# K-HIS FHIR 교육 FHIR Client 구현 실습

Health All Inc. 최미현

#### HAPI FHIR

- HL7 Application Programming Interface
- 오픈 소스 Java API
- HL7과 연계하여 FHIR 표준에 맞춘 프로그램 개발
- HAPI FHIR 기반 Public Test Server 운영 및 Server 프로젝트 공개
- Github를 통해 그 밖의 HAPI FHIR 기반 다양한 프로젝트 공개

## 참고

- HAPI FHIR Document
  - https://hapifhir.io/hapi-fhir/docs/client/introduction.html
- HAPI FHIR Client start
  - https://github.com/FirelyTeam/fhirstarters/tree/master/java/hapi-fhirstarters-client-skeleton/
     n/
- KR Core IG
  - http://www.hl7korea.or.kr/fhir/krcore/



Database Schema

Configuration

Performance

Upgrade Guide

Diff Operation

Search

Bundle Builder CLIENT Introduction Get Started 4 4.1 4.2 Generic (Fluent) Client Annotation Client 4.3 Client Configuration 4.4 Client Examples 4.5 PLAIN SERVER **REST Server Types** 5.0 51 Plain Server Introduction 5.2 Get Started 6 5.3 Resource Providers and Plain Providers **REST Operations: Overview** 5.4 **REST Operations: Search** 5.5 **REST Operations: Extended Operations** 5.6 Paging Search Results 5.7 5.8 Web Testpage Overlay Multitenancy 5.9 JAX-RS Support 5.10 Customizing the CapabilityStatement 5.11 OpenAPI / Swagger 5.12 JPA SERVER Introduction 6.0 Get Started 6 6.1 Architecture 6.2 6.3 Database Support

4.0 Introduction ▼

4.1 Get Started ♦ →

#### 4.0.1 Client Introduction

HAPI FHIR provides a built-in mechanism for connecting to FHIR REST servers.

The HAPI RESTful client is designed to be easy to set up and to allow strong compile-time type checking wherever possible.

There are two types of REST clients provided by HAPI:

- The Generic (Fluent) client is designed to be very flexible, yet easy to use. It allows you to build up FHIR REST invocations using a fluent API that is consistent and powerful. If you are getting started with HAPI FHIR and aren't sure which client to use, this is the client to start with.
- The Annotation Client client relies on static binding to specific operations to give better compile-time checking against servers with a specific set of capabilities exposed. This second model takes more effort to use, but can be useful if the person defining the specific methods to invoke is not the same person who is using those methods.

4.0 Introduction \*

4.1 Get Started ∳ →





64

6.5

6.6

6.7

6.8

6.9

НАРІ НАРІ	USING HAPI	JAVADOCS
Home	Documentation	Core API
Download	Command Line Tool (cli)	Model API (DSTU2)
Changelog		Model API (DSTU3)
License	STARTER PROJECTS	Model API (R4)
News	Client	Model API (R5)
	Plain Server	Client API
TEST SERVER	IDA Sorver	DI-1- 0 ADI

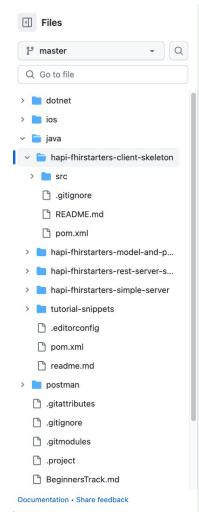
GET HELP

Google Group (Ask Questions) Issue Tracker (Report Bugs/Request Features) FHIR.org Chat (Live Chat with FHIR

How to Ask for Help (read this first!)

