

上下2分割・両面で
印刷すると冊子形式
で見やすくなります

DIYで始めよう！ HL7 FHIR/REST/JSON 90分チュートリアルキット

このチュートリアルキットは、第23回医療情報学春季学術大会 チュートリアル6
「HL7 FHIRをどう導入するか」（日本Mテクノロジー学会主催）を元にしたものです。

文責 一般社団法人日本Mテクノロジー学会

鳥飼 幸太（群馬大学医学部附属病院 システム統合センター）

土井 俊祐（東京大学医学部附属病院 企画情報運営部）



本チュートリアルキットの目的

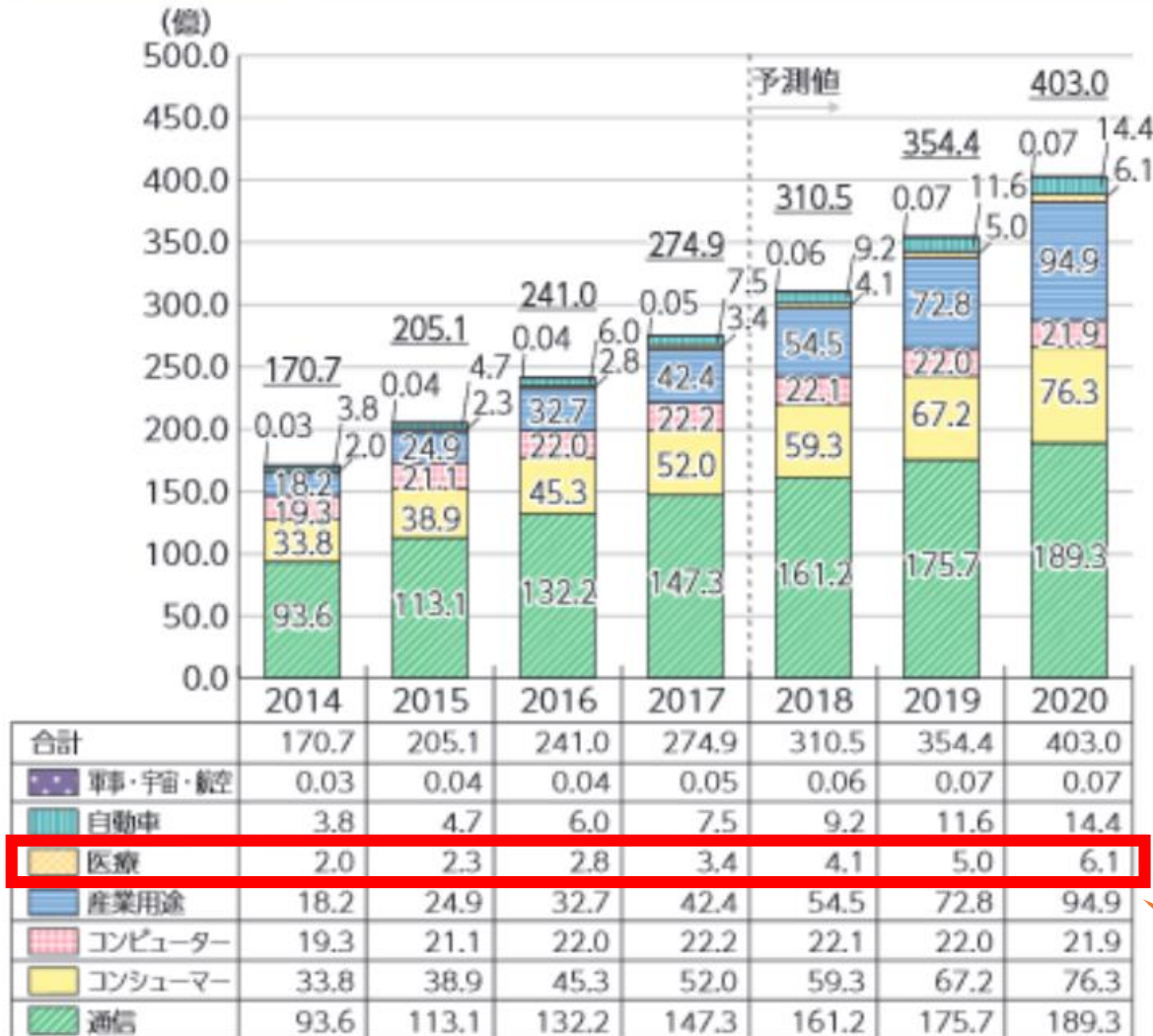
- FHIRの構成要素について、コーディングスキルを身につける
 - -REST
 - -JSON
- Excel VBAからREST/JSONを用いて、FHIRサーバからデータを取得するコーディングを身につける
- これらの取り組みを通じて、FHIRがREST/JSONを採用する、開発上のメリットについて具体的実感をもつ

本チュートリアルキットの流れ

- HL7 FHIRについて
 - 医療情報ユーザーから見たFHIRアプリケーションの利点
 - FHIRで利用するREST, JSONについて
- ハンズオン ～ExcelでDIYするFHIR～（90分程度）
 - 準備（環境の確認、モジュールのダウンロード・インポート）
 - ハンズオン1：サーバからRESTでデータを取得する
 - ハンズオン2：JSONをパースしてVBAのオブジェクトに変換する
 - ハンズオン3：該当患者の複数回の検査結果を取得し、表形式で入力する
 - 選択ハンズオンA&B：FHIRから取得したデータをVBAで加工し様式に挿入
- 日本Mテクノロジー学会のご紹介

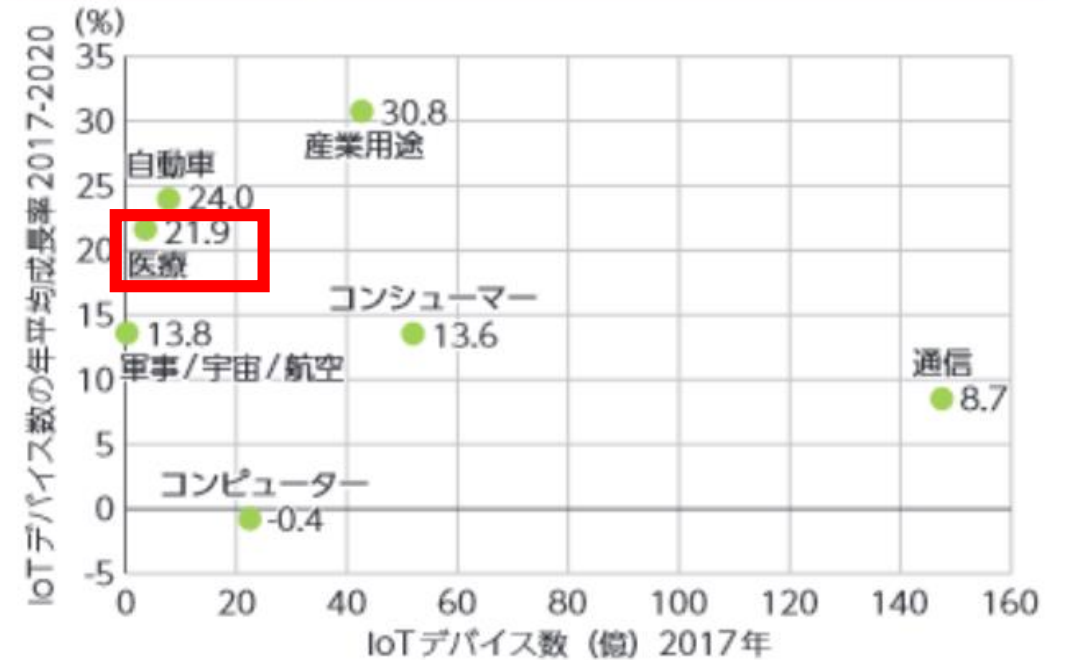
IoTデバイスの広く浸透する医療への変化

図表 1-1-2-1 世界のIoTデバイス^{*2}数の推移及び予測



(出典) IHS Technology

図表 1-1-2-2 分野・産業別のIoTデバイス数及び成長率予測



(出典) IHS Technology

総務省 平成30年版 通信情報白書より引用

様々な産業でIoTデバイスの急速な導入が進んでいますが、医療分野でも同様です。

FHIRとは

FHIRとは



FHIRの略称

- F - Fast
 - 短期間で開発・導入が可能
- H - Health
 - 医療が対象分野
- I - Interoperable
 - 医療情報の相互運用性
- R - Resources
 - リソース

FHIRのコンセプト

- ONCの方針への対応
 - Simple
 - Easy
 - Modular
- 医療情報は特殊？
 - 金融、商取引、その他一般の技術は使えないか？
 - プライバシー、アクセス権の扱いは少し特殊かも？
- 経済的であること



