

# **FHIR環境構築 手順書**

2021年3月29日

第 1 . 0 版

## 目次

1. 本手順書について
2. システム構成
3. Hapi FHIRのセットアップ
  3. 1. Dockerのインストール
  3. 2. Hapi FHIR JPA Serverのインストール
4. Hapi FHIRへのデータ登録
  4. 1. FHIR Data Creatorの設定
  4. 2. FHIR Data Creatorの実行
  4. 3. 処理結果ログの確認
5. FHIR DASHBOARDのセットアップ
  5. 1. R Shiny Serverのインストール
  5. 2. FHIR DASHBOARDの適用
6. FHIR DASHBOARD使用方法
  6. 1. 初期画面
  6. 2. 検索結果一覧
  6. 3. 患者情報明細表示
7. FHIR Data Creator ソースコードの使用方法
  7. 1. FHIR Data Creator 処理の流れ
  7. 2. Visual C# プロジェクトの作成
  7. 3. 必要なパッケージのインストール
  7. 4. App.configの修正
  7. 5. データ移行元テーブルの作成

## 1. 本手順書について

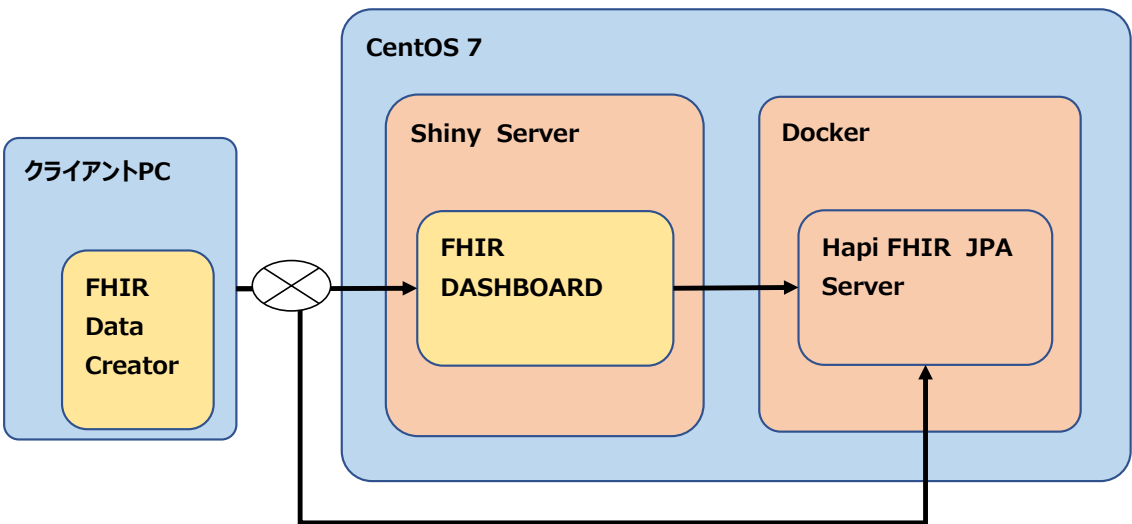
HL7 FHIRの検証環境を構築するための手順をご説明いたします。

※画面イメージについてはダミーデータを用いているため、実際のデータとは

データ数・画面に表示される値が異なりますのでご注意ください。

※各パッケージのインストールには、インターネット接続が必要です。

## 2. システム構成



Docker	<a href="https://www.docker.com/">https://www.docker.com/</a> 本手順書では、バージョン20.10.3を使用します。
Hapi FHIR	<a href="https://github.com/hapifhir/hapi-fhir-ipaserver-starter">https://github.com/hapifhir/hapi-fhir-ipaserver-starter</a> 本手順書では、バージョン5.2.0を使用します。
Shiny Server	<a href="https://rstudio.com/">https://rstudio.com/</a> 本手順書では、バージョン1.5.16.958を使用します。
FHIR Data Creator	所定のレイアウトで作成したデータベースから取得したデータを、 Hapi FHIRに登録するツールです。
FHIR DASHBOARD	FHIRに登録したデータを簡易的に参照するツールです。