Implementors)

Frequently Asked Questions (FAQ)



fhirstarters / java / hapi-fhirstarters-client-skeleton / ↑ Top  $\equiv$ README.md HAPI FHIR - Client Starter @ This project is a simple "skeleton project" (a Maven project containing all dependencies needed to run the HAPI FHIR client). Instructions: Import this project into your IDE • Try executing the class TestApplication.java You will see that this project loads the resource Patient/example from the R4 testing endpoint hosted at http://hapi.fhir.org/baseR4 Let's Build - Introduction @ Check out the Client Documentation to see the syntax for various FHIR client operations. Here are some things to try: Perform a FHIR Search for Patient resources where the patient has a name of "Smith". Print the resource ID for each matching resource Use a FHIR Create operation to create a new Patient resource on the server. Give it your name, or a fictional name you make up. Log the ID that the server assigned to this resource Use a FHIR Read to read it back Use a FHIR Update to update it Use a FHIR Delete to delete it

Try modifying your Read code so that it displays a useful error if the resource has been deleted



#### **KR Core Implementation Guide**

1.0.1 - STU1 (1)



Guidance ▼ FHIR Artifacts ▼

Examples Downloads Change Log

Table of Contents > KR Core Implementation Guide (IG)

This page is part of the KR Core (v1.0.1 - STU1) based on FHIR R4 . This is an authorized publication. For a full list of available versions, see the Directory of published versions

#### KR Core Implementation Guide (IG)

Official URL: http://www.hl7korea.or.kr/fhir/krcore/ImplementationGuide/hl7.fhir.kr.core

Version: 1.0.1

Active as of 2023-06-30

Computable Name: KR Core IG

#### 1.1 소개

KR Core는 국내 보건의료데이터 교류의 상호운용성과 데이터의 질을 보장하기 위한 한국형 FHIR IG(상세규격)으로, HL7 FHIR R4 년를 기반으로 하여 국 내 보건의료데이터 교류에 사용하는 FHIR 리소스 및 FHIR RESTful API에 대한 최소한의 제약조건을 정의한다.

KR Core는 KR CDI(한국형 핵심교류데이터)의 요구사항에 따라 국내 보건의료데이터 교류 시스템이 갖추어야 할 공통 요구사항을 정의함으로써, 시스템간 상호운용성을 확보하고 데이터를 유의미하게 활용할 수 있는 토대를 마련한다.

이를 통해 국내 보건의료데이터 교류를 촉진하고 특정 사용 사례에 대한 추가 제약조건 개발을 용이하게 한다.

#### Contents:

- 소개
- 배경
- 문서의 내용 및 안내

#### 1.2 배경

세계 각 국가들은 국가차원의 FHIR IG를 개발하여 자국의 의료정보교류 상호운용성 확보와 더불어 의료데이터의 질을 높이기 위해 노력하고 있다. 이러한 FHIR IG의 대표적인 예로는 US Core(미국), AU Base(호주), UK Core(영국), CH Core(스위스) 등이 있고, 이 외에도 캐나다, 이탈리아, 인도 등 많은 나라들이 국가수준의 FHIR IG를 개발 중이다.

한국에서도 한국보건의료정보원 [강과 HL7 Korea [강의 주관으로 KR Core를 개발 중이며 목적은 다음과 같다.

- 국내 보건의료 환경에 맞는 FHIR IG를 개발하여 국내 보건의료데이터 교류 생태계 기반을 마련
- 국내 의료정보교류 상호운용성 확보와 의료데이터 질 향상
- 특정 사용 사례에 대한 기반 정보모델 제공
- SMART on FHIR, Blue Button 2.0 등과 같은 다양한 헬스케어 서비스 개발의 활성화

#### 1.3 문서의 내용 및 안내

KR Core는 한국에서 사용하는 FHIR 리소스 및 FHIR RESTful API 등을 정의하고 이에 대한 상세한 설명을 문서화하여 제공한다.

KR Core를 구성하는 문서의 목록과 각 문서의 내용은 다음과 같다.

- IG Home: 본 페이지. KR Core에 대한 소개 및 개발 배경 등에 관한 정보 제공.
- Conformance: KR Core 적합성을 준수하기 위한 규칙들을 제공.

### Client 실습 전 준비 1

- 1. JDK 17 설치
  - https://www.oracle.com/kr/java/technologies/downloads/
  - 'java -version' 으로 버전이 17임을 확인
- 2. SpringToolSuite 설치
  - https://spring.io/tools

## Jdk 17버전 Mac OS 설치

**JDK 21** 

JDK 17

GraalVM for JDK 21

GraalVM for JDK 17

#### JDK Development Kit 17.0.9 downloads

JDK 17 binaries are free to use in production and free to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions (NFTC).

