
Firely .NET SDK 操作手冊

| | |
|--------|------------|
| 發行版本 | 1.0 |
| 最後修訂日期 | 2023-11-10 |

修改歷程

| 版本 | 說明 | 修改者 | 修改時間 |
|-----|----|-----|------------|
| 1.0 | 初稿 | 彭子芳 | 2022-11-10 |

目錄

| | |
|------------------------------------|----|
| 1. 文件概述..... | 4 |
| 1.1. Firely .NET SDK 簡介 | 4 |
| 1.2. 手冊簡介..... | 5 |
| 2. 基本 Firely .Net SDK 操作介紹 | 6 |
| 2.1. 基本設定..... | 6 |
| 2.1.1. 使用 Polyglot Notebooks..... | 6 |
| 2.1.2. Import Libraries | 6 |
| 2.1.3. 連接 FHIR Server | 6 |
| 2.2. CRUD 操作 | 7 |
| 2.2.1. 取得單筆資料(使用 id)..... | 7 |
| 2.2.2. 搜尋(Search) | 8 |
| 2.2.3. 排序(Order)..... | 9 |
| 2.2.4. 新增一筆 Patient (Create) | 10 |
| 2.2.5. 更新 (Update) | 12 |
| 2.2.6. 刪除 (Delete) | 13 |
| 3. 情境應用..... | 14 |
| 3.1. 情境說明..... | 14 |
| 3.2. Track1 病患基本資料 | 14 |
| 3.2.1. 首次建檔病患資料..... | 15 |
| 3.2.2. 更新該筆病患資料..... | 15 |
| 3.2.3. 刪除該筆病患資料..... | 15 |
| 3.3. Track2 生理量測數據 | 16 |
| 3.3.1. 設定 FHIR Server | 16 |
| 3.3.2. 新增一筆體重紀錄..... | 16 |
| 3.3.3. 搜尋該筆體重資料..... | 17 |
| 3.3.4. 更新體重數據..... | 17 |
| 3.3.5. 刪除該筆體重資料..... | 18 |

1. 文件概述

本文件是Firely .NET SDK 操作手冊，介紹Firely .NET SDK基礎操作與基本範例。

1.1. Firely .NET SDK 簡介

依據 [Firely 官網](#)說明，Firely .NET SDK 提供了一組工具和程式庫，讓開發人員可以更輕鬆地在 .NET 應用程式中使用 FHIR 資源，Firely .NET SDK 是開源的，可以從 GitHub 儲存庫下載，並且定期更新以跟上 FHIR 標準的最新發展。Firely .NET SDK 的目的是說明開發人員構建和維護與 FHIR 資料交互的應用程式，從而促進醫療資訊的交流和共用。

以下是 Firely .NET SDK 的一些主要特點和功能：

- ✓ RESTful API: 用於使用 FHIR RESTful API 與 FHIR Server 互動的客戶端程式庫，使用標準 HTTP 方法輕鬆檢索、建立、更新和刪除 FHIR 資源。
- ✓ FHIR 資料模型(FHIR Data Model): 由於代表不同類型 FHIR 資源的類別和接口，可以更輕鬆地在 .NET 應用程式中使用 FHIR 資源。
- ✓ 序列化(Serialization): 用於以不同格式（包括 JSON 和 XML）序列化和反序列化 FHIR 資源的工具
- ✓ Profiling: 使用 FHIR 設定文件，包括使用 SnapshotGenerator 建立設定檔的能力。定義自訂設定文件，並使用它們根據自訂約束驗證資源。
- ✓ 支援多個 FHIR 版本: 使用任何版本的 FHIR 標準，從最早的版本到最新、最前沿的版本，並確保您的應用程式將隨著標準的不斷發展而繼續無縫工作。
- ✓ FHIRPath: 對 FHIR 資源執行複雜的搜尋和操作，以檢索特定的資料元素或將其轉換為新的表示形式。

Firely .NET SDK 是一個強大的工具，可以用於構建各種醫療資訊系統，包括電子健康記錄（EHR）、醫療保險索賠處理、健康資訊交換和研究應用程式等。它

有助於實現醫療資訊的互通性，促進醫療資料的共用和協作，從而提高醫療保健的品質和效率。

1.2. 手冊簡介

本操作手冊將於第二章節先介紹基本的 Firely .Net SDK 操作介紹，包含基本設定、連線 FHIR Server 方法，以及 CRUD 操作。

第三章將以 2023 臺灣醫學資訊互通聯測松之 Track1 和 Track2，分別為病人基本資料和生理量測數據(以體重為例)賽道以實作 Firely .NET SDK。