### 3. Hapi FHIRのセットアップ

サーバにDockerをインストールし、Docker上でHapi FHIRサーバを動作可能にします。

#### 3. 1. Dockerのインストール

対象サーバに root 特権のあるユーザでログインします。

既存の yum パッケージを更新します。

```
# yum update  
  
# yum upgrade
```

公式の安定版 Yum リポジトリを設定します。

```
# yum install -y yum-utils device-mapper-persistent-data lvm2  
  
# yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
```

Docker をインストールします。

```
# yum makecache fast  
  
# yum install -y docker-ce
```

以下のコマンドを実行し、Dockerのバージョン情報が表示されればインストール完了です。

```
# docker version
```

Dockerを起動します。

```
# systemctl start docker
```

### 3. 2. Hapi FHIR JPA Serverのインストール

Hapi FHIRのDockerイメージをDockerHubより取得します。Dockerが起動している状態で実行します。

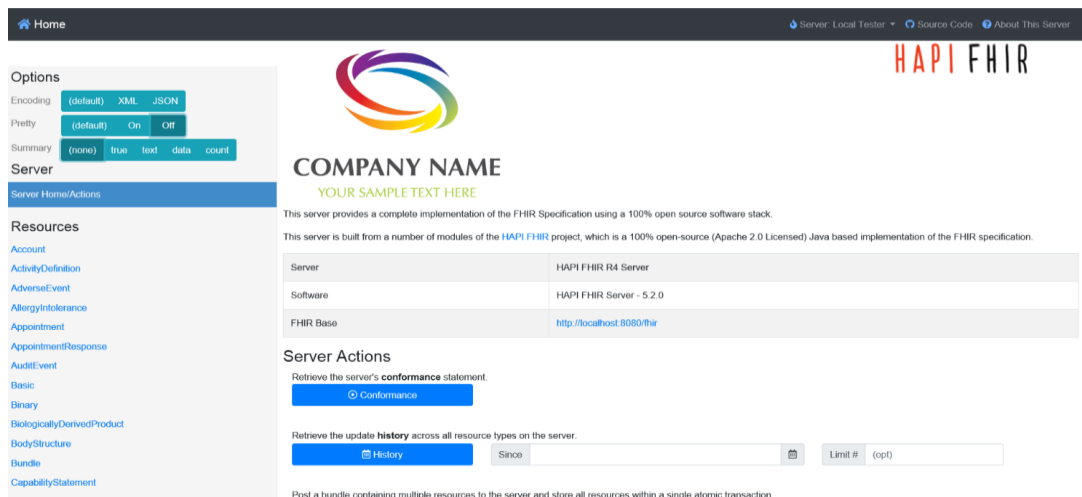
```
# docker pull hapiproject/hapi:latest
```

Hapi FHIRのコンテナを新規作成し、起動します。

```
# docker run -p 8080:8080 hapiproject/hapi:latest
```

Hapi FHIR起動後、ブラウザから「[http://\[ホストのIPアドレス\]:8080](http://[ホストのIPアドレス]:8080)」にアクセスします。

以下のような画面が表示されれば成功です。



The screenshot displays the HAPI FHIR web application interface. On the left, there is a sidebar with navigation links under 'Options' (Encoding, Pretty, Summary), 'Server' (Server Home/Actions), and 'Resources' (Account, ActivityDefinition, AdverseEvent, AllergyIntolerance, Appointment, AppointmentResponse, AuditEvent, Basic, Binary, BiologicallyDerivedProduct, BodyStructure, Bundle, CapabilityStatement, etc.). The main content area features the HAPI FHIR logo, a 'COMPANY NAME' placeholder, and a description of the server as a 100% open-source implementation of the FHIR specification. Below this, a table lists server details:

Property	Value
Server	HAPI FHIR R4 Server
Software	HAPI FHIR Server - 5.2.0
FHIR Base	<a href="http://localhost:8080/fhir">http://localhost:8080/fhir</a>

Below the table, the 'Server Actions' section includes buttons for 'Conformance' and 'History', along with input fields for 'Since' and 'Limit #'. The bottom of the page contains a link to 'Post a bundle containing multiple resources to the server and store all resources within a single atomic transaction.'