FHIRは、米国HL7協会が発表した新しい医療情報の標準規格です。

実装を重視し、早く・簡単に実装できることをコンセプトとしています。

過去の標準規格と相互に活用することができ、相互運用性を確保したまますぐに使用することができます。

JSON, XML, HTTPといったWeb標準の基盤を使用できます。

2017年3月8日”FHIRのご紹介”(HL7協会)より抜粋
http://www.hl7.jp/docs/60seminar_2_HL7.pdf

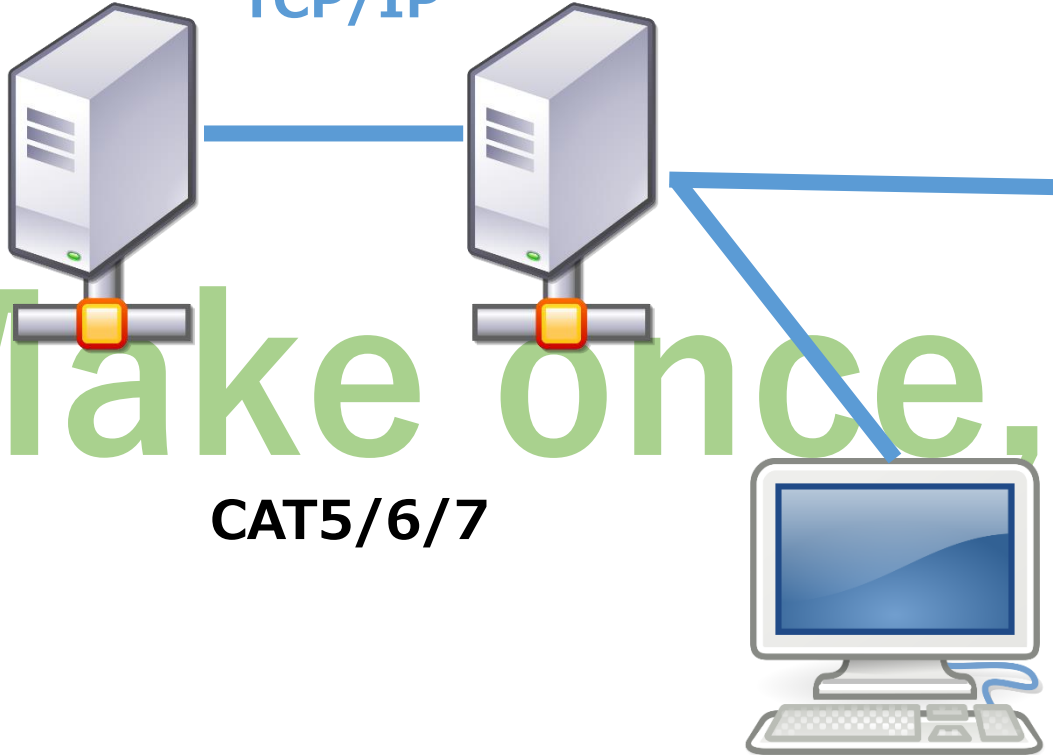
日本HL7協会が様々な資料を掲載していますので、ぜひ一通り読んでみて下さい。

現行システムとFHIRの適用範囲

FHIRの通信は、柔軟な開発や連携が必要なシステムで力を発揮します。

Wired (Fixed) Network

TCP/IP



Mobile Network

HTTP



IEEE802.11
a/b/g/n/ac



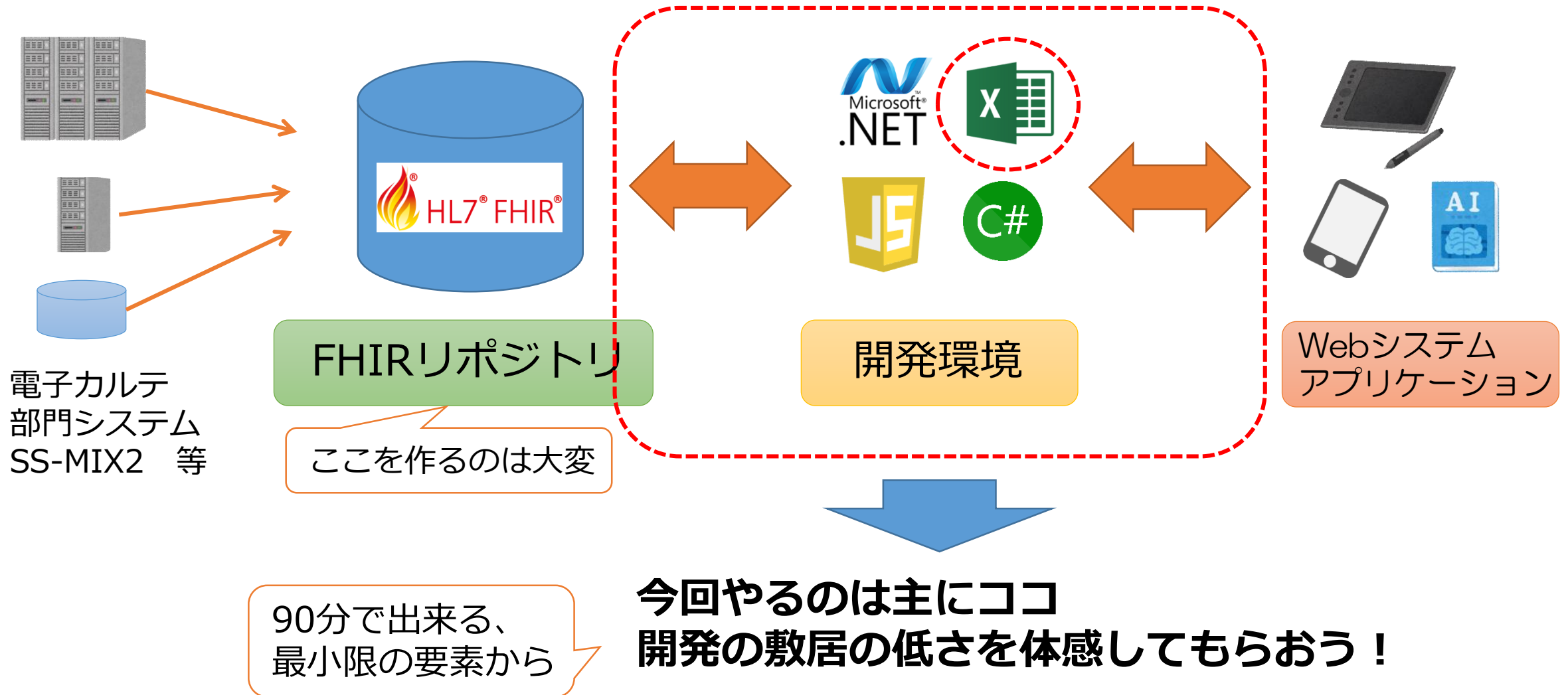
“Make once, Use Many”

スタッフが**情報の側**に移動する
共通タスクを複数スタッフで分担
部門化の流れとともに浸透
共通の端末

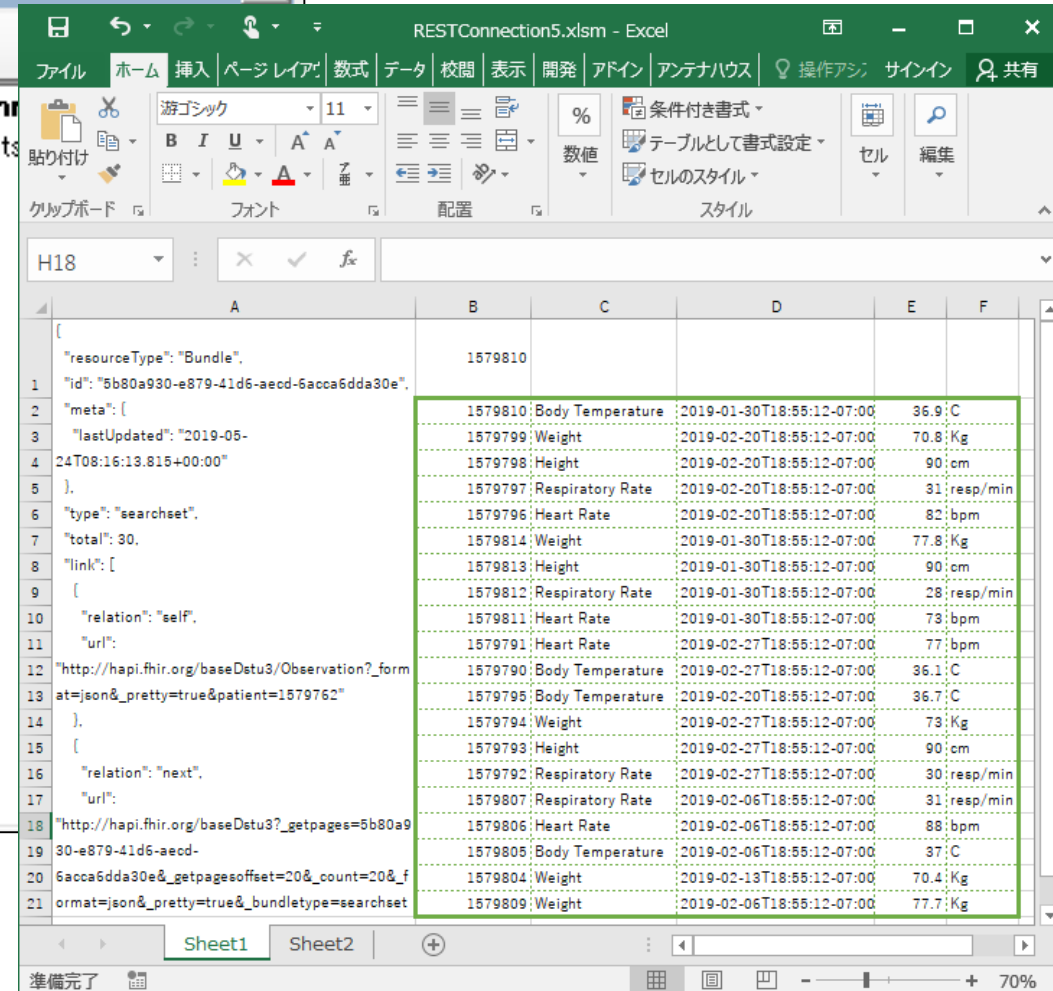
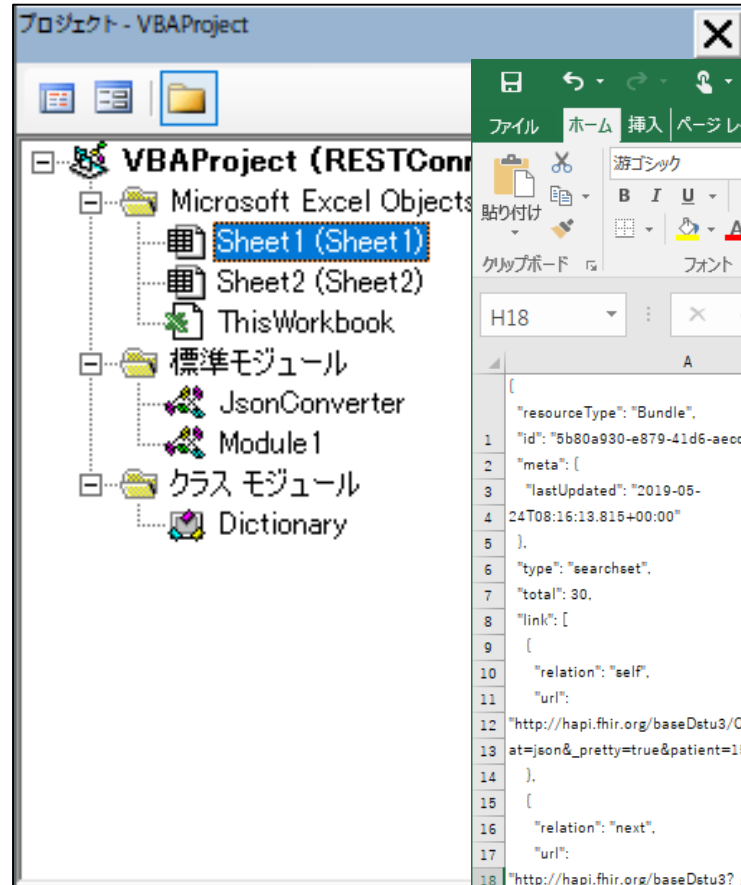
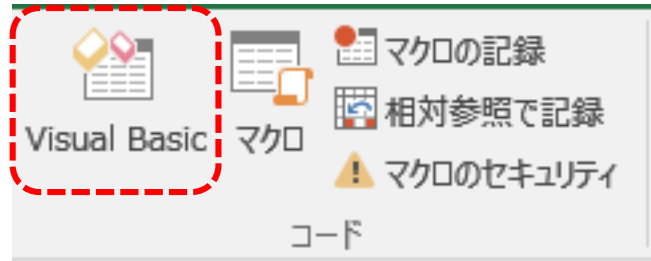
スタッフが**情報とともに**移動する
個別タスクを個別に分担
チーム医療・情報の電子化、即時化
目的ごとのデバイス

