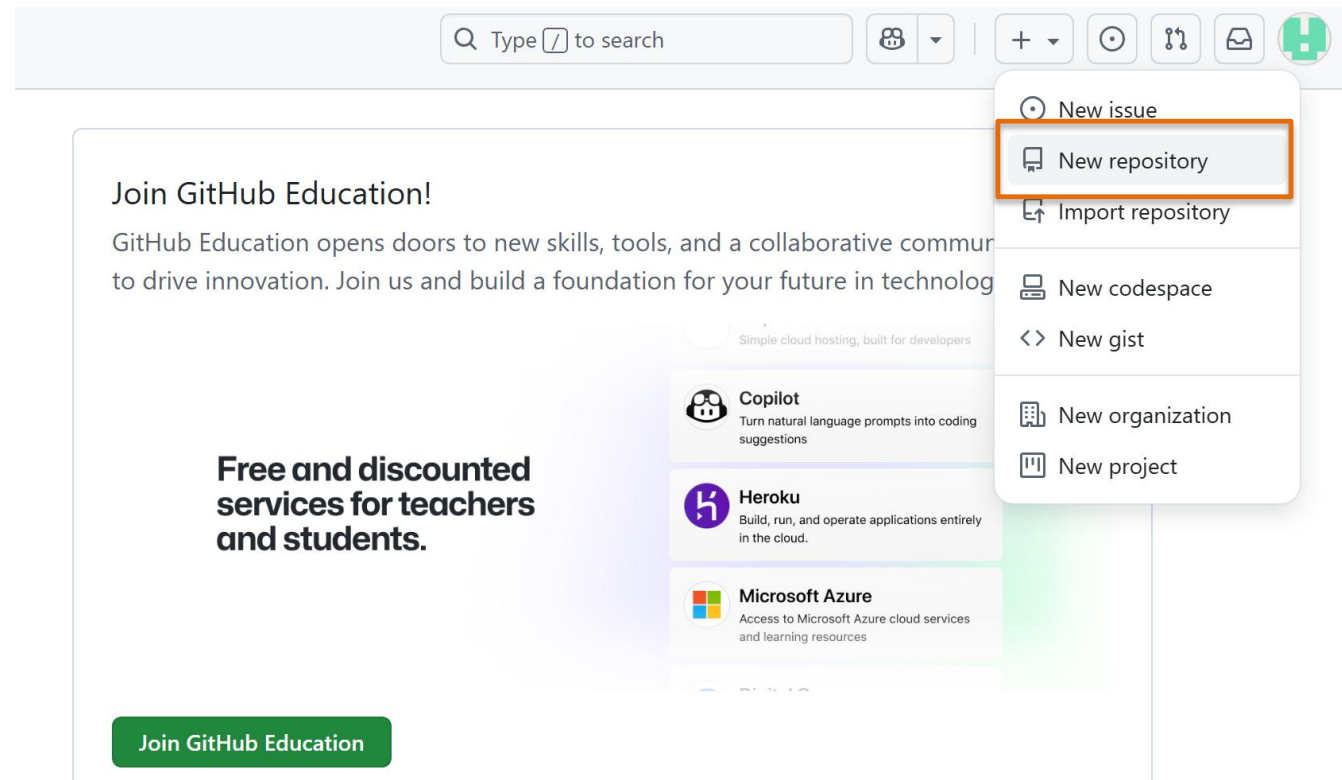
1. 登入 [GitHub](#)

2. 點選 **New Repository**

3. 名稱輸入：

- 若是個人網站 → username.github.io
- 若是專案網站 → 任意名稱 (例如 myproject)