※Hapi FHIRのDocker コンテナを停止する場合の操作

以下のコマンドを実行すると、コンテナの一覧が表示されますので、停止するコンテナの名前を確認します。

```
# docker ps -a
```

※コンテナ一覧の例

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
123xxx456xx7	hapiproject/hapi:latest	"catalina.sh run"	1 hours ago	Up 1 hours	0.0.0.0:8080->8080/tcp	hapi_fhircontainer

コンテナ名

停止するコンテナ名を指定し、以下のコマンドを実行します。

```
# docker stop [停止するコンテナ名]
```

再度起動する場合は起動するコンテナ名を指定し、以下のコマンドを実行します。

```
# docker start [起動するコンテナ名]
```

## 4. Hapi FHIRへのデータ登録

FHIR Data Creatorファイル構成は以下のようになっています。

```
FhirDataCreator
├─ logs
├─ FhirDataCreator.exe
├─ FhirDataCreator.exe.config
├─ Hl7.Fhir.ElementModel.dll
├─ Hl7.Fhir.ElementModel.xml
├─ Hl7.Fhir.R4.Core.dll
├─ Hl7.Fhir.R4.Core.xml
├─ Hl7.Fhir.Serialization.dll
├─ Hl7.Fhir.Serialization.xml
├─ Hl7.Fhir.Support.dll
├─ Hl7.Fhir.Support.Poco.dll
├─ Hl7.Fhir.Support.Poco.xml
├─ Hl7.Fhir.Support.xml
├─ Hl7.FhirPath.dll
├─ Hl7.FhirPath.xml
├─ Newtonsoft.Json.dll
├─ Newtonsoft.Json.xml
├─ NLog.config
├─ NLog.dll
├─ NLog.xml
├─ System.ValueTuple.dll
└─ System.ValueTuple.xml
```

#### 4. 1. FHIR Data Creatorの設定

FhirDataCreator.exe.configをテキストエディタ等で編集します。

##### データベース接続設定

```
<add key="ODBC_SETTING"
value="Driver={PostgreSQL Unicode};
database=XXXXXX;
Server=xxx.x.x.xxx;
Port=5432;
Uid=XXXXXX;
Pwd=XXXXXX;
CommandTimeout=20;
Timeout=5"/>
```

..移行元のデータベース名を指定します。

..移行元データベースサーバのIPアドレスを指定します。

..データベースの接続ポートを指定します。

..データベースに接続するユーザー名を指定します。

..データベースに接続するパスワードを指定します。

##### Hapi FHIR接続設定

```
<add key="FHIR_BASE_URL"
value="http://[xxx.xxx.xxx.xxx]:8080/fhir"/>
```

..Hapi FhirサーバのURLを指定します。

#### 4. 2. FHIR Data Creatorの実行

手順4. 1の設定後、FhirDataCreator.exeをダブルクリックで実行します。

コンソール画面が表示され、処理状況が以下のように表示されます。

PROGRAM START

患者情報を登録しました。1件目 患者ID:1234 ID : 1



登録処理が完了すると、コンソール画面に「PROGRAM END」と表示されます。

何かボタンを押すと、処理を終了します。

```
PROGRAM START

患者情報を登録しました。1件目 患者ID:1234 ID : 1

:

患者情報を登録しました。100件目 患者ID:4321 ID : 7890

PROGRAM END
```