本チュートリアルの内容

- FHIRを現場に導入するまでの道のり



このチュートリアル構成

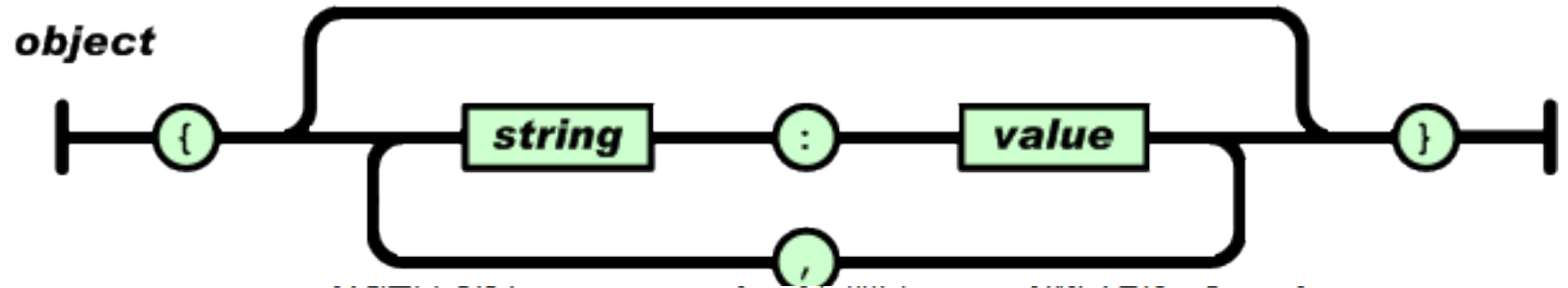


- 本チュートリアルは、Excel VBAを使用してFHIRの通信を実体験します。
- Excelは2013以上のバージョンをご用意下さい。
- ExcelでJsonを扱うために、2つの外部モジュールを使用します。これらはGitHubという開発者向けのオープンソースの共有サイトからダウンロードします。（後述）

使うもの 1 : JSON

- JSON=JavaScript Object Notation
- JavaScriptの文法に沿っています
- 内容 1 : 名前/値
- 内容 2 : 配列

<https://www.json.org/json-ja.html>



例

{ "羊羹" : "とらや" }

```
{ "name" : "John Smith",  
  "sku" : "20223",  
  "price" : 23.95,  
  "shipTo" : { "name" : "Jane Smith",  
               "address" : "123 Maple Street",  
               "city" : "Pretendville",  
               "state" : "NY",  
               "zip" : "12345" },  
  "billTo" : { "name" : "John Smith",  
               "address" : "123 Maple Street",  
               "city" : "Pretendville",  
               "state" : "NY",  
               "zip" : "12345" }  
}
```

使うもの2 : REST

- REST (REpresentational State Transfer)
 - HTTP通信 (URL) でアクセスしデータ送受信を行うWebサービス

RESTで使用するURLの例

"http://hapi.fhir.org/baseDstu3/Patient/1646554/_history/1?_format=json"

データの場所 (WebサイトのURL)

引数 (取得するデータの条件指定)

RESTは設計に際し以下を設計原則とするよう提言されている

1. アドレス指定可能なURIで公開されていること
2. インターフェース(HTTPメソッドの利用)の統一がされていること
3. ステートレスであること
4. 処理結果がHTTPステータスコードで通知されること

処理	HTTPメソッド	CRUD操作
登録	POST	CREATE
取得	GET	READ
更新	PUT	UPDATE
削除	DELETE	DELETE

今回は主に利用者の観点からGETメソッドを使用します

ハンズオン準備 1 (外部モジュールのダウンロード)

- VBA-JSON (JsonConverter)

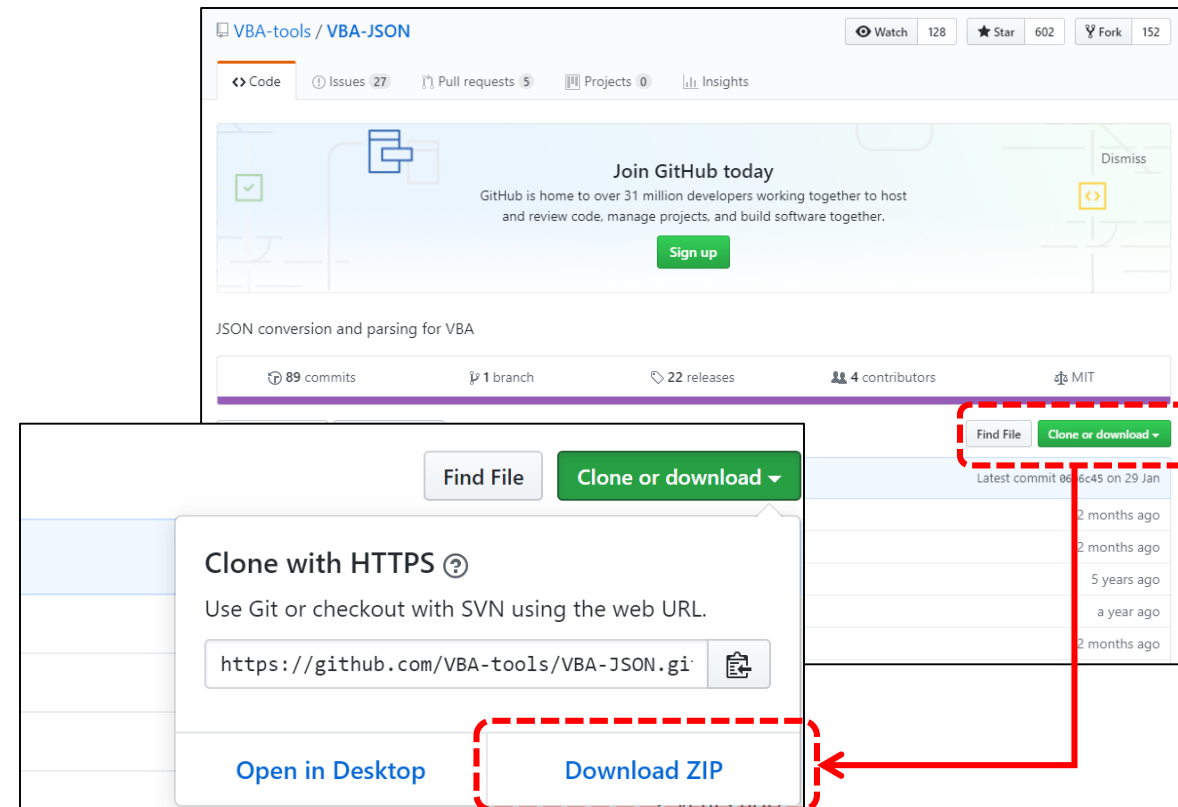
<https://github.com/VBA-tools/VBA-JSON>

- VBA-Dictionary (Dictionary)

<https://github.com/VBA-tools/VBA-Dictionary>

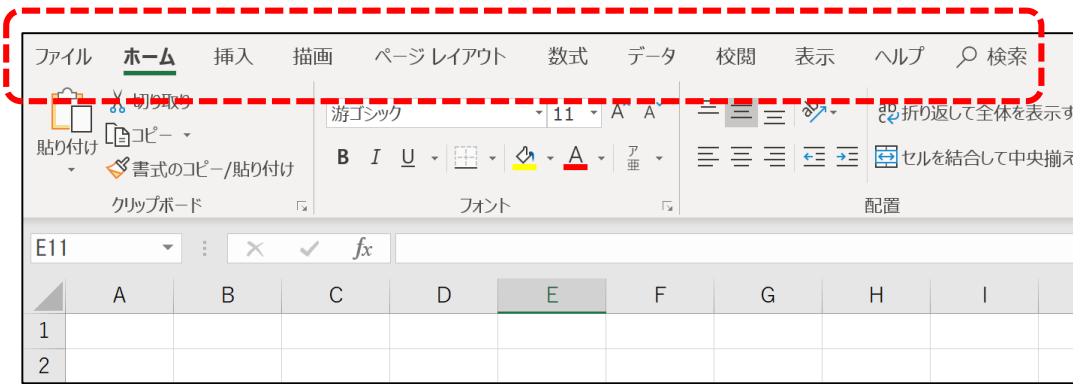
- 2つのツールをそれぞれ右上の「Clone or download」からZipファイルをダウンロードして下さい

- ダウンロードしたら、zipファイルを任意の場所に展開しておいて下さい



ハンズオン準備 2（開発タブの追加）

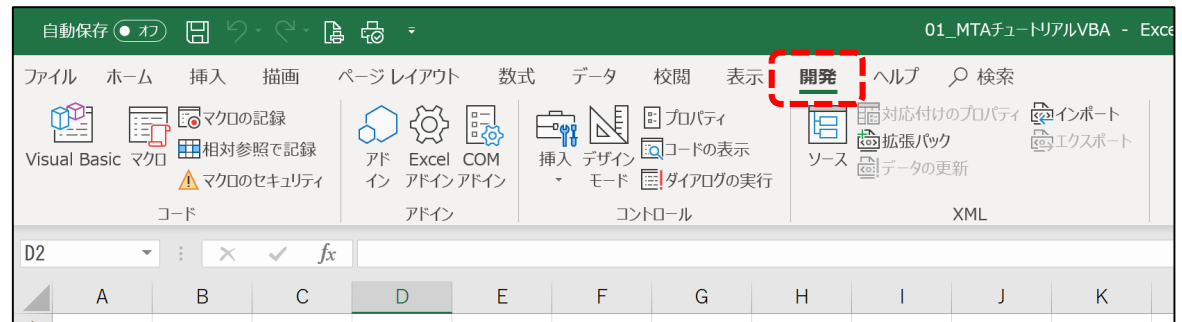
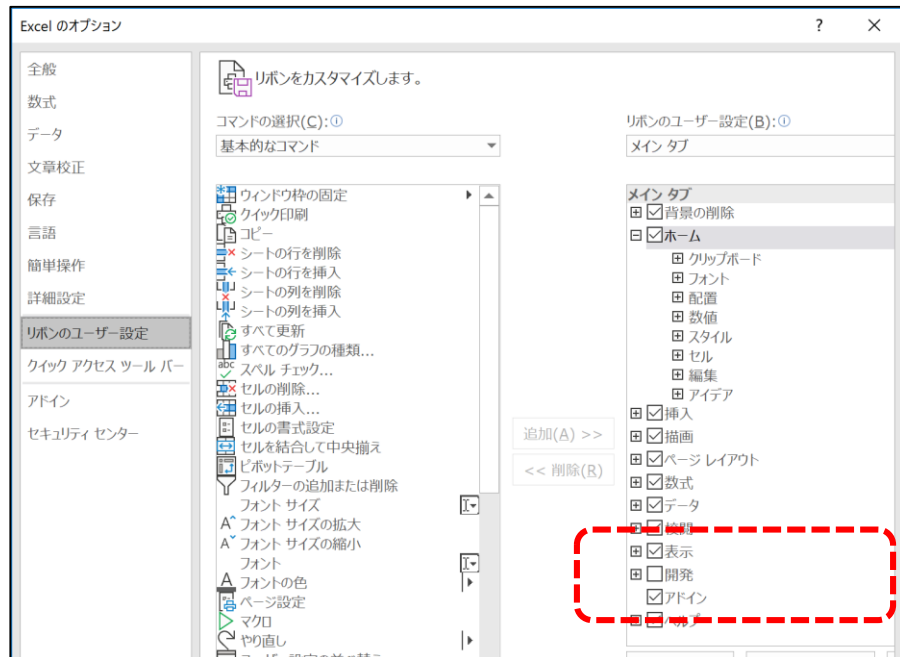
- ・ 開発タブがない方はオプションから追加して下さい。