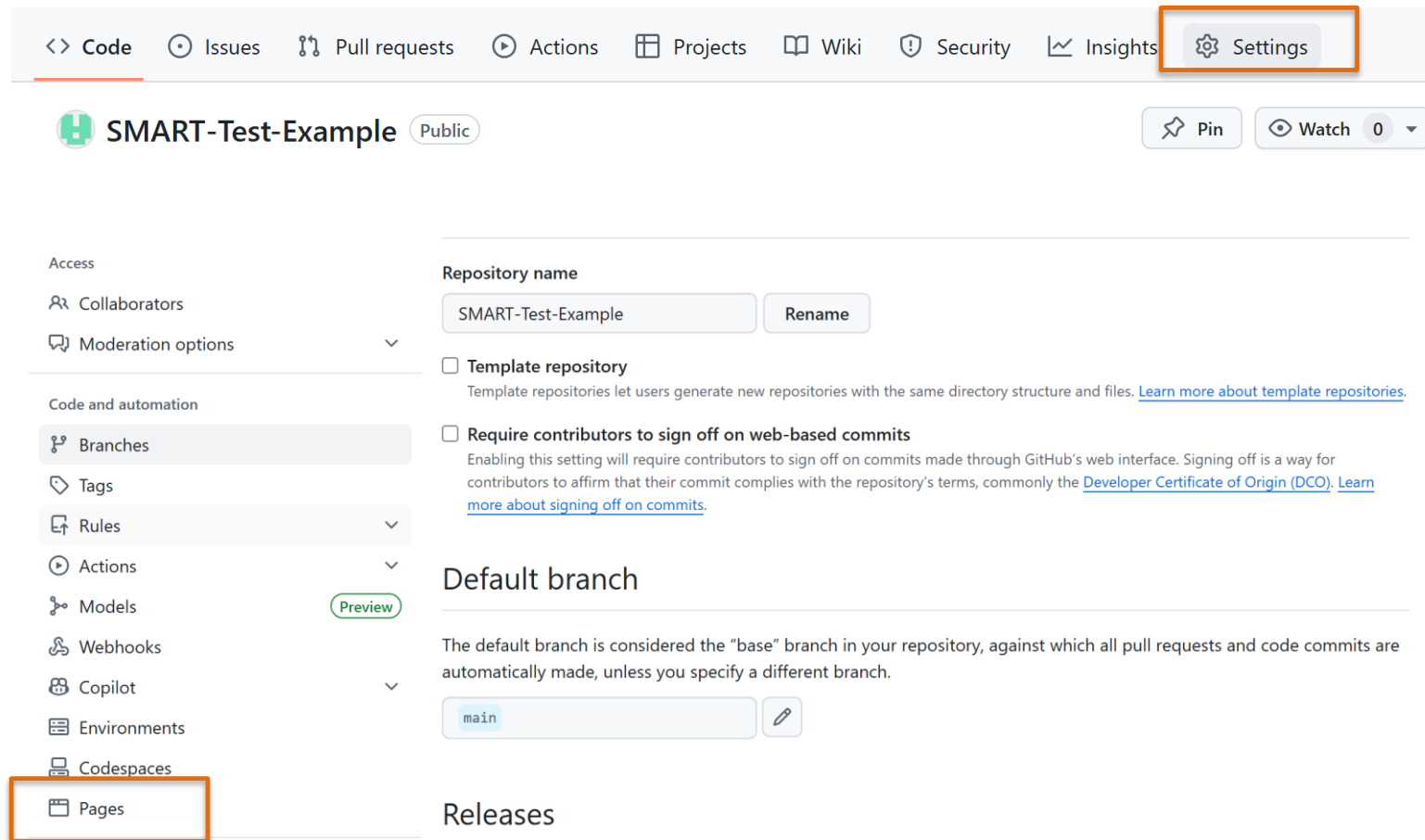
4. 建立完成後可選「Add a README」方便初始化。



建立可發布網站的環境(Web-Based App)

Step 2 : 啟用 GitHub Pages

- 1.到該 repo 頁面，點選上方的 **Settings**
- 2.左邊選單 → **Pages**
- 3.在「Source」選擇：
 - Branch : main
 - Folder : / (root)
- 4.按 **Save**



建立可發布網站的環境(Web-Based App)

GitHub Pages

[GitHub Pages](#) is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <https://chienju315.github.io/SMART-Test-Example/>
Last [deployed](#) by  [chienju315](#) 1 hour ago

[Visit site](#)

[Unpublish site](#)