JDK 17 will receive updates under the NFTC, until September 2024. Subsequent JDK 17 updates will be licensed under the Java SE OTN License (OTN) and production use beyond the limited free grants of the OTN license will require a fee.

Linux macOS Windows			
Product/file description	File size	Download	
ARM64 Compressed Archive	168.18 MB	https://download.oracle.com/java/17/latest/jdk-17_macos-aarch64_bin.tar.gz ( sha256)	
ARM64 DMG Installer	167.60 MB	https://download.oracle.com/java/17/latest/jdk-17_macos-aarch64_bin.dmg ( sha256)	M1,M2
x64 Compressed Archive	170.62 MB	https://download.oracle.com/java/17/latest/jdk-17_macos-x64_bin.tar.gz ( sha256)	
x64 DMG Installer	170.03 MB	https://download.oracle.com/java/17/latest/jdk-17_macos-x64_bin.dmg ( sha256)	인텔 기반

# Spring Tools 4 for Eclipse

The all-new Spring Tool Suite 4. Free. Open source.

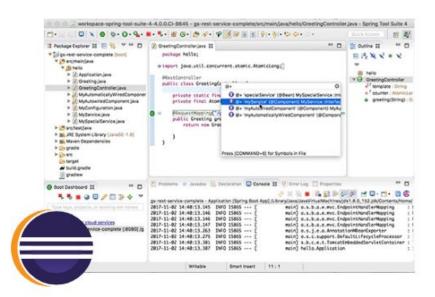
4.20.1 - LINUX X86\_64

4.20.1 - LINUX ARM\_64

4.20.1 - MACOS X86\_64

4.20.1 - MACOS ARM 64

4.20.1 - WINDOWS X86\_64



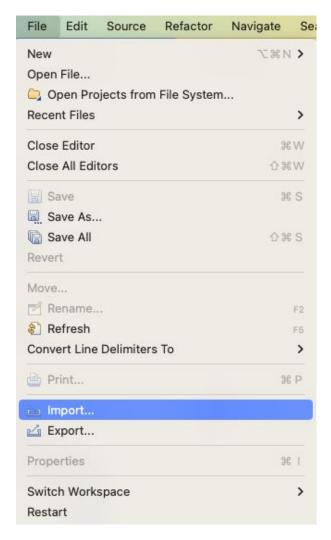
인텔 기반 Mac OS

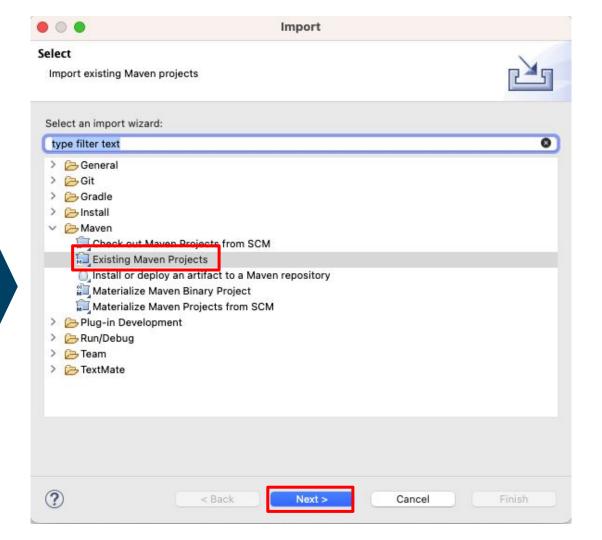
M1,M2 Mac OS

Windows x86

#### Client 실습 전 준비 2

- 실습 자료 압축 파일 해제
- 2. SpringToolSuite에 압축 해제한 파일 import
  - a. 메뉴 > File > Import...
  - b. Maven > Existing Maven Projects
  - c. Browse.. > 압축 해제한 폴더 선택 > open
  - d. pom.xml 체크 > Finish





