**PKBFhirExtractor**

Contents

[Contents 1](#_Toc44346756)

[1. Introduction 2](#_Toc44346757)

[2. Scope 2](#_Toc44346758)

[3. PKBFhirExtractorApplication overview 2](#_Toc44346759)

[4. REST APIs 3](#_Toc44346760)

[5. Development Environment Configuration 4](#_Toc44346761)

[7. References 6](#_Toc44346762)

# Introduction

PKB is a personal Healthcare application. It is a social enterprise and technology platform, designed to bring together patient data from health and social care providers and the patient's own data, into one secure personal health record. The purpose of this document is to outline the design, technical information and deployment steps for PKBFhirExtractor project which uses PKB endpoints to push the Patient data.

# Scope

The PKBFHIRExtractor application sends all Patient data in FHIR bundle to PKB end points. The application uses Create or Update PKB endpoints based on the Patient data validation.

# PKBFhirExtractorApplication overview

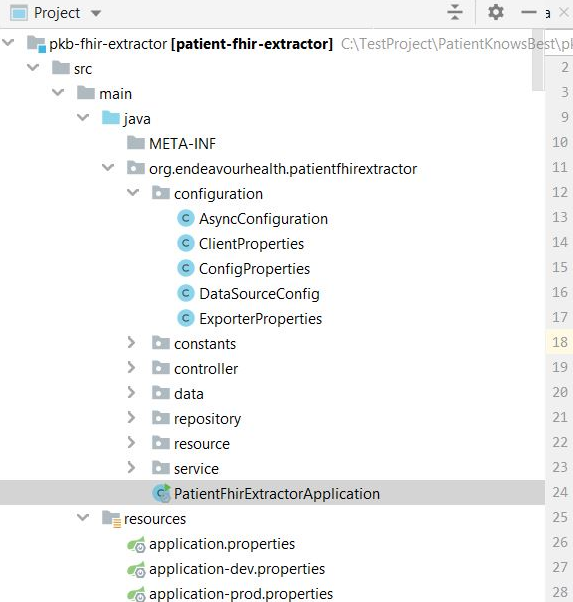
PKBFhirExtractorApplication is developed using Spring Boot framework with Java 11 and Maven. Spring Boot used to create stand alone application providing a range of non functional features like embedded servers (Tomcat used by default), security, metrics, profiling, health-check and externalized configuration.

Asynchronous execution support from Spring has been used for patient processing. This makes the execution in a separate thread and the caller need not wait for the completion of the caller method.

Spring boot allows defining profile specific property files in the form of application-{profile}.properties. It automatically loads the properties in an application-properties file for all profiles, and the ones in profile-specific property files only for the specified profile.

Code organize as this

* **configuration** is the layer for adding configuration details for client, datasource, async properties etc.
* **constants** is the folder where the constant class is stored
* **controller** is the controller class for REST api calls.
* **data** is the business model for entities
* **resource** - folder for implementing FHIR Patient related resources.
* **service** - folder for high level services for queries with data transfer objects.



# REST APIs

Application supports the following REST API’s

1. GET /start?queue=queueId

This API is used to start the patient’s processing for the queueId mentioned.

Return:

HttpStatus code 200 if processing is successful

HttpStatus code 500 for internal server errors.

1. GET /stop

This API is used to stop the patient processing in between.

Return:

HttpStatus code 200 if processing is successful

1. GET /procedure/cohort/start

This API is used to run the cohort procedures

Return:

HttpStatus code 200 if processing is successful

HttpStatus code 500 for internal server errors.

1. GET /procedure/delta/start

This API is used to run the delta procedures

Return:

HttpStatus code 200 if processing is successful

HttpStatus code 500 for internal server errors.

# Development Environment Configuration

1. Install MySQL and Intellij.

2. In MySQL create a new schema called config. We need to create a new table called config inside config schema.

CREATE TABLE `config` (

`app\_id` varchar(100) NOT NULL,

`config\_id` varchar(100) NOT NULL,

`config\_data` text NOT NULL,

PRIMARY KEY (`app\_id`,`config\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1

Insert the below data inside config table:

INSERT INTO `config`.`config` (`app\_id`, `config\_id`, `config\_data`) VALUES 'patientknowsbest', 'database', '{"url": "jdbc:mysql://localhost:3306/pkb\_extracts?useSSL=false", "password": “password", "username": "root"')

Change the username and password.

3. subscriber\_ui

Unzip subscriber\_UI and run the SQL script.