Build and deployment

Source

Deploy from a branch ▼

Branch

Your GitHub Pages site is currently being built from the /docs folder in the main branch. [Learn more about configuring the publishing source for your site.](#)

 main ▼

 /docs ▼

Save

建立可發布網站的環境(Web-Based App)

SMART-Test-Example / docs /

Add file ▾

chienju315 Update index.html ✓ 74abcc5 · 1 hour ago History

Name	Last commit message	Last commit date
..		
fhir-client-v0.1.12.js	Add files via upload	1 hour ago
fhir-client-v2.6.0.js	Add files via upload	1 hour ago
index.html	Update index.html	1 hour ago
launch.html	Update launch.html	1 hour ago

Launch.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Launch My APP</title>
    <script src="https://cdn.jsdelivr.net/npm/fhirclient/build/fhir-client.js"></script>
  </head>
  <body>
    <script>
      FHIR.oauth2.authorize({

        // The client_id that you should have obtained after registering a client at the EHR.
        clientId: "my_web_app",

        // The scopes that you request from the EHR. In this case we want to:
        scope: "launch openid fhirUser patient/*.read",

        // Typically, if your redirectUri points to the root of the current directory
        // (where the launchUri is), you can omit this option because the default value is ".".
        redirectUri: "index.html"
      });
    </script>
  </body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Example SMART App</title>
    <script src="https://cdn.jsdelivr.net/npm/fhirclient/build/fhir-client.js"></script>
    <style>
      #patient, #meds {
        font-family: Monaco, monospace;
        white-space: pre;
        font-size: 13px;
        height: 30vh;
        overflow: scroll;
        border: 1px solid #CCC;
      }
    </style>
  </head>
  <body>
    <h4>Current Patient</h4>
    <div id="patient">Loading...</div>
    <br/>
    <h4>Medications</h4>
    <div id="meds">Loading...</div>
```

```
<script type="text/javascript">
  FHIR.oauth2.ready().then(function(client) {
    // Render the current patient (or any error)
    client.patient.read().then(
      function(pt) {
        document.getElementById("patient").innerText = JSON.stringify(pt, null, 4);
      },
      function(error) {
        document.getElementById("patient").innerText = error.stack;
      }
    );
    // Get MedicationRequests for the selected patient
    client.request("/MedicationRequest?patient=" + client.patient.id, {
      resolveReferences: [ "medicationReference" ],
      graph: true
    })
    // Reject if no MedicationRequests are found
    .then(function(data) {
      if (!data.entry || !data.entry.length) {
        throw new Error("No medications found for the selected patient");
      }
      return data.entry;
    })
  })
}
```

```
// Render the current patient's medications (or any error)
    .then(
      function(meds) {
        document.getElementById("meds").innerText = JSON.stringify(meds, null, 4);
      },
      function(error) {
        document.getElementById("meds").innerText = error.stack;
      }
    );

  }).catch(console.error);
</script>
</body>
</html>
```

SMART on FHIR：可替代的醫療應用程式和可重複使用的技術

- SMART App測試



啟動你的SMART App

 SMART

首頁 SMART Marketplace Sand box SMART 提案 李 ▾

SAND-BOX

請先使用 Postman 或其他工具，將測試資料匯入下方 FHIR Server URL，完成後再輸入 Launch URL進行應用程式測試。

FHIR Server URL：<https://emr-smart.appx.com.tw/v/r4/fhir> 

請輸入網址

送出

證 

啟動你的SMART App

Practitioner Login

Practitioner

Dr. Albertine Orn



Password

.....

This login is for demonstration purposes only. ANY password will be accepted.

Login

啟動你的SMART App

Select Patient

☐ Search

Name ☐	Gender	Age
☐ Mr. Barney Abbott	M	25 years deceased
☐ Mr. Geoffrey Abbott	M	33 years
☐ Mr. Omar Abernathy	M	25 years
☐ Mrs. María del Carmen Abreu	F	40 years
☐ Theo Abshire	F	9 years
☐ Mr. Christopher Abshire	M	97 years deceased
☐ Ms. Mariana Acuña	F	37 years
☐ Ms. Charita Adams	F	24 years
☐ Mrs. Maricarmen Alarcón	F	39 years
☐ Mrs. Lourdes Alcántar	F	77 years deceased

啟動你的SMART App

Current Patient