- ・ 「開発」タブがある方はそのまま次ページへ
- ・ タブがない方は下の手順で追加

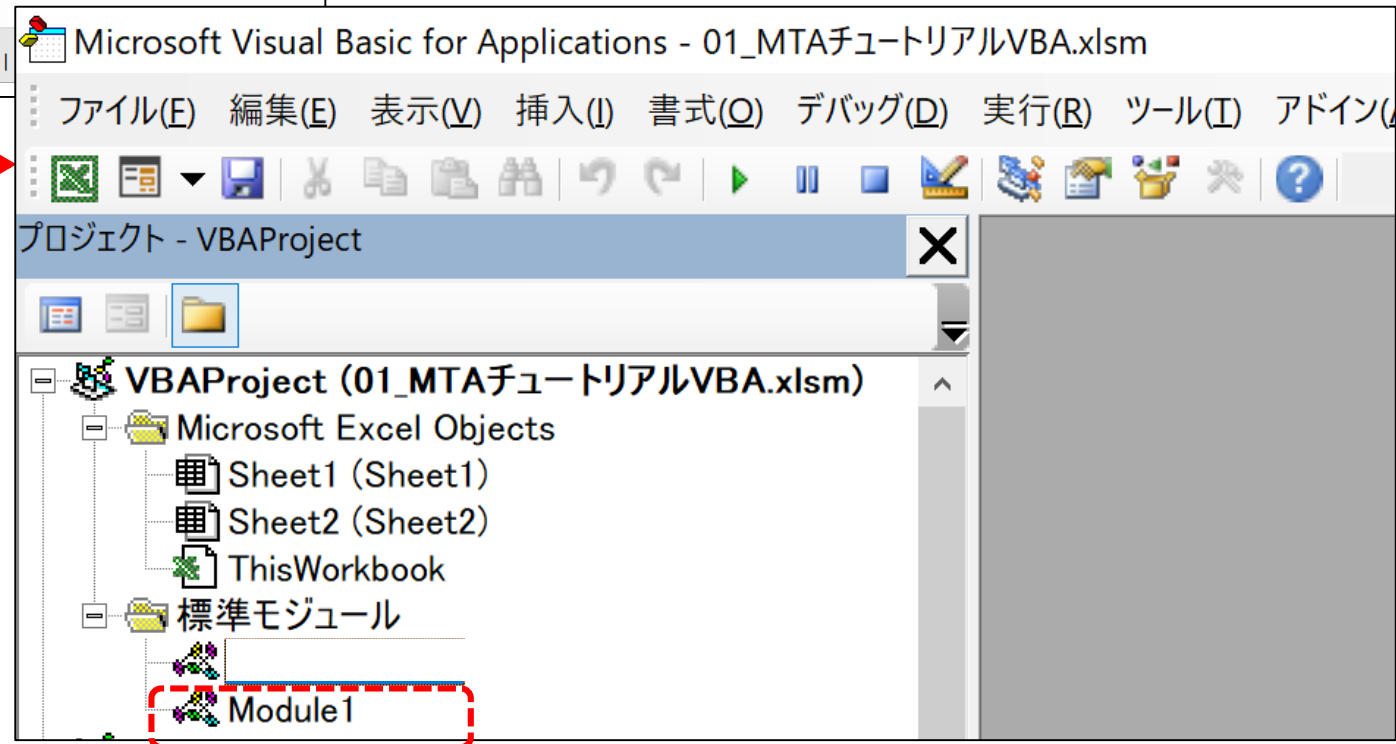
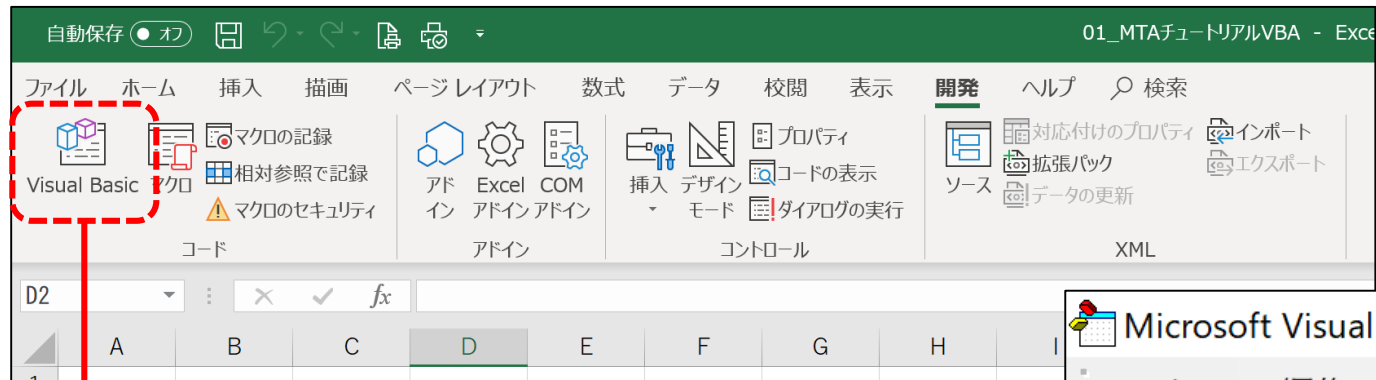
- ・ ファイル → オプション
- ・ Excelのオプションが開いたら、「リボンのユーザー設定」で「開発」のタブにチェックを入れる
- ・ OKを押下

※Excelのバージョンにより設定する場所が異なることがあります。



ハンズオン準備 2 (開発タブの追加)

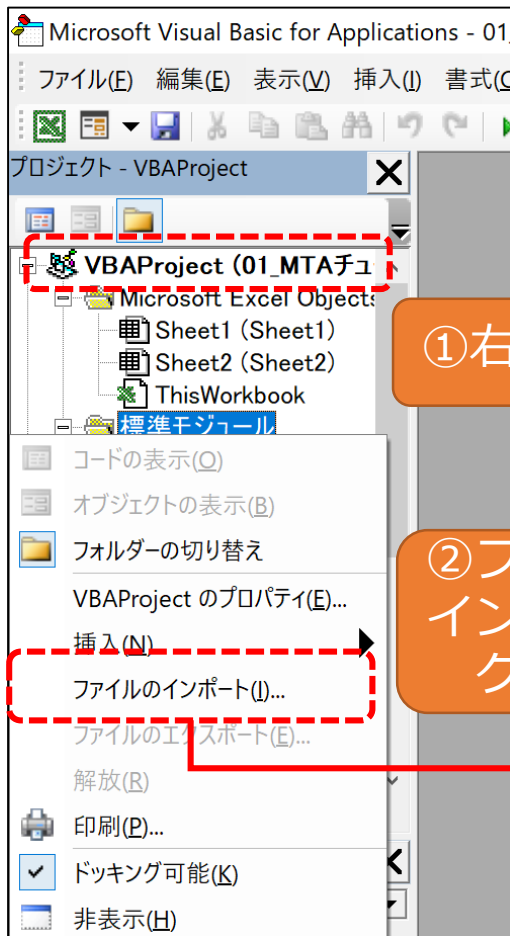
- 「Visual Basic」のボタンを押すと、VBAの開発画面が開きます



「MTAチュートリアル」のファイルには標準モジュールが1つ作成されています

ハンズオン準備3 (JsonConverterのインポート)

- GitHubでダウンロードした「VBA-JSON」「VBA-Dictionary」のzipファイルを解凍しておきインポートします

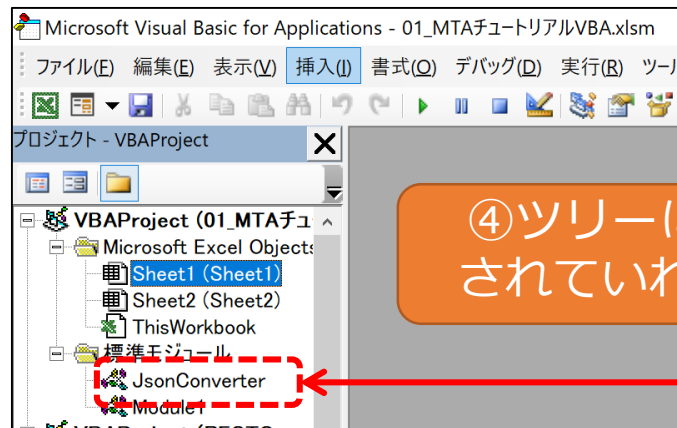


①右クリック

②ファイルのインポートをクリック



③「JsonConverter.bas」を選択し開く



④ツリーに展開されていればOK

同様に「Dictionary.cls」もインポートします

ハンズオン準備 4 (VBAプログラムの実行方法)

- ・モジュール、プロシージャを選択して「▶」ボタンで実行します

③ 「▶」 (F5) を押す

The screenshot shows the VBA editor interface. The 'Project - VBAProject' window on the left shows 'Module1' selected under '標準モジュール'. The main editor window shows the code for 'Module1 (コード)'. The 'Run' button (▶) in the ribbon is highlighted with a red dashed box. A red arrow points from the 'Run' button to the 'Initialize' procedure in the code. The code includes several string declarations and a 'Sub Initialize()' procedure with a comment: 'FHIRリポジトリへの接続先のURLを指定'. The cursor is positioned at the end of the 'Initialize' procedure.

① 使用中のモジュール

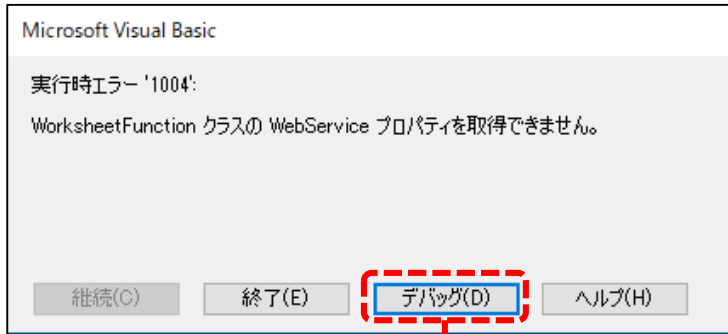
② 実行するプロシージャをアクティブにする
(入力カーソルが点滅している状態。SubからEnd Subの間であればどこでもOK)

The 'Macro' dialog box is shown. The 'Macro名(M):' list contains 'hansonA', 'hansonB', 'Initialize', 'RestConnection', and 'RestConnectionformac'. 'hansonA' is selected. The 'マクロの場所(A):' dropdown shows 'VBAProject (01_MTAチュートリアルFHIR-VBA.xlsm)'. Buttons for '実行(R)', 'キャンセル', 'ステップ イン(S)', '編集(E)', '作成(C)', and '削除(D)' are visible.