#### 4. 3. 処理結果ログの確認

登録処理実行時、FHIR Data Creator資源の「logs」フォルダに実行ログが出力されます。

登録処理完了後に実行ログを確認し、処理エラーが発生していないことを確認してください。

##### 実行ログの例

```
2021-03-20 10:30:26.0000 [INFO] ===== 処理を開始します。=====
2021-03-20 10:30:42.0000 [INFO] 患者情報を登録します。患者ID:12345
2021-03-20 10:30:55.0000 [INFO] 患者情報を登録しました。患者ID:12345 ID : 10
2021-03-20 10:30:55.0000 [INFO] 身長・体重を登録します。
2021-03-20 10:30:59.0000 [INFO] 身長・体重を登録しました。 2件
2021-03-20 10:30:59.0000 [INFO] 検体検査情報を登録します。
2021-03-20 10:31:10.0000 [INFO] 検体検査情報を登録しました。 13件
2021-03-20 10:31:10.0000 [INFO] 病名を登録します。
2021-03-20 10:31:13.0000 [INFO] 病名を登録しました。 3件
2021-03-20 10:31:13.0000 [INFO] 既往歴を登録します。
2021-03-20 10:31:15.0000 [INFO] 既往歴を登録しました。 3件
2021-03-20 10:31:15.0000 [INFO] レジメンを登録します。
2021-03-20 10:31:17.0000 [INFO] レジメンを登録しました。 1件
2021-03-20 10:31:17.0000 [INFO] 処方(注射)を登録します。
2021-03-20 10:31:18.0000 [INFO] 処方(注射)を登録しました。 2件
2021-03-20 10:31:18.0000 [INFO] 処方(処方)を登録します。
2021-03-20 10:31:21.0000 [INFO] 処方(処方)を登録しました。 4件
2021-03-20 10:31:21.0000 [INFO] ===== 処理を終了します。=====
```

## 5. FHIR DASHBOARDのセットアップ

R Shiny Serverをインストールし、Hapi FHIRに登録したデータを参照するツールを使用可能にします。

### 5. 1. R Shiny Serverのインストール

EPELをインストールし、サードパーティ製のパッケージをインストール可能にします。

```
# yum install epel-release
```

Rをインストールします。

```
# yum install R
```

FHIR DASHBOARDの動作に必要なRパッケージをインストールします。

•R Shiny

```
# R -e "install.packages('shiny', repos='http://cran.rstudio.com/')
```

•Shiny DashBoard

```
# R -e "install.packages('shinydashboard', repos='http://cran.rstudio.com/')
```

•JSONパーサー

```
# R -e "install.packages('jsonlite', repos='http://cran.rstudio.com/')
```

•DataTable

```
# R -e "install.packages('DT', repos='http://cran.rstudio.com/')
```

Shiny Serverをインストールします。

```
# wget https://download3.rstudio.org/centos7/x86_64/shiny-server-1.5.16.958-x86_64.rpm

# yum install --nogpgcheck shiny-server-1.5.16.958-x86_64.rpm
```

インストール完了後、Shiny Serverを起動します。

```
# systemctl start shiny-server
```

※Shiny Serverを停止する場合は、以下のコマンドを実行します。

```
# systemctl stop shiny-server
```

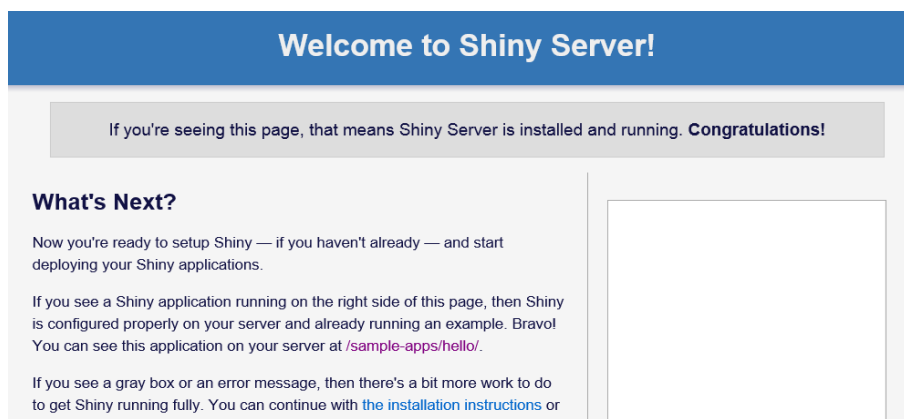
Shiny Serverで使用するポートを開放するため、Firewallの設定を変更します。

```
# firewall-cmd --add-port=3838/tcp --permanent

# firewall-cmd --reload
```