```
{
  "resourceType": "Patient",
  "id": "9e691882-981e-4dd3-af5e-be44d315d715",
  "meta": {
    "versionId": "75",
    "lastUpdated": "2025-06-06T03:08:36.699-04:00",
    "tag": [
      {
        "system": "https://smarthealthit.org/tags",
        "code": "synthea-5-2019"
```

Medications

```
[
  {
    "fullUrl": "https://emr-smart.appx.com.tw/v/r4/fhir/MedicationRequest/3c0e21c5-654c-4af7-891e-7bc129c4a35a",
    "resource": {
      "resourceType": "MedicationRequest",
      "id": "3c0e21c5-654c-4af7-891e-7bc129c4a35a",
      "meta": {
        "versionId": "73",
        "lastUpdated": "2025-09-28T03:12:34.980-04:00",
        "tag": [
```

Test Environments

- [SMART App Launcher \(no registration required\)](#): Developer tool for SMART apps
 - [Docker Container](#): For local installation or experiments
 - [R4 open endpoint](#) (see also [R2](#), [R3](#))
- [SMART Bulk Data Server \(no registration required\)](#): Developer tool for Bulk Data clients
 - [Source Code](#)
- [Logica Health Sandbox](#): Manage your own sandbox server and users over time

Vendor Sandboxes

- [Allscripts](#)
- [athenahealth - athenaOne](#)
- [Cerner - Provider and Patient Facing Apps](#)
- [Epic Provider Facing Apps](#)
- [Epic Patient Facing Apps](#)
- [Intersystems](#)
- [Meditech](#)

SMART Launcher

App Launch Options

Client Registration & Validation

Launch Type

Provider EHR Launch

Practitioner opens the app from within an EHR

FHIR Version

R4

Select what FHIR version your app should work with

Simulated Error

None

Force the server to throw certain type of error (useful for manual testing).

Misc. Options

☐ Simulate launch within the EHR UI (launch within an iFrame)

Patient(s)

Patient ID(s)



Simulates the active patient in EHR when app is launched. If no Patient ID is entered or if multiple comma delimited IDs are specified, a patient picker will be displayed as part of the launch flow.

Provider(s)

Provider ID(s)

Simulates user who is launching the app. If no provider is selected, or if multiple comma delimited Practitioner IDs are specified, a login screen will be displayed as part of the launch flow.

Encounter

Select the most recent encounter if available

How to select the current Encounter

 App's Launch URL

SMART App Launch Test Kit

SMART App Launch IG : <http://hl7.org/fhir/smart-app-launch/index.html>

本實施指南描述了一組基於 OAuth 2.0 的基本模式，供客戶端應用程式授權、身份驗證以及與基於 FHIR 的資料系統整合。

SMART 定義兩種用戶端授權模式

- ✓ 應用程式啟動授權：與應用程式共享當前選擇的患者資料
 - EX. Apple Health
- ✓ 後端服務授權：允許後端服務與 EHR 連動，無任何使用者介入。

SMART 定義兩種用戶端身份驗證模式