どのプロシージャもアクティブでない場合、実行するプロシージャを選択するポップアップが表示されます。

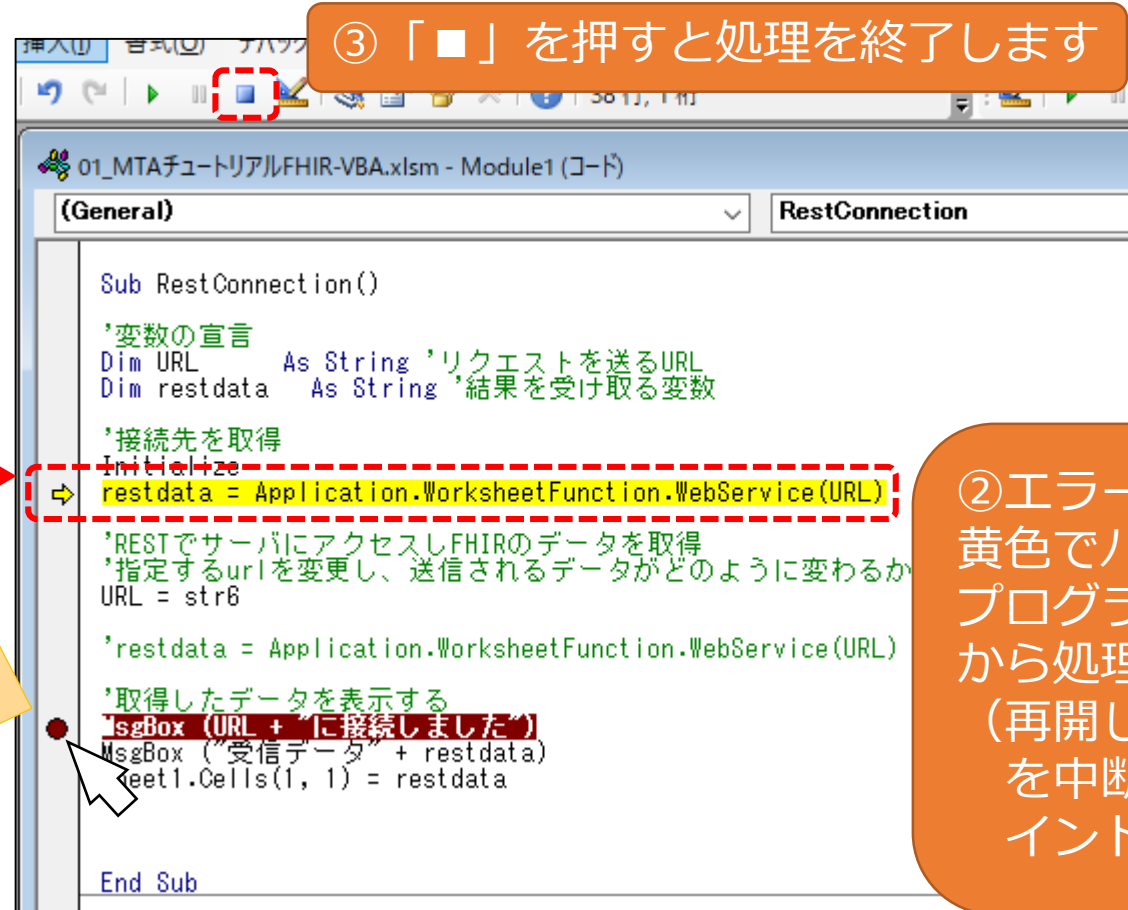
ハンズオン準備 5 (エラー処理、デバッグ)

- ・エラーが出たら「デバッグ」を押下してエラーの場所を確認します



①デバッグを押下

④左側の灰色の部分をクリックすると、ブレークポイントを設定できます。ここで処理を1度ストップさせることができます。再開する場合は「▶」を押下するか、ブレークポイントを削除します。



②エラーが発生している箇所が黄色でハイライトされます。プログラムを修正すると、ここから処理が再開されます。(再開したくない場合は③処理を中断するか、④ブレークポイントを設定しておきます)

ハンズオン準備（注意点）

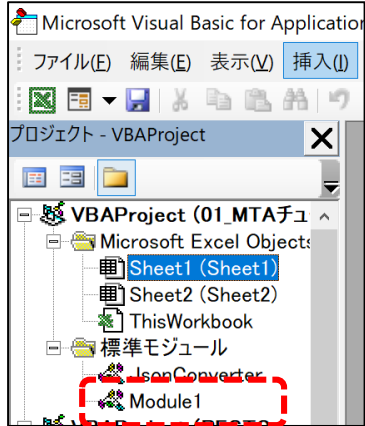
- VBA-JSONのモジュールをインストールする前にプログラムを実行すると、後からインストールしてもエラーが出る場合があります。
- その場合は、Excelを一度保存して閉じ、再度ファイルを開くことで解消します。

- 今回使用するFHIRのテストサイト「Hapi」は、世界中からアクセスがあり、データの書き換えも自由に行われます。
- そのため、該当のPatient IDやObservation IDのデータが消えることがあります。その場合は、PatientやObservationのデータが存在するURLに変更する必要があります。
→Entryを使用することで有効なIDを調べられますが、不慣れな方は事務局にご連絡いただければお知らせできます。

次のページからハンズオンが始まります

ハンズオン 1 : サーバにRESTでアクセスする(1/5)

- 「標準モジュール」の中から使用する「Module 1」を開きます



① Module1を開きます

```
(General) | RestConnection
Public restdata As String
Public str1 As String
Public str2 As String
Public str3 As String
Public str4 As String
Public str5 As String

Sub Initialize()
' FHIRリポジトリへの接続先のURLを指定
str1 = "http://hapi.fhir.org/baseDstu3/Patient?pretty=true"
str2 = "http://hapi.fhir.org/baseDstu3/Patient/1646554"
str3 = "http://hapi.fhir.org/baseDstu3/Patient/1646554/_history/1?_format=json"
str4 = "http://hapi.fhir.org/baseDstu3/Observation/23129"
str5 = "http://hapi.fhir.org/baseDstu3/Observation?pretty=true&code=8310-5&patient=23129&_format=json"
End Sub

Sub RestConnection()
' 変数の宣言
Dim url As String ' リクエストを送るURL
Dim restdata As String ' 結果を受け取る変数

' 接続先を取得
Initialize

' RESTでサーバにアクセスしFHIRのデータを取得
' 指定するurlを変更し、送信されるデータがどのように変わるか見てみましょう。
url = str2
restdata = Application.WorksheetFunction.WebService(url)

' 取得したデータを表示する
MsgBox (url + "に接続しました")
MsgBox ("受信データ" + restdata)
Sheet1.Cells(1, 1) = restdata
End Sub

Sub RestConnectionformac()
' 変数の宣言
Dim xmlHttp As Object ' HTTPリクエストオブジェクト
Dim url As String ' リクエストを送るURL
Dim strParam As String ' リクエストパラメータ
Dim restdata As String ' 結果を受け取る変数

' 接続先を取得
Initialize
```

ここでは、Module1の中にある以下の3つのプロシージャを使います

- Initialize
→ FHIRサーバのURL指定
- RestConnection
→ 基本はこちらを使用 (RESTを使用するコードが簡単)
(Windowsの方はこちら)
- RestConnectionformac
→ RestConnectionでエラーが出る方はこちらを使用
(Macの方はこちら)

ハンズオン 1 : サーバにRESTでアクセスする(2/5)

- 「Initialize」ではRESTで接続するFHIRサーバのURLを指定します

```
Sub Initialize()  
  'FHIRリポジトリへの接続先のURLを指定
```

```
  str1 = "http://hapi.fhir.org/baseDstu3/Patient?_pretty=true"  
  str2 = "http://hapi.fhir.org/baseDstu3/Patient/1646554"  
  str3 = "http://hapi.fhir.org/baseDstu3/Patient/1646554/_history/1?_format=json"  
  str4 = "http://hapi.fhir.org/baseDstu3/Observation/1579810"  
  str5 = "http://hapi.fhir.org/baseDstu3/Observation/1579810/_history/1?_format=json"  
  str6 = "http://hapi.fhir.org/baseDstu3/Observation?_pretty=true&patient=1579762&_format=json"  
  str7 = "http://hapi.fhir.org/baseDstu3/Observation?_pretty=true&code=8310-5&patient=1579810&_format=json"
```

```
End Sub
```

②サンプルとして7つのURLを指定しています

| str3 = "http://hapi.fhir.org/baseDstu3/Patient/1646554/_history/1?_format=json"

サーバアドレス

データベース

id

最新の履歴 json形式で出力

リソースの種類（ここではpatient（基本情報）、Observation（検査結果））

Point

このURLをコピーしてブラウザで見ると、どのようなデータが返ってくるかわかりやすいです。

cf : リソースとは

- FHIRリポジトリに保存されるデータ構造そのもの

The screenshot shows the HL7 FHIR R4 website. At the top, there is a navigation bar with links for Home, Getting Started, Documentation, Resources, Profiles, Extensions, Operations, and Terminologies. Below this is a breadcrumb trail: Table of Contents > Resources. A yellow banner states: "This is the current officially released version of FHIR, which is R4 (v4.0.0). For a full list of all versions, see the Directory of published versions." The main heading is "1.2 Resource Index". Below this is a table with columns for "FHIR Infrastructure Work Group", "Maturity Level: N/A", and "Standards Status: Informative". A paragraph explains the page's purpose: "This page is provided to help find resources quickly. There is also a more detailed classification, ontology, and description. For background to the layout on the layers in this page, see the Architect's Overview. See also the abstract Base Resources Resource and DomainResource." Below the paragraph are tabs for "Categorized", "Alphabetical", "R2 Layout", "By Maturity", "Security Category", "By Standards Status", and "By Committee". The "Categorized" tab is active, showing a grid of resource categories: Conformance, Terminology, Security, Documents, Other, Individuals, Entities #1, Entities #2, Workflow, and Management. The "Individuals" category is highlighted with a red dashed box and a yellow callout box containing the text "Patientのリソース".

Category	Resources
Conformance	• CapabilityStatement N • StructureDefinition N • ImplementationGuide 1 • SearchParameter 3 • MessageDefinition 1 • OperationDefinition N • CompartmentDefinition 1 • StructureMap 2 • GraphDefinition 1 • ExampleScenario 0
Terminology	• CodeSystem N • ValueSet N • ConceptMap 3 • NamingSystem 1 • TerminologyCapabilities 0
Security	• Provenance 3 • AuditEvent 3 • Consent 2
Documents	• Composition 2 • DocumentManifest 2 • DocumentReference 3 • CatalogEntry 0
Other	• Basic 1 • Binary N • Bundle N • Linkage 0 • MessageHeader 4 • OperationOutcome N • Parameters N • Subscription 3
Individuals	• Patient N • Practitioner 3 • PractitionerRole 2
Entities #1	• Organization 3 • OrganizationAffiliation 0 • HealthcareService 2
Entities #2	• Substance 2 • BiologicallyDerivedProduct 0 • Device 0
Workflow	• Task 2 • Appointment 3 • AppointmentResponse 3
Management	• Encounter 2 • EpisodeOfCare 2 • Flag 1

