PDDI-CDSサービス　機能仕様書　附属書

インストール手順書

2024年1月30日

株式会社メタキューブ

目次

[1. 全体におけるPDDI-CDSサービスの位置 2](#_Toc157355211)

[2. インストール手順 2](#_Toc157355212)

[2.1. Dockerコンテナへのインストール 2](#_Toc157355213)

[2.1.1. 事前準備 2](#_Toc157355214)

[2.1.2. インストール手順 2](#_Toc157355215)

[2.2. ソースコードからのインストール 3](#_Toc157355216)

[2.2.1. 事前準備：使用するソフトウェアのインストール 4](#_Toc157355217)

[2.2.2. pddiCdsServiceのインストール 5](#_Toc157355218)

[2.3 CQL to ELM Translator インストール手順 6](#_Toc157355219)

図目次

[図 1　システム全体のイメージ 2](#_Toc157421093)

# 全体におけるPDDI-CDSサービスの位置

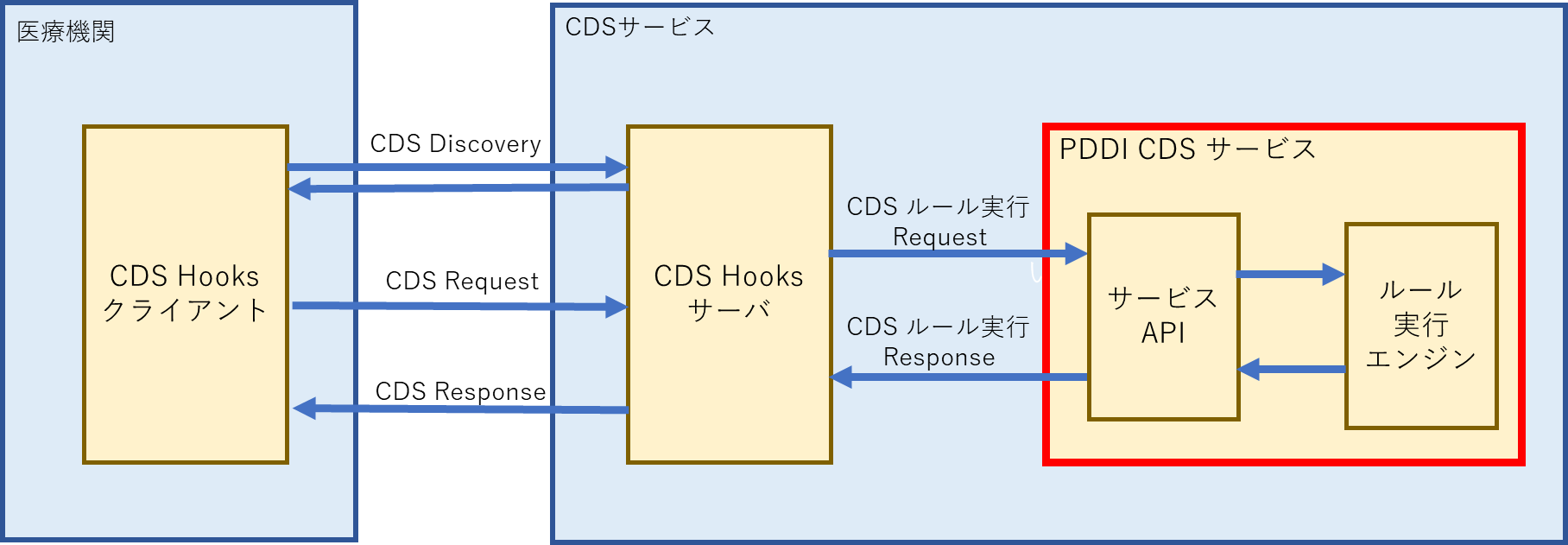


図 1　システム全体のイメージ

図 1にシステム全体のイメージを示す。このうちPDDI-CDSサービスは図中の赤枠で示す部分となり、本文書での取り扱い範囲もこの部分となる。

# インストール手順

　本PDDI-CDSサービスのインストール手順は

* Dockerのコンテナとして動かす場合（2.1参照）
* Windows上及びLinux上で直接動かす場合（2.2参照）

の2つの場合について説明する。

また本文書では、CQLルールをELMに変換するCQL to ELM Translatorのインストール手順についても2.3において説明する。

## Dockerコンテナへのインストール

### 事前準備

　あらかじめDocker（Windowsで動作するDocker Desktopなど）がインストールされ、利用可能な状態になっていること。 Dockerのインストール方法はWeb上にある解説記事を参照のこと。