- ✓ 非對稱（「私鑰 JWT」）身份驗證：SMART 的首選身份驗證方法，因為它避免透過線路發送共用金鑰。
- ✓ 對稱（「客戶端秘密」）身份驗證：使用客戶端和伺服器之間預先共用的金鑰對客戶端進行身份驗證。

Epic 中建置和啟動 SMART on FHIR 應用程式



FHIR API Specifications **Build Apps** Documentation ▾ Jump To ▾

My Apps

Sign Up to Access

Epic on FHIR is a free resource for developers who create apps for use by patients and healthcare organizations.



Testing Sandbox
Test APIs against example data



Client Registration
Software registration and client identifier management



Documentation
Additional developer support documentation



FHIR API Specifications **Build Apps** Documentation ▾ Connection Hub

Create Test Ready

Create an App

Please enter some basic information about the app you're creating. You'll also be able to make changes to this information later.

Application Name
mytestapp1

Application Audience

☐ Patients ☒ Clinicians or Administrative Users ☐ Backend Systems

Incoming APIs

Search available service list

Available		Selected
Account.Read (Premium Billing) (R4)		Patient.Read (R4)
Account.Search (Premium Billing) (R4)		Patient.Search (R4)
AdverseEvent.Read (R4)		MedicationRequest.Read (Orders) (R4)
AdverseEvent.Search (R4)		MedicationRequest.Search (Orders) (R4)
AllergyIntolerance.Create (R4)		AllergyIntolerance.Read (R4)
AllergyIntolerance.Create (STU3)		AllergyIntolerance.Search (R4)
AllergyIntolerance.Read (DSTU2)		
AllergyIntolerance.Read (STU3)		
AllergyIntolerance.Search (DSTU2)		
AllergyIntolerance.Search (STU3)		

Redirect URI

https:// www.fhir.epic.com

Please enter a valid redirect URI.

Add Another URI

Does this app use CDS Hooks?
☐ Uses CDS Hooks

Is this app a confidential client?
☐ Is Confidential Client

Save Cancel

SMART on FHIR：可替代的醫療應用程式和可重複使用的技術

- SMART App提案及上架流程



SMART App提案及上架流程



[首頁](#) [SMART Marketplace](#) [Sand box](#) [SMART 提案](#)

[登入](#) [註冊](#)

歡迎來到
SMART

SMART App提案及上架流程



歡迎來到
SMART

SMART

註冊

帳號與密碼

帳號*

請輸入8-12碼帳號，僅可使用英文與數字

密碼*

密碼確認*

請再次輸入密碼

會員基本資訊

會員姓名*

請輸入您的姓名

會員單位*

請輸入您的單位

會員E-mail*

首頁

SMART Marketplace

Sand box

SMART 提案

登入

註冊

SMART App提案及上架流程



[首頁](#) [SMART Marketplace](#) [Sand box](#) [SMART 提案](#) [登入](#) [註冊](#)

歡迎來到
SMART



登入

帳號*

kevinli0315

密碼*

.....



驗證碼*

請輸入右方驗證碼



[忘記密碼](#)

[登入](#)

[還不是會員？立即加入](#)

SAND-BOX

Sandbox 是專為提案者打造之符合 SMART on FHIR
標準的獨立測試環境。

在這裡，您可自由測試或執行任何程式碼，為您的提
案提供安全、零風險的驗證空間。

軟體檢核驗證



SMART App提案及上架流程



授權標註與免責聲明

本網站所提供之 Sandbox 測試環境使用以下開源套件

1. [SMART-ON-FHIR/SMART-LAUNCHER-V2](#)

- Copyright © Copyright 2018 Boston Children's Hospital
- License: [Apache-2.0 license](#)

2. [HAPIFHIR/HAPI-FHIR](#)