8.1.2 Resource Content

Structure

UML

XML

JSON

Turtle

R3 Diff

All

Structure

Name	Flags	Card.	Type	Description & Constraints
Patient	N		DomainResource	Information about an individual or animal receiving health care services Elements defined in Ancestors: id , meta , implicitRules , language , text , contained , extension , modifierExtension
identifier	Σ	0..*	Identifier	An identifier for this patient
active	?! Σ	0..1	boolean	Whether this patient's record is in active use
name	Σ	0..*	HumanName	A name associated with the patient
telecom	Σ	0..*	ContactPoint	A contact detail for the individual
gender	Σ	0..1	code	male female other unknown AdministrativeGender (Required)
birthDate	Σ	0..1	date	The date of birth for the individual
deceased[x]	?! Σ	0..1		Indicates if the individual is deceased or not
deceasedBoolean			boolean	
deceasedDateTime			dateTime	
address	Σ	0..*	Address	An address for the individual
maritalStatus		0..1	CodeableConcept	Marital (civil) status of a patient MaritalStatus (Extensible)
multipleBirth[x]		0..1		Whether patient is part of a multiple birth
multipleBirthBoolean			boolean	
multipleBirthInteger			integer	
photo		0..*	Attachment	Image of the patient
contact	I	0..*	BackboneElement	A contact party (e.g. guardian, partner, friend) for the patient <i>+ Rule: SHALL at least contain a contact's details or a reference to an organization</i>
relationship		0..*	CodeableConcept	The kind of relationship Patient Contact Relationship (Extensible)
name		0..1	HumanName	A name associated with the contact person
telecom		0..*	ContactPoint	A contact detail for the person
address		0..1	Address	Address for the contact person
gender		0..1	code	male female other unknown AdministrativeGender (Required)
organization	I	0..1	Reference(Organization)	Organization that is associated with the contact

Patientリソースの構造（階層構造）

ハンズオン 1 : サーバにRESTでアクセスする(3/5)

- WorksheetFunction オブジェクトを利用して指定したURLからデータを取得する

```
Sub RestConnection()  
  
' 変数の宣言  
Dim url As String ' リクエストを送るURL  
Dim restdata As String ' 結果を受け取る変数  
  
' 接続先を取得  
Initialize  
  
' RESTでサーバにアクセスしFHIRのデータを取得  
' 指定するurlを変更し、送信されるデータがどのように変わるか見てみましょう。  
url = str2  
restdata = Application.WorksheetFunction.WebService(url)  
  
' 取得したデータを表示する  
MsgBox (url + "に接続しました")  
MsgBox ("受信データ" + restdata)  
Sheet1.Cells(1, 1) = restdata  
  
End Sub
```

ここでURLを指定します

③Web APIからデータを引き出すときに使用する。
FHIRでなくとも例えばWikipediaでの検索結果などを表示させることも可能。

↓Mac版の代替プログラム

```
'RESTでサーバにアクセスしFHIRのデータを取得  
'指定するurlを変更し、送信されるデータがどのように変わるか見てみましょう。  
URL = str3  
  
Dim command As String  
command = "do shell script ""curl --get " & URL & """"  
  
restdata = MacScript(command)
```

ハンズオン 1 : サーバにRESTでアクセスする(4/5)

- WorksheetFunction オブジェクトでエラーが出る方は…

```
Sub RestConnectionformac()  
  
'変数の宣言  
Dim xmlHttp As Object 'HTTPリクエストオブジェクト  
Dim URL As String 'リクエストを送るURL  
Dim strParam As String 'リクエストパラメータ  
Dim restdata As String '結果を受け取る変数  
  
'接続先を取得  
Initialize  
  
'RESTでサーバにアクセスしFHIRのデータを取得  
'指定するurlを変更し、送信されるデータがどのように変わるか見てみましょう。  
URL = str3  
  
Dim command As String  
command = "do shell script ""curl --get " & URL & """"  
  
restdata = MacScript(command)  
  
'取得したデータを表示する  
MsgBox (URL + "に接続しました")  
MsgBox ("受信データ" + restdata)  
Sheet1.Cells(1, 1) = restdata  
  
End Sub
```

Excel Online, Excel 2016 for Macではこのオブジェクトは使用できない。
赤点線の部分は代替のプログラムで置き換えたもの。

ハンズオン 1 : サーバにRESTでアクセスする(5/5)

• RestConnectionプロシージャを実行する

```
' 取得したデータを表示する  
MsgBox (url + "に接続しました")  
MsgBox ("受信データ" + restdata)  
Sheet1.Cells(1, 1) = restdata  
  
End Sub
```

Microsoft Excel

http://hapi.fhir.org/baseDstu3/Patient?_pretty=trueに接続しました

OK

⑤実行するとURLに対応したデータが返されます。条件format="json"を入れるとjson形式になります。

※エラー処理はしていないので、インターネット接続がないとエラーが表示されます。

Microsoft Excel

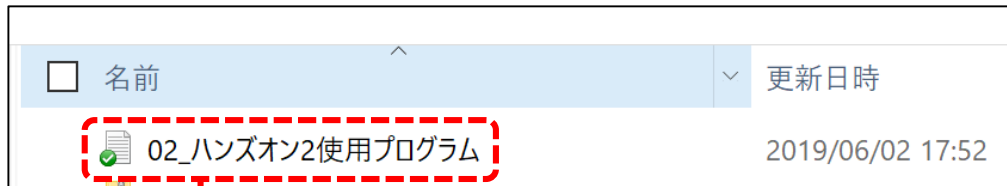
```
受信データ  
{  
  "resourceType": "Bundle",  
  "id": "22f6a359-a077-485e-9012-1337c2965628",  
  "meta": {  
    "lastUpdated": "2019-06-02T06:11:23.774+00:00"  
  },  
  "type": "searchset",  
  "link": [  
    {  
      "relation": "self",  
      "url": "http://hapi.fhir.org/baseDstu3/Patient?_pretty=true"  
    },  
    {  
      "relation": "next",  
      "url": "http://hapi.fhir.org/baseDstu3?_getpages=22f6a359-a077-485e-9012-1337c2965628&_getpagesoffset=20&_count=20&_pretty=true&_bundletype=searchset"  
    }  
  ],  
  "entry": [  
    {  
      "fullUrl": "http://hapi.fhir.org/baseDstu3/Patient/1627582",  
      "resource": {  
        "resourceType": "Patient",  
        "id": "1627582",  
        "meta": {  
          "versionId": "1",  
          "lastUpdated": "2019-03-10T16:24:06.679+00:00"  
        },  
        "text": {  
          "status": "generated",  
          "div": "<div xmlns='http://www.w3.org/1999/xhtml'> </div>"  
        },  
        "identifier": [  
          {  
            "use": "usual",  
            "type": {  
              "text": "Comp
```

A1

```
{  
  "resourceType": "Bundle",  
  "id": "22f6a359-a077-485e-9012-1337c2965628",  
  "meta": {  
    "lastUpdated": "2019-06-02T06:11:23.774+00:00"  
  },  
  "type": "searchset",  
  "link": [  
    {  
      "relation": "self",  
      "url": "http://hapi.fhir.org/baseDstu3/Patient?_pretty=true"  
    },  
    {  
      "relation": "next",  
      "url": "http://hapi.fhir.org/baseDstu3?_getpages=22f6a359-a077-485e-9012-1337c2965628&_getpagesoffset=20&_count=20&_pretty=true&_bundletype=searchset"  
    }  
  ],  
  "entry": [  
    {  
      "fullUrl": "http://hapi.fhir.org/baseDstu3/Patient/1627582",  
      "resource": {  
        "resourceType": "Patient"
```


ハンズオン2 : json形式のデータをパースして VBAのオブジェクトにする(1/4)

- Excel2016現在では、json形式を直接扱うワークシート関数がないので、GitHubで公開されている変換プログラムを使用します。



① 配布プログラムの「02_…」を開く

```
Windows | 02_ハンズオン2使用プログラム
下記のプログラムをEnd Subの上に追記します。↓
--- この下の行から ---↓
'JSONからVBAに変換↓
Dim jsonObj As Object↓
Set jsonObj = JsonConverter.ParseJson(restdata)↓
↓
'Patientリソースのデータを格納する↓
If url = str3 Then↓
↓
Sheet1.Cells(3, 1) = "(1)jsonから取り出した「Patient」の情報をセルに格納"↓
Sheet1.Cells(4, 1) = "id"↓
Sheet1.Cells(5, 1) = "name"↓
Sheet1.Cells(6, 1) = "gender"↓
Sheet1.Cells(7, 1) = "birthDate"↓
Sheet1.Cells(8, 1) = "address"↓
Sheet1.Cells(9, 1) = "phone number"↓
↓
Sheet1.Cells(4, 2) = jsonObj("id")↓
Sheet1.Cells(5, 2) = jsonObj("name")(1)("family") & " " & jsonObj("name")(1)("given")(1)↓
Sheet1.Cells(6, 2) = jsonObj("gender")↓
Sheet1.Cells(7, 2) = jsonObj("birthDate")↓
Sheet1.Cells(8, 2) = jsonObj("address")(1)("text")↓
Sheet1.Cells(9, 2) = jsonObj("telecom")(1)("value")↓
↓
End If↓
↓
'Observationリソースのデータを格納する↓
If url = str6 Then↓
```

先ほどインポートした「JsonConverter」と「Dictionary」のモジュールを使います

- Dictionary
→インポートしておくだけでOK
- JsonConverter
→VBAのプログラムに組み込みます。
FHIRリポジトリから取得したjson形式のデータを引数に入れて実行すると、VBAオブジェクトに変換されて戻ってきます。

ハンズオン 2 : json形式のデータをパースして VBAのオブジェクトにする(2/4)

• VBAオブジェクトでのリソースの書き方

【FHIRリソース】

8.1.2 Resource Content

Structure UML XML JSON Turtle R3 Diff All

Structure

Name	Flags	Card.	Type	Descri
Patient	N		DomainResource	Informa Elemen An ider
Identifier	Σ	0..*	Identifier	An ider
active	? Σ	0..1	boolean	Whethe
name	Σ	0..*	HumanName	A name
telecom	Σ	0..*	ContactPoint	A conta
gender		0..1	code	male Admini
birthDate	Σ	0..1	date	The da
deceased[x]	? Σ	0..1		Indicat
deceasedBoolean			boolean	
deceasedDateTime			dateTime	
address	Σ	0..*	Address	An add
maritalStatus		0..1	CodeableConcept	Marital Marital
multipleBirth[x]	? Σ	0..1		Whethe
multipleBirthBoolean			boolean	
multipleBirthInteger			integer	
photo		0..*	Attachment	Image
contact	1	0..*	BackboneElement	A conta + Rule
relationship		0..*	CodeableConcept	The kir Patient
name		0..1	HumanName	A name
telecom		0..*	ContactPoint	A conta

【jsonでの表現】

```
{
  "name": [
    {
      "use": "official",
      "family": "山田",
      "given": "太郎"
    },
    {
      "family": "ヤマダ",
      "given": "タロウ"
    }
  ],
  "telecom": [
    {
      "value": "0353216200"
    }
  ],
  "gender": "male",
  "birthDate": "1970-01-01",
  "address": [
    {
      "text": "東京都新宿区西新宿6丁目",
      "postalCode": "1600023"
    }
  ]
}
```

【VBAオブジェクトでの表現】

```
'Patientリソースのデータを格納する↓
If url = str3 Then↓
↓
Sheet1.Cells(3, 1) = "(1)jsonから取り出した「Patient」の情報をセルに格納"↓
Sheet1.Cells(4, 1) = "id"↓
Sheet1.Cells(5, 1) = "name"↓
Sheet1.Cells(6, 1) = "gender"↓
Sheet1.Cells(7, 1) = "birthDate"↓
Sheet1.Cells(8, 1) = "address"↓
Sheet1.Cells(9, 1) = "phone number"↓
↓
Sheet1.Cells(4, 2) = jsonObj("id")↓
Sheet1.Cells(5, 2) = jsonObj("name")(1)("family") & " " & jsonObj("name")(1)("given")↓
Sheet1.Cells(6, 2) = jsonObj("gender")↓
Sheet1.Cells(7, 2) = jsonObj("birthDate")↓
Sheet1.Cells(8, 2) = jsonObj("address")(1)("text")↓
Sheet1.Cells(9, 2) = jsonObj("telecom")(1)("value")↓
↓
End If↓
```