### インストール手順

1. pddiCdsServiceの取得

GitHubのcds-hooks-proto

https://github.com/NeXEHRS/cds-hooks-proto/tree/main

からパッケージをダウンロードし、フォルダ「pddi-cds-service-proto」に含まれるフォルダ「pddiCdsService」とDockerFileを適切なフォルダへコピーする。以下コピー先のフォルダへのパスを<Folder>と記す。

1. pddiCdsServiceのサービス名設定

フォルダpddiCdsServiceに含まれるbuild.propertiesをテキストエディタで開き、その中の「servlet.name」を設定する。 なお、servlet.nameはCDS Hooks サーバに合わせて、あらかじめ「order-sign」に設定されている。そのままでよければ、build.propertiesは変更せずに次の手順3を実行する。

　　 ### Servlet Name

　　 servlet.name=order-sign

1. pddiCdsServiceのビルド実行

端末（Windowsの場合、PowerShellまたはコマンドプロンプト）を起動し、以下を実行する。なおWindowsのPowerShellやコマンドプロンプトの場合下記「$」を「>」に読み替える。以下も同様。

$ cd <Folder>

$ docker build -t pddi-cds-service ./

1. pddiCdsServiceのDockerコンテナ作成、起動

上記 3につづき、以下を実行する。これにより、pddiCdsServiceはDockerのコンテナとして起動する。pddiCdsServiceのAPIは1.3を参照のこと。

$ docker run -d -p 18080:8080 --name service1 pddi-cds-service:latest

1. 動作中コンテナの停止、再起動

* 上記 4 で起動したコンテナを停止する場合 下記のコマンドで再起動した場合も同じ

$ docker stop service1

* 停止したコンテナを再起動する場合

$ docker start service1

## ソースコードからのインストール

### 事前準備：使用するソフトウェアのインストール

　以下をあらかじめインストールしておくこと。java、node.js、apache ant、apache tomcatのインストール方法はここでは説明しない。Web上にある説明を参照のこと。

* Java: OpenJDK17

開発並びにCQL関係のオープンソースのビルドで使用する。Java（OpenJDK）は以下のサイトから取得した。なおJavaをインストールした後はJavaのbinフォルダにパスを通しておく。

https://adoptium.net/temurin/releases/?version=17&package=jdk

* node.js

https://nodejs.org/dist/v20.10.0/node-v20.10.0-linux-x64.tar.xz

* apache ant

https://dlcdn.apache.org//ant/binaries/apache-ant-1.10.14-bin.tar.gz

* apache tomcat

https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.83/bin/apache-tomcat-9.0.83.tar.gz

以下はCQLの解析に使用するオープンソースである。便宜上、インストール方法も簡単に説明する

* cql-execution

https://github.com/cqframework/cql-execution/archive/refs/tags/v3.0.1.zip

**インストール方法**

　上記URLから「cql-execution-3.0.1.zip」をダウンロードする。

　このzipファイルを任意のフォルダへコピーし、端末（PowerShell、コマンドプロンプト）上で以下のコマンドを実行する。このフォルダへの絶対パスを“<CQL-EXECUTION>”で表す。