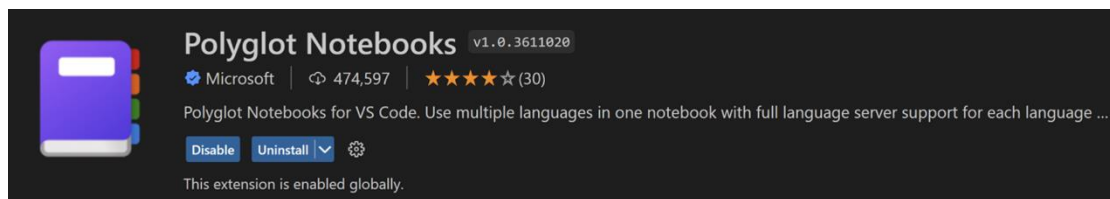
2. 基本 Firely .Net SDK 操作介紹

2.1. 基本設定

2.1.1. 使用 Polyglot Notebooks

本操作手冊將使用 Polyglot Notebooks 方便編輯與說明，欲在 VS Code 中使用 Polyglot Notebooks 可於以下連結下載：

<https://code.visualstudio.com/docs/languages/polyglot>



2.1.2. Import Libraries

在開始實作之前必須先將 Firely .NET SDK import 進來，詳細資料可參考

<https://docs.fire.ly/projects/Firely-NET-SDK/en/latest/>

可使用以下方式引入：

```
#r "nuget:h17.fhir.r4"
```

2.1.3. 連接 FHIR Server

連接欲使用的 FHIR Server，於此將使用 Firely 測試 Server 作為範例：

<http://server.fire.ly>

程式碼：

```
using H17.Fhir.Model;  
using H17.Fhir.Rest;  
var client = new FhirClient("http://server.fire.ly");
```

H17.Fhir.Model 包含與 FHIR 資源和資料類型相對應的模型類，例如 Patient 和 HumanName，該程式碼由 FHIR 發布工具生成，用於建立新版本的 FHIR 網站。

建立一個新的 FhirClient Instance，這是透過將 FHIR server endpoint 的 url 作為參數傳遞給建構函數來完成。

2.2. CRUD 操作

2.2.1. 取得單筆資料(使用 id)

以 Patient 為例，於此實作取得 Patient id= 58dbc419-a91c-4870-a3a6-15fdb4a840db 的病患資料，並取得他的姓與名

Patient id= 58dbc419-a91c-4870-a3a6-15fdb4a840db 的資料內容為:

```
{
  "resourceType": "Patient",
  "identifier": [
    {
      "value": "A111111111"
    }
  ],
  "active": true,
  "name": [
    {
      "family": "Lee",
      "given": [
        "dk"
      ]
    }
  ],
  "telecom": [
    {
      "system": "phone",
      "value": "0912345678"
    }
  ],
  "gender": "male",
  "birthDate": "1997-02-18",
  "address": [{
    "city": "Taipei",
    "country": "TW"
  }],
}
```

```
"id": "58dbc419-a91c-4870-a3a6-15fdb4a840db",  
"meta": {  
    "versionId": "439a0580-93f3-486f-b1fe-a3c2cc45df94",  
    "lastUpdated": "2023-11-08T13:28:03.537+00:00"  
}  
}
```

程式碼:

```
var location_A = new Uri("http://server.fire.ly/Patient/58dbc419-a91c-4870-a3a6-15fdb4a840db");  
var pat_A = new Patient();  
pat_A = await client.ReadAsync<Patient>(location:location_A);  
if (pat_A != null){  
    Console.WriteLine(pat_A.Name[0].Family +  
pat_A.Name[0].Given.First());  
}
```

結果: Leedk

2.2.2. 搜尋(Search)

以 Patient 為例，於此實作取得出生年月日在 2020 年之後的病患資料，使用 SearchParams，並取得病患總數和病患們的姓名

程式碼:

```
var q = new SearchParams().Where("birthdate>2020");  
Bundle patients = await client.SearchAsync<Patient>(q);  
  
Console.WriteLine("總計:" + patients.Total.ToString() + "人");  
  
foreach (var p in patients.Entry){  
    var r = (Patient)p.Resource;  
    Console.WriteLine(r.Name[0].Family + " " +  
r.Name[0].Given.First());  
}
```


結果:

總計:240 人

test_family_name2 test_first_name2

test_family_name2 test_first_name2

...