### pom.xml

```
<!-- This dependency includes the core HAPI-FHIR classes -->
<dependency>
    <groupId>ca.uhn.hapi.fhir</groupId>
   <artifactId>hapi-fhir-base</artifactId>
    <version>${hapifhir_version}</version>
</dependency>
<!-- Include the client -->
<dependency>
    <groupId>ca.uhn.hapi.fhir</groupId>
    <artifactId>hapi-fhir-client</artifactId>
    <version>${hapifhir_version}</version>
</dependency>
<!-- At least one "structures" JAR must also be included -->
<dependency>
    <groupId>ca.uhn.hapi.fhir</groupId>
    <artifactId>hapi-fhir-structures-r4</artifactId>
    <version>${hapifhir_version}</version>
</dependency>
```

# 실습순서

- 1. 리소스 생성
- 2. FHIR 서버에 리소스 등록 (Create / Update)
- 3. 등록한 리소스 조회 (Read / Search)

### 리소스 생성

- 간략한 버전의 Patient 리소스
  - 이번 실습을 위해 KR Core Patient 리소스를 간략하게 재구성
  - ㅇ 출처
    - KR Core IG Patient 리소스
    - http://www.hl7korea.or.kr/fhir/krcore/Patient-krcore-patient-example-01.xml.html
- 간략한 버전의 Observation 리소스
  - 이번 실습을 위해 KR Core Patient 리소스를 간략하게 재구성
  - 출처
    - KR Core IG Observation 리소스 (KR Core Observation Profile for Function Test)
    - http://www.hl7korea.or.kr/fhir/krcore/Observation-krcore-observation-function-test-ex ample-01.xml.html

#### 1. Patient 리소스 생성 - Patient element

- 환자 개인에 대한 인적 사항을 표현한 리소스.
- 관할 지역에 따라 확장된 프로파일을 사용할 수 있음.
- 속성
  - identifier
  - name
  - o telecom
  - o gender
  - birthdate
  - address
  - o contact

# 1. Patient 리소스 생성 - Patient element

- 생성 후 등록되는 element
  - o id
  - o meta
  - text

### 1. Patient 리소스 생성

- HAPI FHIR 라이브러리를 활용하여 Patient 리소스를 생성하는 방법
  - o Patient 인스턴스 생성
  - 생성하고자 하는 요소에 대한 인스턴스 생성
  - 요소 인스턴스에 값 입력
  - Patient 인스턴스에 요소를 set
- 테스트할 Patient 리소스 예제 (다음 페이지)

```
<Patient xmlns="http://hl7.org/fhir">
     <meta> rofile value="http://www.hl7korea.or.kr/fhir/krcore/StructureDefinition/krcore-patient"/> </meta>
     <identifier>
           <type> <coding> <system value="http://terminology.hl7.org/CodeSystem/v2-0203"/> <code value="MR"/> </coding> </type>
           <system value="urn:oid:1.2.3.4.5.6"/>
           <value value="PID12345"/>
     </identifier>
     <name> <text value="박아품"/> </name>
     <telecom> <system value="phone"/> <value value="010-1234-5678"/> </telecom>
     <telecom> <system value="email"/> <value value="patientKim@example.co.kr"/> </telecom>
     <qender value="male"></qender>
     <br/>

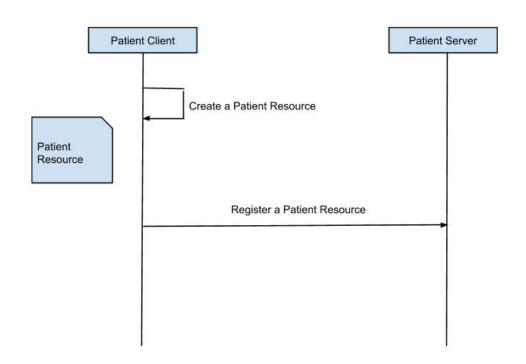
     <address>
           <extension url="http://www.hl7korea.or.kr/fhir/krcore/StructureDefinition/krcore-roadNameAddress">
                  <extension url="district"> <valueString value="서울특별시 강남구"/> </extension>
                  <extension url="roadName"> <valueString value="일원로"/> </extension>
                  <extension url="detailedAddress"> <valueString value="81"/> </extension>
                  <extension url="complements"> <valueString value="일원동"/> </extension>
           </extension>
           <text value="서울특별시 강남구 일원로 81 (일원동)"/>
           <postalCode value="06351"/>
                                                                                                                                                                                                                                                        출처:
     </address>
                                                                                                                                                                                                                                                        http://www.hl7korea.or.kr/fhir/krcore/Patient-krcore-
</Patient>
                                                                                                                                                                                                                                                        patient-example-01.xml.html
```