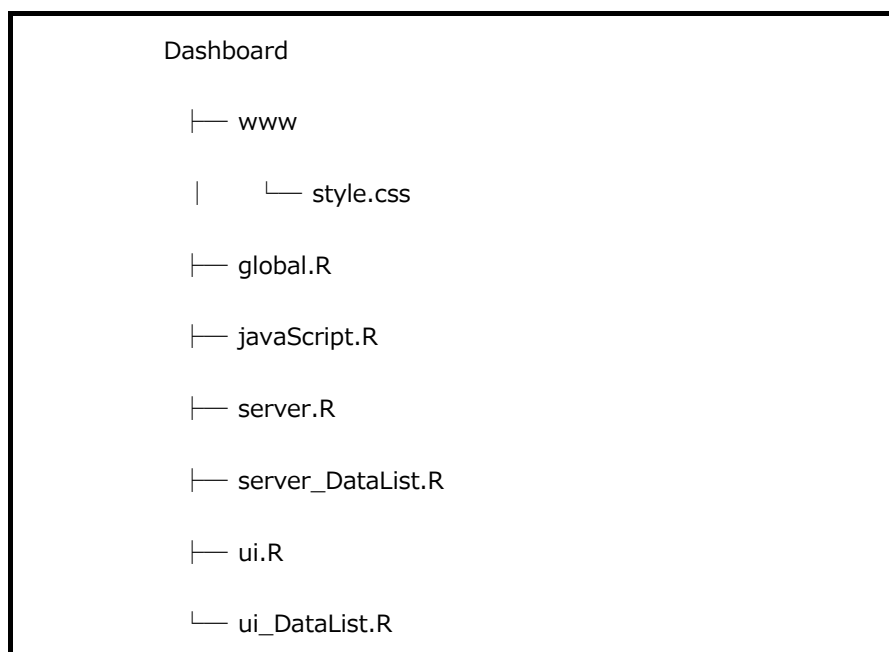
ブラウザから「[http://\[ホストのIPアドレス\]:3838](http://[ホストのIPアドレス]:3838)」にアクセスします。

以下のような画面が表示されれば成功です。



## 5. 2. FHIR DASHBOARDの適用

FHIR DASHBOARDのファイル構成は以下のようになっています。



FTPにて、FHIR DASHBOARDの資源「Dashboard」をディレクトリごとサーバの「/srv/shiny-server」に配置します。

「/srv/shiny-server/Dashboard/global.R」に記載されているURLを、以下のように修正します。

```
FHIR_URI <- "http://[ホストのIPアドレス]:8080/fhir"
```

ブラウザから「[http://\[ホストのIPアドレス\]:3838/Dashboard](http://[ホストのIPアドレス]:3838/Dashboard)」にアクセスします。

以下のような画面が表示されれば成功です。



## 6. FHIR DASHBOARD使用方法

### 6. 1. 初期画面

画面起動時の状態です。検索条件を指定し、「検索」ボタンをクリックすると、検索条件に一致する

患者情報を一覧表示します。

FHIR DASHBOARD

FHIR Data LIST

FHIR ID: [ ] Patient ID: [ ] Birth Date: [ ] to [ ] Display Count (1~1000): [100] [検索]

患者ID	性別	年齢	住所
1884356	male	1977-04-24	
1884351	male	1977-04-24	
1885825	male	1977-04-24	
1885835	male	1977-04-24	
1885840	male	1977-04-24	

FHIR ID	FHIRリソース「Patient」の「ID」を指定して検索する事ができます。
患者ID	施設のローカルIDを指定して検索する事ができます。
生年月日	患者の生年月日を範囲指定で検索する事ができます。
表示件数	検索結果に表示する件数を指定します。最大表示件数は1000件です。