2.2.3. 排序(Order)

以 Patient 為例，於此實作取得出生年月日在 2020 年之後的病患資料，依據出生年月日排序，使用 OrderBy()，並取得病患總數和病患們的姓名

程式碼:

```
var q = new
SearchParams().Where("birthdate>2020").OrderBy("birthdate");
Bundle patients = await client.SearchAsync<Patient>(q);

Console.WriteLine("總計:" + patients.Total.ToString() + "人");
foreach (var p in patients.Entry){
    var r = (Patient)p.Resource;
    Console.WriteLine(r.BirthDate.ToString() + " " + r.Name[0].Family + "
" + r.Name[0].Given.First());
}
```

結果:

總計:240 人

2021-01-01 Merveille Bernard

2021-01-01 Merveille Bernard

2022-01-01 Doe John

...

2.2.4. 新增一筆 Patient (Create)

以 Patient 為例，將先創建一筆 Patient 資料

程式碼:

```
var pat = new Patient();

var id = new Identifier();
id.System = "http://hl7.org/fhir/sid/us-ssn";
id.Value = "000-12-3456";
pat.Identifier.Add(id);

var name = new
HumanName().WithGiven("Christopher").WithGiven("C.H.").AndFamily("Parks
");
name.Prefix = new string[] { "Mr." };
name.Use = HumanName.NameUse.Official;

var nickname = new HumanName();
nickname.Use = HumanName.NameUse.Nickname;
nickname.GivenElement.Add(new FhirString("Chris"));

pat.Name.Add(name);
pat.Name.Add(nickname);

pat.Gender = AdministrativeGender.Male;

pat.BirthDate = "1983-04-23";

var birthplace = new Extension();
birthplace.Url = "http://hl7.org/fhir/StructureDefinition/birthPlace";
birthplace.Value = new Address() { City = "Seattle" };
pat.Extension.Add(birthplace);

var birthtime = new
Extension("http://hl7.org/fhir/StructureDefinition/patient-
birthTime", new FhirDateTime("1983-04-23"));
pat.BirthDateElement.Extension.Add(birthtime);
```

```

var address = new Address()
{
    Line = new string[] { "3300 Washtenaw Avenue, Suite 227" },
    City = "Ann Arbor",
    State = "MI",
    PostalCode = "48104",
    Country = "USA"
};
pat.Address.Add(address);

var contact = new Patient.ContactComponent();
contact.Name = new HumanName();
contact.Name.Given = new string[] { "Susan" };
contact.Name.Family = "Parks";
contact.Gender = AdministrativeGender.Female;
contact.Relationship.Add(new
CodeableConcept("http://hl7.org/fhir/v2/0131", "N"));
contact.Telecom.Add(new
ContactPoint(ContactPoint.ContactPointSystem.Phone, null, ""));
pat.Contact.Add(contact);

pat.Deceased = new FhirBoolean(false);

```

建立後使用 CreateAsync 新增該筆 Patient 資料:

```

var created_pat = await client.CreateAsync<Patient>(pat);
Console.WriteLine("新增資料的 ID:" + created_pat.Id.ToString());

```

結果:

新增資料的 ID:0fd4afc3-17a0-4560-a800-835455d81b89

當出現問題時將拋出異常，在大多數情況下是 `FhirOperationException`。此異常有一個包含 `OperationOutcome` 資源的 `Outcome` 屬性，您可以檢查該屬性以查找有關交互失敗原因的更多資訊，大多數 FHIR Server 將在 `OperationOutcome` 中傳回人類可讀的錯誤描述來幫助您解決問題。

如果操作成功，Server 將傳回所建立資源的實例，其中包含 ID、metadata 以及您剛發佈到 Server 的儲存資料的副本。根據 Server 實現的不同，這可能與您發送的內容有所不同。例如如果您的請求中未設定這些欄位的值，Server 可能會使用預設值填入欄位。

2.2.5. 更新 (Update)

以 Patient 為例，欲更新剛剛新增的 ID=0fd4afc3-17a0-4560-a800-835455d81b89 的病患資料(例如把 name.family 改為"Hello World")，使用 Update