4. data\_extracts\_pkb:

Create data\_extracts\_pkb schema and create the sql files mentioned in the below URL.(Note: Dont need to run the sql files) <https://github.com/endeavourhealth/data_extracts/tree/master/PatientKnowsBest(PKB)>

5. Clone <https://github.com/endeavourhealth-discovery/PatientKnowsBest> from GIT.

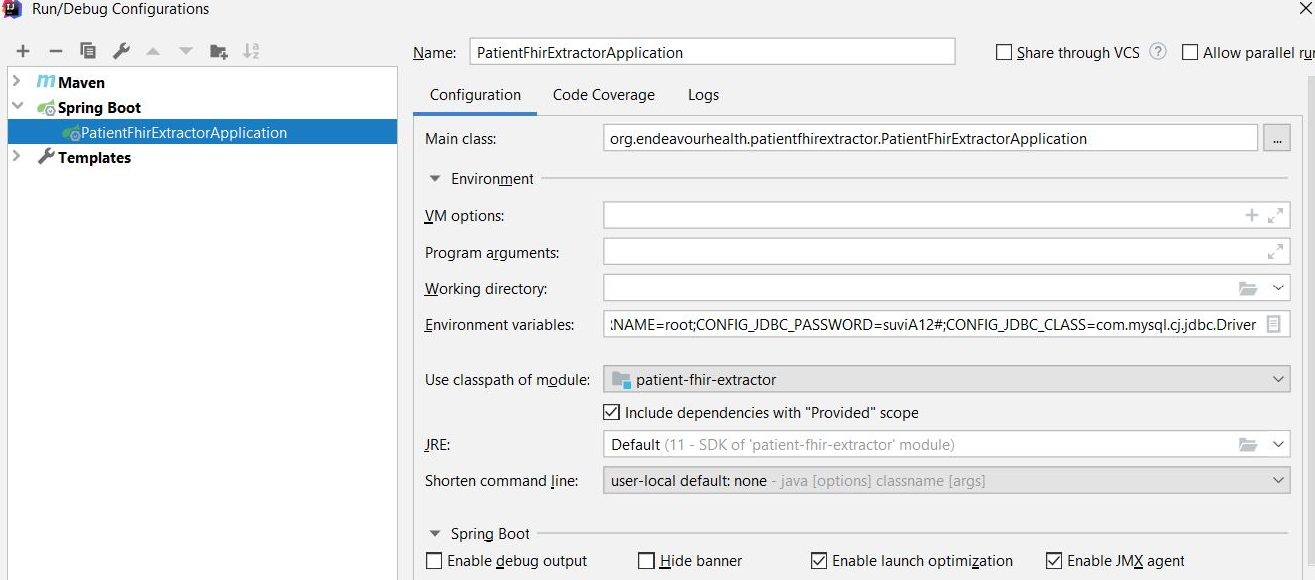
6. Launch Intellij and open pkb-fhir-extractor project. (Inside PatientKnowsBest)

Go to src/main/resources → application-dev.properties and fill the following details with appropriate values. (These values are used for connecting to PKB endpoint)

config.clientID=  
config.clientSecret=  
config.scope=  
config.tokenUrl=  
config.baseUrl=

8. Run PatientFhirExtractorApplication with the following environment properties.

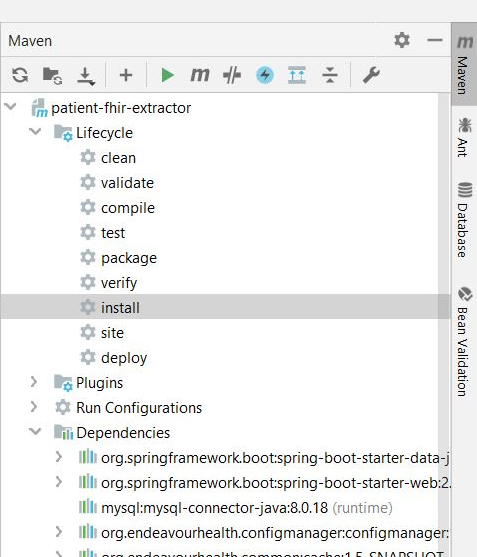
CONFIG\_JDBC\_URL=jdbc:mysql://localhost:3306/config?useSSL=false;CONFIG\_JDBC\_USERNAME=root;CONFIG\_JDBC\_PASSWORD=1qaz1qaz;CONFIG\_JDBC\_CLASS=com.mysql.cj.jdbc.Driver



9. Compile the code and run

6.DEV box deployment

Get the jar file for PKBFhirExtractor spring application by running Maven install.

****

Export the following variables to the DEV box.

export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64 (Java home path)  
export CONFIG\_JDBC\_USERNAME=(username)   
export CONFIG\_JDBC\_PASSWORD=(password)  
export CONFIG\_JDBC\_URL="jdbc:mysql://[hl7transform.csjxcq8rzerp.eu-west-2.rds.amazonaws.com:3306/config?useSSL=false](http://hl7transform.csjxcq8rzerp.eu-west-2.rds.amazonaws.com:3306/config?useSSL=false)" (Config path)  
export CONFIG\_JDBC\_CLASS=com.mysql.cj.jdbc.Driver

execute : nohup java jar patient-fhir-extractor.jar

This will start the spring boot application with tomcat.

# 7. References

http://dev.patientsknowbest.com/home/fhir-api/fhir-api-roadmap/operations/process-message