- Copyright © Copyright 2015 Smile CDR Inc.
- License: [Apache-2.0 license](#)

3. [SMART-ON-FHIR/GENERATED-SAMPLE-DATA](#)

- Copyright © Copyright 2018 Boston Children's Hospital
- License: [Apache-2.0 license](#)

修改說明：

1. 為確保對CapabilityStatement的向下相容性，執行團隊修改部分SMART-LAUNCHER-V2程式邏輯，以確保其相容性。
2. 預設測試資料集使用SMART-ON-FHIR/GENERATED-SAMPLE-DATA/R4/Synthea，執行團隊無法保證其資料完整性，啟動測試前，先使用API測試工具(Postman..等)匯入符合APP使用情境之測試資料。

免責聲明：

1. 本網站使用之部分軟體來自開源社群，依各自授權條款提供。
2. 開源工具原作者對本系統的功能與服務不承擔任何責任。
3. 本網站所提供的沙盒測試僅提供使用者驗證其SMART App程式檔案是否能正常執行，不對其宣稱功能與服務進行擔保，亦不承擔SMART App於其他EMR環境中是否能正常運行之責任。
4. 若您的SMART App引用外部函式庫或開源套件，請於您的程式中清楚標註引用來源。
5. 本網站為公開測試環境，請勿將醫院真實病歷資料上傳測試。

我已了解



衛生福利部
Ministry of Health and Welfare

驗證



[免責聲明 \(點此查看\)](#)

台北市南港區忠孝東路6段488號

02-890-6666

02-890-6000

standard@itri.org.tw

SMART提案

[首頁](#) > [SMART 提案](#)

SMART系統說明

SMART on FHIR 是一種結合 HL7FHIR 標準與 OAuth2、OpenIDConnect 等安全協定的醫療資料互通架構應用程式，在於促進醫療應用的開發與部署，提升醫療資訊系統的整合性與靈活性，避免造成資料孤島。這種架構提升了醫療系統的資料互通性與共享效率，同時確保隱私保護。

SMART App提案及上架流程



STEP1 Sand box軟體測試

- 確認規則能正常運行並取得測試結果。

[前往SAND-BOX測試](#)



STEP2 測試成功 → 填寫「提案表」

- 輸入規則名稱、應用說明、使用的FHIR resources / 術語值集等。



STEP3 審查確認

資訊處將根據應用程式資訊、聯絡人資訊、系統展示等項目進行審查。

應用程式資訊

應用程式類型*

請選擇應用程式類型



應用程式名稱*

請輸入應用程式名稱

應用程式網址*

請輸入應用程式網站網址

機構名稱*

請輸入機構名稱

分類*

最多可選3個分類



專科別*

最多可選3個分類



是否進行軟體收費？*

請選擇是否進行收費



應用程式簡短描述*

請輸入應用程式簡短描述，上限300字

SMART App提案及上架流程



應用程式描述*

請輸入140字以上之應用程式描述，如果您的應用程式涉及病人資訊，請於下方欄位提供隱私權政策（Model Privacy Notice）連結

隱私權政策

請輸入隱私權政策（Model Privacy Notice）連結

程式畫面截圖*

請上傳至少一張螢幕截圖(檔案格式限JPG或PNG，單一檔案大小需小於1MB)

上傳檔案

0.00MB / 1MB

應用程式Icon*

請上傳應用程式Icon(尺寸為256px*256px)

上傳檔案

0.00MB / 1MB

影片連結

輸入YouTube或Vimeo的網址以嵌入影片

適用對象*

請選擇應用程式主要使用族群 ▾

作業系統-選擇您的應用程式可運行的作業系統，若為Android或iOS，請提供應用商店連結。*

☐ Android

☐ iOS

SMART App提案及上架流程



- ☐ Web
- ☐ Mac
- ☐ Windows
- ☐ Linux

FHIR相容性*

選擇您的應用程式可使用的FHIR版本



此SMART APP使用到哪些詞彙、術語值集等之標準？（例如，SNOMED CT、LOINC、RxNorm、ICD-10等）*

請說明所使用之值集內容

使用此SMART APP的實施指南（IG）或設定檔（例如，基於FHIR R4 的 HL7® FHIR® TW Core IG v0.3.2）

請說明所使用到的FHIR Profiles

請說明SMART APP的維護及更新計畫

請描述您的維護及更新計畫

聯絡人資訊

開發者姓名*

請輸入開發者姓名

開發者電子郵件*



行政聯絡人*

請輸入電子郵件方便聯絡行政事宜(此電子郵件地址不會公開)

業務聯絡人

請輸入電子郵件供我們回報使用者在瀏覽應用程式時的問題(此電子郵件地址不會公開)

系統展示

示範類型

請從以下清單中選擇一種示範類型



啟動連結(Launch URL)

請輸入應用程式的OAuth啟動連結

重新導向連結(Redirect URL)

請輸入應用程式的OAuth重新導向連結

放棄提案

提案

我的提案

個人資訊

我的提案

登出

[首頁](#) > [我的提案](#)

全部(0)

照護協調(0)

臨床研究(0)

資料視覺化(0)

疾病管理(0)

基因體學(0)

藥物管理(0)

全部 0

狀態 | 全部 ▲ 排序 | 更新日期 (新 → 舊) ▲

Exercise 1 : Building a JavaScript App [SMART -- Tutorials – JavaScript](#)



Exercise 2 : [SMART on FHIR app tutorial](#)



THANK YOU