程式碼:

```
var location_A = new Uri("http://server.fire.ly/Patient/0fd4afc3-17a0-4560-a800-835455d81b89");
var pat_A = new Patient();
pat_A = await client.ReadAsync<Patient>(location:location_A);
if (pat_A != null){
    pat_A.Name[0].Family = "Hello World";
    var updated_pat = client.Update<Patient>(pat_A);
    Console.WriteLine("更新後的 ID:" + updated_pat.Id.ToString());
    Console.WriteLine("更新後的版本:" + updated_pat.Meta.VersionId);
    Console.WriteLine("更新後的姓名:" + updated_pat.Name[0].Family);
}
```

結果:

更新後的 ID:0fd4afc3-17a0-4560-a800-835455d81b89

更新後的版本:686cc416-208b-45bb-acc3-ba055e24c525

更新後的姓名:Hello World

2.2.6. 刪除 (Delete)

欲刪除 ID=0fd4afc3-17a0-4560-a800-835455d81b89 的病患資料，使用 Delete

程式碼：

```
// Delete based on a url or resource location
var location = new Uri("http://server.fire.ly/Patient/0fd4afc3-17a0-4560-a800-835455d81b89");
client.Delete(location);
// or
// client.Delete("Patient/33");

// You may also delete based on an existing resource instance
// client.Delete(pat_A);
```

如果資源已被刪除或資源在刪除之前不存在，則刪除互動將失敗並拋出

FhirOperationException，並且 Server 傳回錯誤來指示這一點。

3. 情境應用

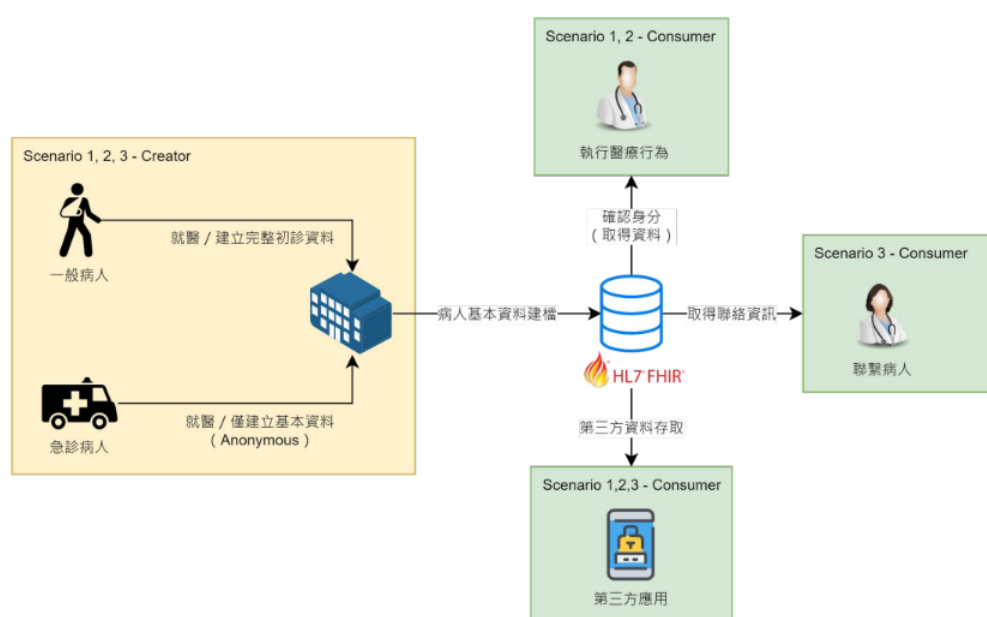
3.1. 情境說明

本章節將以 2023 臺灣醫學資訊互通聯測松之 Track1 和 Track2，分別為病人基本資料和生理量測數據(以體重為例)賽道以實作 Firely .NET SDK。

3.2. Track1 病患基本資料

Track1 病患基本資料為醫療資訊系統中最基本元素，涵蓋所有電子病歷的範疇，本賽道主要分析台灣病人基本資料之規格，根據國內現況制定出適合台灣病患基本資要用跨系統交換，以達到基本的互通性，並相容台灣核心規範(TW Core IG)。

詳細規範請參考 2023 臺灣醫學資訊互通聯測松官網: <https://mitw.dicom.org.tw/>



Track1 病患基本資料賽道情境圖(來源: 聯測松官網)