Point

項目名はどの形式でも変更ありません。リソースの基本構造のみおさえおけば、VBAでの記述方法もわかります。

ハンズオン 2 : json形式のデータをパースして VBAのオブジェクトにする(3/4)

• JsonConvertorを用いてJsonをオブジェクト形式に変換する

```
' RESTでサーバにアクセスしFHIRのデータを取得  
' 指定するurlを変更し、送信されるデータがどのように変わるか身  
url = str5  
  
restdata = Application.WorksheetFunction.WebService(url)  
  
' 取得したデータを表示する  
MsgBox (url + "に接続しました")  
MsgBox ("受信データ" + restdata)  
Sheet1.Cells(1, 1) = restdata  
  
End Sub
```

②テキストファイルのコードをここに
コピー&ペーストする

このプログラムでは下記の通り指定

- (1) Patientの場合 : url = str3
- (2) Observationの場合 : url = str5

```
ウインドウ 02_ハンズオン2使用プログラム  
下記のプログラムをEnd Subの上に追記します。↓  
↓  
--- この下の行から ---↓  
' JSONからVBAに変換↓  
Dim jsonObj As Object↓  
Set jsonObj = JsonConvertor.ParseJson(restdata)↓  
↓  
' Patientリソースのデータを格納する↓  
If url = str3 Then↓  
↓  
Sheet1.Cells(3, 1) = "(1)jsonから取り出した「Patient」の情報をセルに格納"↓  
Sheet1.Cells(4, 1) = "id"↓  
Sheet1.Cells(5, 1) = "name"↓  
Sheet1.Cells(6, 1) = "gender"↓  
Sheet1.Cells(7, 1) = "birthDate"↓  
Sheet1.Cells(8, 1) = "address"↓  
Sheet1.Cells(9, 1) = "phone number"↓  
↓  
Sheet1.Cells(4, 2) = jsonObj("id")↓  
Sheet1.Cells(5, 2) = jsonObj("name")(1)("family") & " " & jsonObj("name")(1)("given")(1)↓  
Sheet1.Cells(6, 2) = jsonObj("gender")↓  
Sheet1.Cells(7, 2) = jsonObj("birthDate")↓  
Sheet1.Cells(8, 2) = jsonObj("address")(1)("text")↓  
Sheet1.Cells(9, 2) = jsonObj("telecom")(1)("value")↓  
↓  
End If↓  
↓  
' Observationリソースのデータを格納する↓  
If url = str5 Then↓
```

Json→VBAオブ
ジェクトに変換

Patientリソースの
データをセルに格納

ハンズオン 2 : json形式のデータをパースして VBAのオブジェクトにする(4/4)

- 実行するとExcelのセルにリソースのデータが格納されます
※正しいURLを選択しないとエラーが出ます

	A	B	C	D	E	F	G	H	I
1	l								
2	"resourceType": "Patient",								
3	"id": "1953016",								
4	"meta": {								
5									
6									
7									
8									
9									

③ それぞれ正しく実行された場合の実行結果

(1)jsonから取り出した「Patient」の情報をセルに格納

id	1953016
name	山田 太郎
gender	male
birthDate	1970/1/1
address	東京都新宿区西新宿6丁目
phone number	03

(2)jsonから取り出した「Observation」の情報をセルに格納

id	1579810
value	36.9
unit	C
effectiveD	2019-01-30T18:55:12-07:00

【str3】 http://hapi.fhir.org/baseDstu3/Patient/1646554/_history/1?_format=json

【str5】 http://hapi.fhir.org/baseDstu3/Observation/1579810/_history/1?_format=json

Point

この階層はPatient IDではなく Observation IDなので注意

ハンズオン3 : 該当患者の複数回の検査結果を取得し、結果を表形式で入力する(1/3)

- Observationリソースのデータを患者番号で検索し、VBAオブジェクトに入れた後、Excelに表形式で出力します。

[str6] http://hapi.fhir.org/baseDstu3/Observation?_pretty=true&patient=1579762&_format=json

「_?」より後ろの部分は検索ないしは出力の条件となります

Point

```
- entry: [
  - {
    fullUrl: "http://hapi.fhir.org/baseDstu3/Observation/1579810",
    resource: {
      resourceType: "Observation",
      id: "1579810",
      meta: {
```

検索条件を入れてリクエストすると、検索結果はそれぞれ「entry」のツリーの下に並列で出力されます。

個々の検索結果は、検索条件を指定しないURLでも表示させることができます。

```
- entry: [
  - {
    fullUrl: "http://hapi.fhir.org/baseDstu3/Observation/1579810",
    resource: {
      resourceType: "Observation",
      id: "1579810",
      meta: {
        versionId: "1",
        lastUpdated: "2019-03-07T01:55:14.157+00:00"
      },
      text: {
        status: "generated",
        div: "<div xmlns='http://www.w3.org/1999/xhtml'>Body Temperature, 36.9 C</div>"
      },
      status: "final",
      code: {
        coding: [
          {
            system: "http://loinc.org",
            code: "8310-5"
          }
        ],
        text: "Body Temperature"
      },
      subject: {
        reference: "Patient/1579762"
      },
      effectiveDateTime: "2019-01-30T18:55:12-07:00",
      valueQuantity: {
        value: 36.9,
        unit: "C"
      }
    },
    search: {
      mode: "match"
    }
  },
  - {
    fullUrl: "http://hapi.fhir.org/baseDstu3/Observation/1579799",
    resource: {
      resourceType: "Observation",
      id: "1579799",
      meta: {
```

1つ目の検索結果
jsonObj("entry")("1")

2つ目の検索結果
jsonObj("entry")("2")

ハンズオン3 : 該当患者の複数回の検査結果を取得し、結果を表形式で入力する(2/3)

- jsonObj("entry").Countで、entryの数(=検索結果の数)を取得します。その後、Forループを使用してそれぞれの結果を出力します。

```
'Observationリソースのデータを格納する
If url = str5 Then

  Sheet1.Cells(3, 4) = "(2)jsonから取り出した「Observation」の情報をセルに格納"
  Sheet1.Cells(4, 4) = "id"
  Sheet1.Cells(5, 4) = "value"
  Sheet1.Cells(6, 4) = "unit"
  Sheet1.Cells(7, 4) = "effectiveDateTime"

  Sheet1.Cells(4, 5) = jsonObj("id")
  Sheet1.Cells(5, 5) = jsonObj("valueQuantity")("value")
  Sheet1.Cells(6, 5) = jsonObj("valueQuantity")("unit")
  Sheet1.Cells(7, 5) = jsonObj("effectiveDateTime")

End If
End Sub
```

②テキストファイルのコードをEnd Subの上にコピー&ペーストする

```
'RESTでサーバにアクセスしFHIRのデータを取得
'指定するurlを変更し、送信されるデータがどのように変わるか見てみましょう。
url = str6
```

③str=6と入力して実行します

名前	更新日時	種類	サイズ
02_ハンズオン2使用プログラム	2019/06/03 19:02	テキストドキュメント	2 KB
03_ハンズオン3使用プログラム	2019/06/03 20:15	テキストドキュメント	1 KB

①「03_ハン…」を開く

```
↓
(1)下記のプログラムをEnd Subの上に追記します。↓
--- この下の行から ---↓
'複数回のデータがある検査結果のデータを格納する↓
If url = str6 Then↓
↓
'Jsonオブジェクトのレコード数を確認する↓
Dim entryCount As Integer↓
entryCount = jsonObj("entry").Count↓
↓
'entryリソースデータを取り出し↓
Sheet2.Cells(1, 1) = "No."↓
Sheet2.Cells(1, 2) = "id"↓
Sheet2.Cells(1, 3) = "code"↓
Sheet2.Cells(1, 4) = "effectiveDateTime"↓
Sheet2.Cells(1, 5) = "valueQuantity"↓
Sheet2.Cells(1, 6) = "unit"↓

Dim i As Integer↓
For i = 1 To entryCount↓
  Sheet2.Cells(i + 1, 1) = i↓
  Sheet2.Cells(i + 1, 2) = jsonObj("entry")(i)("resource")("id")↓
  Sheet2.Cells(i + 1, 3) = jsonObj("entry")(i)("resource")("code")("text")↓
  Sheet2.Cells(i + 1, 4) = jsonObj("entry")(i)("resource")("effectiveDateTime")↓
  Sheet2.Cells(i + 1, 5) = jsonObj("entry")(i)("resource")("valueQuantity")("value")↓
  Sheet2.Cells(i + 1, 6) = jsonObj("entry")(i)("resource")("valueQuantity")("unit")↓
Next i↓
End If↓
--- この上の行まで ---↓
```

entryの数を取得

個々の検査結果を出力させる

ハンズオン3 : 該当患者の複数回の検査結果を取得し、結果を表形式で入力する(3/3)

- 実行するとSheet2に結果が表形式で出力されます。このデータさえ出力できれば、残りはVBAの機能でUIを整えることができます。

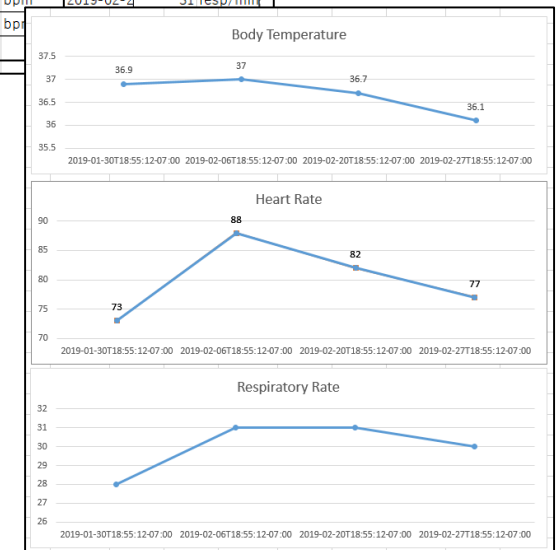
正しく実行された場合の実行結果

	id	code	effectiveDateTime	valueQuar	unit
1	1579810	Body Tem	2019-01-30T18:55:12-07:00	36.9	C
2	1579799	Weight	2019-02-20T18:55:12-07:00	70.8	Kg
3	1579798	Height	2019-02-20T18:55:12-07:00	90	cm
4	1579797	Respirator	2019-02-20T18:55:12-07:00	31	resp/min
5	1579796	Heart Rate	2019-02-20T18:55:12-07:00	82	bpm
6	1579814	Weight	2019-01-30T18:55:12-07:00	77.8	Kg
7	1579813	Height	2019-01-30T18:55:12-07:00	90	cm
8	1579812	Respirator	2019-01-30T18:55:12-07:00	28	resp/min
9	1579811	Heart Rate	2019-01-30T18:55:12-07:00	73	bpm
10	1579791	Heart Rate	2019-02-27T18:55:12-07:00	77	bpm
11	1579790	Body Tem	2019-02-27T18:55:12-07:00	36.1	C
12	1579795	Body Tem	2019-02-20T18:55:12-07:00	36.7	C
13	1579794	Weight	2019-02-27T18:55:12-07:00	73	Kg
14	1579793	Height	2019-02-27T18:55:12-07:00	90	cm
15	1579792	Respirator	2019-02-27T18:55:12-07:00	30	resp/min
16	1579807	Respirator	2019-02-06T18:55:12-07:00	31	resp/min
17	1579806	Heart Rate	2019-02-06T18:55:12-07:00	88	bpm
18	1579805	Body Tem	2019-02-06T18:55:12-07:00	37	C
19	1579804	Weight	2019-02-13T18:55:12-07:00	70.4	Kg
20	1579809	Weight	2019-02-06T18:55:12-07:00	77.7	Kg