### 6. 2. 検索結果一覧

検索条件に一致した患者情報を一覧表示します。

患者IDのリンクをクリックすると、選択した患者情報の明細を表示します。

FHIR DASHBOARD

FHIR Data LIST

FHIR ID: [ ] Patient ID: [1873908] Birth Date: [ ] to [ ] Display Count (1~1000): [100] [検索]

16件見つかりました。

ID	患者ID	性別	生年月日
1884356	1873908	male	1977-04-24
1884351	1873908	male	1977-04-24
1885825	1873908	male	1977-04-24
1885835	1873908	male	1977-04-24
1885840	1873908	male	1977-04-24

Previous 1 Next

### 6. 3. 患者情報明細表示

検索結果一覧で選択した患者情報の詳細を表示します。

The screenshot displays the 'FHIR DASHBOARD' interface. On the left is a sidebar with a 'FHIRデータリスト' menu. The main area is titled 'FHIR Data LIST' and contains a search filter section with fields for 'FHIR ID', '患者ID' (Patient ID), '生年月日' (Date of Birth), and '表示件数(1~1000)' (Number of items to display). Below the search filters, a message states '16件見つかりました。' (16 items found). A table lists patient records with columns for ID, 患者ID, 性別 (Gender), and 生年月日. The table shows five entries, all with 患者ID '573009' and 生年月日 '1977-04-24'. The last row is highlighted. Below the table, a detailed view of the selected patient is shown, enclosed in an orange border. This view includes a summary section with fields for 患者ID, 性別, 身長, 年齢, and 体重. Below this is a section for '患者プロフィール' (Patient Profile) with fields for 居住歴 (Residence History) and 居住歴 (Residence History). At the bottom, there is a section for 'レシメン' (Resimen) with fields for パス名 (Pass Name), パス有効期限日 (Pass Validity Date), and パス有効終了 (Pass Validity End). The interface includes navigation buttons like 'Previous' and 'Next' at the bottom of the detailed view.

ID	患者ID	性別	生年月日
1952454	573009	male	1977-04-24
1952485	573009	male	1977-04-24
1957889	573009	male	1977-04-24
1963186	573009	male	1977-04-24
1963286	573009	male	1977-04-24