於此將以情境 1 (SC1)作為示範，並以 TW Core IG patient Profile 為規範。

情境 1 (SC1)核心資料交換包含: 首次建檔時建立的病人身分資訊，最少僅需輸入 identifier、gender 即可，且須符合台灣核心-病人規範格式。

3.2.1. 首次建檔病患資料

請參考 [2.2.4. 新增一筆 Patient \(Create\)](#)

3.2.2. 更新該筆病患資料

請參考 [2.2.5. 更新 \(Update\)](#)

3.2.3. 刪除該筆病患資料

請參考 [2.2.6. 刪除 \(Delete\)](#)

3.3. Track2 生理量測數據

Track2 生理量測數據是為了患者監測和識別健康變化或問題而記錄的重要信息。生理量測數據通常在各種醫療保健場所定期和常規進行，包括入院檢查、住院監護、術後醫療程序、治療和療法程序以及密切監護。

3.3.1. 設定 FHIR Server

```
using Hl7.Fhir.Model;
using Hl7.Fhir.Rest;
var client = new FhirClient("http://server.fire.ly");
```

3.3.2. 新增一筆體重紀錄

以體重為例，創建一筆 Observation 資料

程式碼:

```
var observation = new Observation();
observation.Status = Hl7.Fhir.Model.ObservationStatus.Preliminary;
var category = new CodeableConcept();
var coding = new CodeableConcept();
var o_value = new Hl7.Fhir.Model.Quantity();

category.Add("http://hl7.org/fhir/observation-category", "vital-signs");
observation.Category.Add(category);
coding.Add("http://loinc.org", "29463-7", "Body weight");
observation.Code = coding;
observation.Effective = new FhirDateTime("2023-04-19");
o_value.Value = 85;
o_value.System = "http://unitsofmeasure.org";
o_value.Code = "kg";
observation.Value = o_value;
var rr = new ResourceReference("Patient/" + patient.Id);
observation.Subject = rr;
```

上傳該筆 Observation:

```
var created = await client.CreateAsync<Observation>(observation);
Console.WriteLine(created.Id);
```


結果: fbdae518-011c-4813-8261-9202c9061793

3.3.3. 搜尋該筆體重資料

查詢剛剛新增的體重資料:

```
var q = new SearchParams().Where("Subject:id=" + patient.Id);
Bundle observations = await client.SearchAsync<Observation>(q);

foreach (var o in observations.Entry){
    var r = (Observation)o.Resource;
    Console.WriteLine( r.Id.ToString());
}
```

結果:

fbdae518-011c-4813-8261-9202c9061793 ←
d2c4f3f8-2ee4-4694-a934-94ccb8d4cb44
42434706-5bad-42f3-bcb7-c9cd1281f733
...

3.3.4. 更新體重數據

欲更新剛剛新增的 ID= fbdae518-011c-4813-8261-9202c9061793 的體重數據(改為 100kg)，使用 Update

程式碼:

```
var location_A = new Uri("https://server.fire.ly/Observation/fbdae518-011c-4813-8261-9202c9061793");
var ob = new Observation();
ob = await client.ReadAsync<Observation>(location:location_A);
if (ob != null){
    ob.Value = new Quantity(){
        Value = 100,
        Unit = "kg"
    };
};
var updated_pat = client.Update<Observation>(ob);
Console.WriteLine("更新後的 ID:" + updated_pat.Id.ToString());
Console.WriteLine("更新後的版本:" + updated_pat.Meta.VersionId);
```

```
}
```

結果:

更新後的 ID:fbdae518-011c-4813-8261-9202c9061793

更新後的版本:c5590403-e0a2-4de9-990c-82448c7c3c93

GET https://server.fire.ly/Observation/fbdae518-011c-4813-8261-9202c9061793

200 OK 639 ms 1.28 KB

```
18 ],
19   "code": {
20     "coding": [
21       {
22         "system": "http://loinc.org",
23         "code": "29463-7",
24         "display": "Body weight"
25       }
26     ]
27   },
28   "subject": {
29     "reference": "https://server.fire.ly/Patient/3f9d760d-08af-4e93-8575-5bf23883766d"
30   },
31   "effectiveDateTime": "2023-04-19",
32   "valueQuantity": {
33     "value": 100,
34     "unit": "kg"
35   }
36 }
```

3.3.5. 刪除該筆體重資料

欲刪除 ID= fbdae518-011c-4813-8261-9202c9061793 的體重資料，使用 Delete

程式碼:

```
var location = new Uri("https://server.fire.ly/Observation/fbdae518-011c-4813-8261-9202c9061793");
client.Delete(location);
```