検査結果説明用紙					
体温		心拍数		呼吸数	
検査日	数値	検査日	数値	検査日	数値
2019-01-30	36.9 C	2019-01-30	73 bpm	2019-01-30	28 resp/min
2019-02-06	37 C	2019-02-06	88 bpm	2019-02-06	31 resp/min
2019-02-20	36.7 C	2019-02-20	82 bpm	2019-02-20	31 resp/min
2019-02-27	36.1 C	2019-02-27	77 bpm	2019-02-27	30 resp/min

患者ID	1953016	患者番号を指定して基本情報を入力			
診療情報提供書					
(紹介先医療機関名)					
科	先生	平成31年6月20日 1日付が入ります			
所在地	名称	印			
電話番号	医師氏名				
患者氏名	山田 太郎	性別	male		
患者住所	東京都新宿区西新宿6丁目	電話番号	0353210000		
生年月日	昭和45年1月1日 (49歳)	職業			
1年齢が入ります					



各種様式作成、グラフの作成等でUIを整えるのはVBAが得意

選択ハンズオン : A or Bのお好きな方を進めて下さい

名前	更新日時	種類	サイズ
VBA-Dictionary-master	2019/06/06 15:15	ファイルフォルダー	
VBA-JSON-master	2019/06/06 15:15	ファイルフォルダー	
選択ハンズオンA	2019/06/04 13:25	ファイルフォルダー	
選択ハンズオンB	2019/06/04 22:09	ファイルフォルダー	
01_MTAチュートリアルFHIR-VBA	2019/06/05 19:19	Microsoft Excel マ...	46 KB
02_ハンズオン2使用プログラム	2019/06/10 10:22	テキストドキュメント	2 KB
03_ハンズオン3使用プログラム	2019/06/03 20:54	テキストドキュメント	1 KB
31_全プログラム初期	2019/06/13 13:40	テキストドキュメント	3 KB
32_全プログラム回答	2019/06/05 19:17	テキストドキュメント	10 KB
33_MTAチュートリアルFHIR-VBA(復習用)	2019/06/04 12:51	Microsoft Excel マ...	99 KB

名前	更新日時	種類	サイズ
11_ハンズオンAプログラム初期	2019/06/04 13:18	テキストドキュメント	1 KB
12_ヒント1 : URLに患者番号を埋め込む	2019/06/04 13:23	テキストドキュメント	1 KB
13_ヒント2 : 指定のセルにデータを入力	2019/06/04 13:24	テキストドキュメント	1 KB
14_ヒント3 : 生年月日から年齢を計算	2019/06/04 13:25	テキストドキュメント	1 KB
15_ハンズオンAプログラム回答	2019/06/04 13:25	テキストドキュメント	2 KB

- 「選択ハンズオンA」と「選択ハンズオンB」のフォルダには、ヒントとなるプログラムと回答があります。
- 困ったとき（時間がないとき）は、「ヒント」や「回答」のプログラムをコピーしてプロシージャを完成させて下さい。
- FHIRのリソースをVBAを使って特定の場所に表示させるハンズオンです。

Point

回答プログラムを入れて動作を確かめるだけでもVBAにおけるオブジェクトの扱い方を体感できます。

選択ハンズオンA：患者情報の自動入力

- Patientリソースから患者基本情報を取得し、診療情報提供書の必要事項を自動入力するプログラムを作ります。

ハンズオンA：患者情報の自動入力

患者ID	1953016	患者番号を指定して基本情報を入力	
診療情報提供書			
(紹介先医療機関名)			
科	先生	平成31年6月20日 ↑日付が入ります	
	所在地		
	名称		
	電話番号		
	医師氏名		印
患者氏名	山田 太郎	性別	male
患者住所	東京都新宿区西新宿6丁目	電話番号	0353210000
生年月日	昭和45年1月1日 (49歳)	職業	
	↑年齢が入ります		

- **Sheet3**と「**handsonA()**」のプロシージャを使用します。
- 赤枠のボタンを押すとhandsonAを実行するようにしてあります
- Sheet3.Cells(1,2)の患者IDを読み取り、**Patient**リソースからデータを取得
- 必要なデータをオブジェクトから抜き出してセルに入力して下さい。

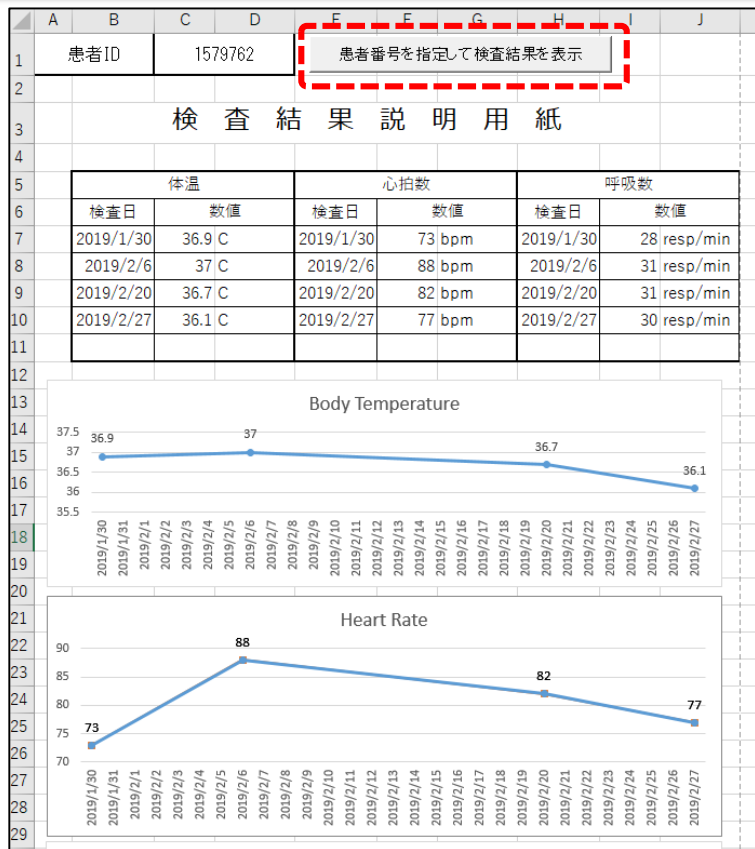
Point

RESTのURLは「str3」の形式です。年齢の計算にはVBAのDateDiff関数を使用すると簡単です。

選択ハンズオンB：検査結果説明用紙の作成

- ハンズオン3で取得したObservationリソースの検査結果をソートし、様式の結果表に自動入力するプログラムを作ります。

ハンズオンB：検査結果グラフ自動作成



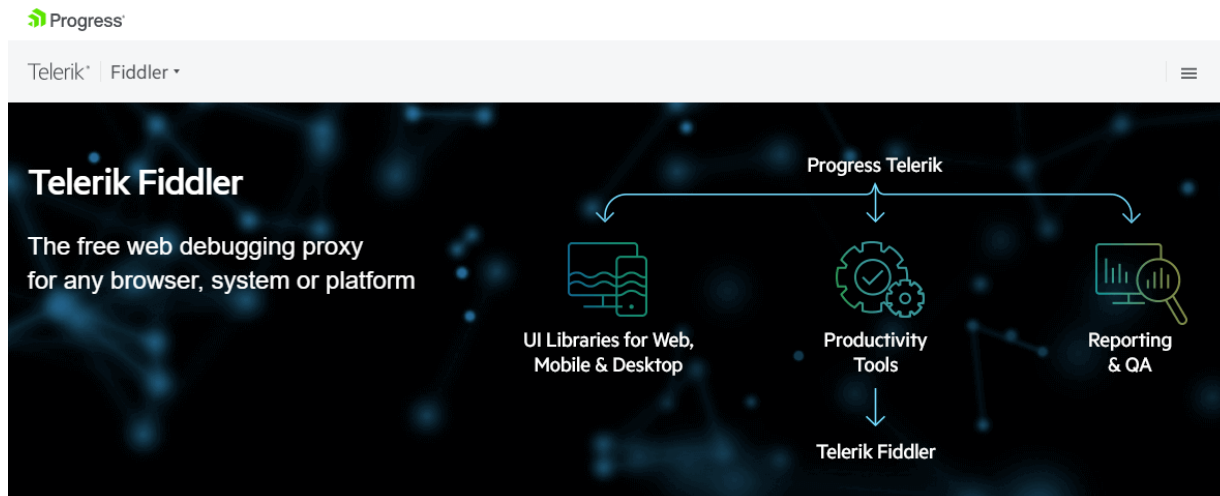
- Sheet4と「handsonB()」のプロシージャを使用します。
- 赤枠のボタンを押すとhandsonBを実行するようにしてあります
- Sheet3.Cells(1,3)の患者IDを読み取り、Observationリソースからデータを取得
- Sheet2の結果をコピーして作成する方式でもOKです。

Point

RESTのURLは「str6」の形式です。表形式にするためどの検査かをプログラムで識別する必要があります。

通信状況の確認をしたい方は

- <https://www.telerik.com/fiddler>
- Webサイトとの通信を可視化できるソフトウェアです



これでハンズオンは終了です

日本Mテクノロジー学会について



- (主に)医療データベース、プログラミング等に関連する領域の利用、応用、改良、及び普及を行うことを目的とした団体です。
 - 現代のMテクノロジーは関数型のプログラム、ツリー型とテーブル型の両方式のDBを統合できるオブジェクト指向型の開発環境です。
 - より良い医療システムアーキテクチャを探究しています。
- プログラミングやデータベースの技能・知識を持った情報部門担当者の育成を目的としたチュートリアルを年3回実施します
 - 学術部会（主に大学・病院関係）と技術部会（主にベンダ関係）が中心
 - 年次大会ではユーザとベンダのそれぞれの立場からの学術発表、技術討論、チュートリアル等を行い、会員のレベルアップを図っています。



第47回日本Mテクノロジー学会大会

in ポートアイランド（兵庫県神戸市）

【会期】 2019年9月13日(金)～14日(土)

【開催地】 兵庫県県立大学神戸情報科学キャンパス

【大会長】 兵庫県立大学応用情報科学研究科

教授 竹村 匡正 先生

本日ご紹介したオブジェクトベースプログラミングの
本格的なチュートリアル（半日）を実施します！！

多数の皆様のご参加を
お待ちしております！