患者ID: 573009  
性別: male  
身長: 170 cm  
年齢: 43  
体重: 60 kg

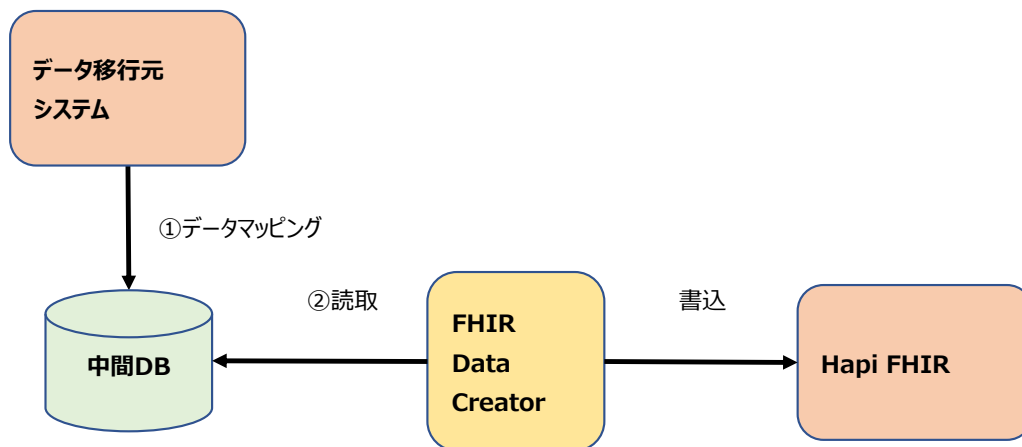
居住歴  
居住歴  
居住歴

パス名: リバーストパス  
パス有効期限日: 2018-11-10T12:00:00  
パス有効終了: 2018-11-10T12:00:00

## 7. FHIR Data Creator ソースコードの使用方法

本書の手順「4. Hapi FHIRへのデータ登録」で使った、FHIR Data Creatorをソースコードから使用する場合の手順を示します。

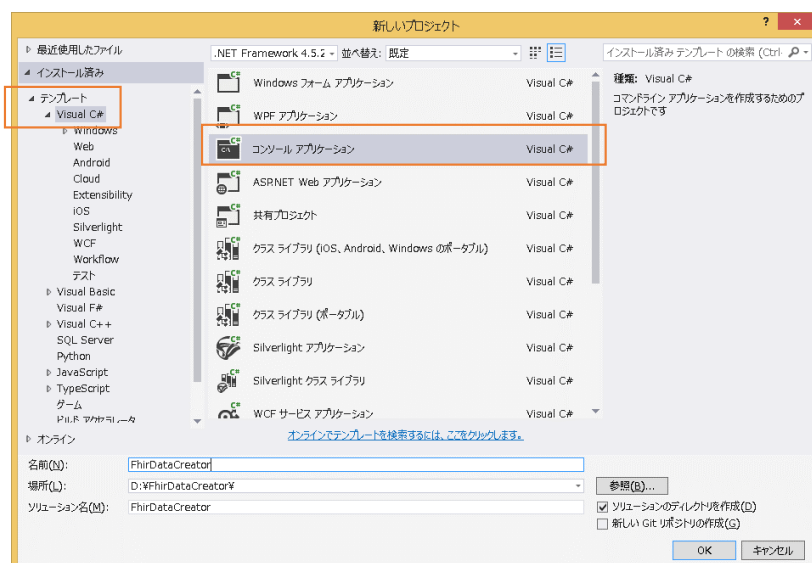
### 7. 1. FHIR Data Creator 処理の流れ



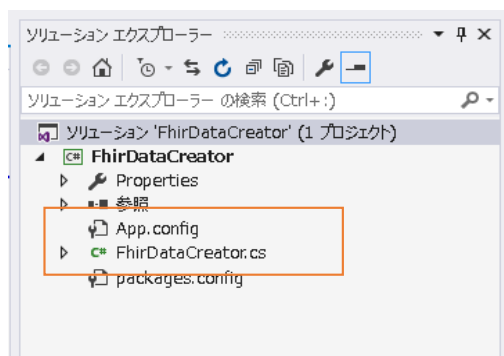
- ①あらかじめ、Hapi FHIRへのデータ移行元となるシステムから、手順「7. 5. データ移行元テーブルの作成」で作成するテーブルへデータ項目をマッピングし、データ投入します。
- ②FHIR Data Creatorは中間DBからデータを読み取り、FHIRのRESTful APIへ投入できる形に編集します。
- ③FHIR Data Creatorは②で編集したデータをHapi FHIRサーバへ送信し、FHIRリソースへ登録します。

## 7. 2. Visual C# プロジェクトの作成

Microsoft Visual Studio 2015にて、Visual C# コンソール アプリケーションのプロジェクトを作成します。



作成したプロジェクトに、配布の「App.config」、「FhirDataCreator.cs」を追加します。





### 7. 3. 必要なパッケージのインストール

NuGetパッケージマネージャーにて、以下のパッケージをインストールします。

HL7.Fhir.R4	バージョン3.0.0を使用します。
Nlog	バージョン4.7.9を使用します。
Nlog.Config	バージョン4.7.9を使用します。

### 7. 4. App.configの修正

本書の手順「4. 1. FHIR Data Creatorの設定」に従い、App.configを修正します。

### 7. 5. データ移行元テーブルの作成

配布の「公開用ddl.txt」を任意のDBサーバ上で実行し、中間DB用のテーブルを作成します。

移行元のシステムのデータを中間DBに投入後、FHIR Data Creatorを実行すると、

Hapi FHIRのリソース上に中間DBのデータが作成されます。