$ cd <CQL-EXECUTION>

$ unzip cql-execution-3.0.1.zip

$ cd ./cql-execution-3.0.1

$ npm install

$ npm audit fix

* cql-fhir-exec

　https://github.com/cqframework/cql-exec-fhir/archive/refs/tags/v2.1.5.zip

**インストール方法**

　上記URLから「cql-exec-fhir-2.1.5.zip」をダウンロードする。

　このzipファイルを任意のフォルダへコピーし、端末（PowerShell、コマンドプロンプト）上で以下のコマンドを実行する。このフォルダへの絶対パスを“<CQL-EXEC-FHIR>”で表す。

$ cd <CQL-EXEC-FHIR>

$ unzip cql-exec-fhir-2.1.5.zip

$ cd ./cql-exec-fhir-2.1.5

$ npm install

$ npm audit fix

### pddiCdsServiceのインストール

1. GitHubのcds-hooks-proto

https://github.com/NeXEHRS/cds-hooks-proto/tree/main

からパッケージをダウンロードしたとき、フォルダ「pddi-cds-service-proto」に含まれるフォルダ「pddiCdsService」を適切なフォルダへコピーする。以下コピー先のフォルダへのパスを<pddi-cds-folder>と記す。

1. テキストエディタで<pddi-cds-folder>/pddiCdsService/build.propertiesを開き、編集する。

build.propertiesに含まれる以下の項目を編集する。

|  |  |
| --- | --- |
| build.propertiesパラメータ | 説明 |
| tomcat.home | Apache Tomcatのインストール先フォルダへのパス |
| servlet.name | PDDI-CDSサービスの名前。下記参照 |
| cql.execution.home | cql-executionのインストール先フォルダへのパス  <CQL-EXECUTION>/ cql-execution-3.0.1を指定する |
| cql.exec.fhir.home | cql-exec-fhirのインストール先フォルダへのパス  <CQL-EXEC-FHIR>/cql-exec-fhir-2.1.5を指定する |
| nodejs.exec.path | nodejs実行ファイルへの絶対パス |

ファイルbuild.propertiesで設定するパラメータのうち、servlet.nameはCDS Hooks サーバに合わせて、あらかじめ「order-sign」に設定されている。そのままでよければ、build.propertiesは変更しなくてもよい。

1. フォルダ<pddi-cds-folder>/pddiCdsServiceに「logs」と「works」の2個のフォルダを作成する。
2. ビルドを実行する

$ cd <pddi-cds-folder>/pddiCdsService

$ ant

1. Tomcatへデプロイする。

デプロイ先のTomcatが起動中ならば停止して、以下を実行する

$ ant deploy

デプロイ完了後、Tomcatを（再）起動する。

## 2.3 CQL to ELM Translator インストール手順

CQL to ELM Translatorは下記URLから取得できる「clinical\_quality\_language」に含まれている。

https://github.com/cqframework/clinical\_quality\_language/archive/refs/tags/v3.3.2.zip

**インストール方法**

　上記URLから「clinical\_quality\_language-3.3.2.zip」をダウンロードする。

　このzipファイルを任意のフォルダへコピーし、端末（PowerShell、コマンドプロンプト）上で以下のコマンドを実行する。このフォルダへの絶対パスを“<CQL-DIR>”で表す。

$ cd <CQL-DIR>

$ unzip clinical\_quality\_language-3.3.2.zip

$ cd ./clinical\_quality\_language-3.3.2/Src/java

$ ./gradlew build -x test

$ ./gradlew :cql-to-elm-cli:installDist

**CQL to ELM Translatorによる変換実行**

変換対象のCQLファイルへの絶対パスを<CQLdir>/cql-rule.cql、変換結果の出力先ファイルへの絶対パスを<ELMdir>/cql-rule2elm.jsonとした場合、以下を実行する。

$ cd <CQL-DIR>/clinical\_quality\_language-3.3.2/Src/java

$./cql-to-elm-cli/build/install/cql-to-elm-cli/bin/cql-to-elm-cli --format JSON --input <CQLdir>/cql-rule.cql --output <ELMdir>/cql-rule2elm.json

上記実行によりCQLルール「cql-rule.cql」は変換され、変換結果はフォルダ<ELMdir>にJSON形式のELMファイル「cql-rule2elm.json」に格納される。