## 2. FHIR 서버에 리소스 등록

- FHIR 서버에 Patient 리소스 등록 (create / update)
  - 생성한 Patient 인스턴스 활용
  - 테스트 서버
    - http://localhost:8080/fhir
    - http://hapi.fhir.org/baseR4

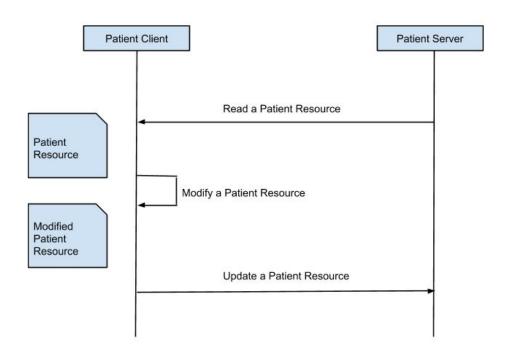
# 2. FHIR 서버에 등록 - Create

- Endpoint
  - /Patient
- Content-Type
  - application/fhir+xml
  - application/fhir+json



# 2. FHIR 서버에 등록 - Update

- Endpoint
  - /Patient/{리소스ID}
- Content-Type
  - application/fhir+xml
  - application/fhir+json



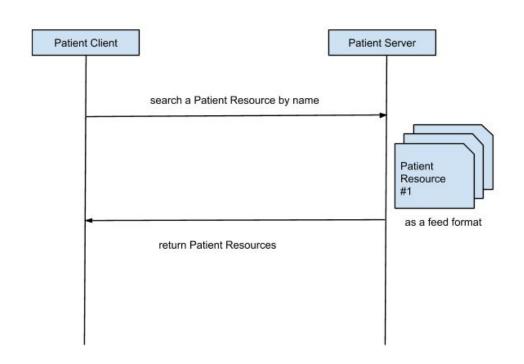
# 3. 등록한 리소스 조회 - Read & Search

#### Read

- Endpoint
  - /Patient/{리소스ID}
- Content-Type
  - application/fhir+xml
  - application/fhir+json

#### Search

- Endpoint
  - /Patient
- Query Parameters
- Content-Type
  - application/fhir+xml
  - o application/fhir+json



### 3. 등록한 리소스 조회 - Read & Search

- 등록한 리소스 조회 (Read)
  - Response Body
    - 조회한 리소스 타입으로 반환됨.
      - Patient 조회 → Patient 반환
      - Observation 조회 → Observation 반환
- 등록한 리소스 검색 (Search)
  - Response Body
    - Bundle 타입으로 반환됨.
      - Patient 검색 → Bundle 반환 (type: searchset)
      - Observation 검색 → Bundle 반환 (type: searchset)

### 3. 등록한 리소스 조회 - Search

Search Parameter

#### Types

- o number
- date
- string
- token
- reference
- o uri 등

Common Parameters

- o \_id
- \_lastUpdated
- \_profile
- c \_text
- \_tag, \_security,\_content, \_list, \_has,\_type

Search result

#### parameters

- o \_sort
- \_count
- \_include
- c \_total
- \_revinclude, \_summary,\_elements, \_contained,\_containedType

# 3. 등록한 리소스 조회 - Search

#### Prefix

- 검색 시 조건과 일치율을 조정하기 위한 접두사
- o eq:equal
- o ne : not equal
- o gt: greater than
- o ge: greater or equal
- le: less or equal
- o It: less than
- o sa, eb, ap 등

#### Prefix 사용법

- {BaseURL}/{Resource}?{[parameter]=[prefix][value]}
- o **number**, date, quantity에만 사용 가능

## 3. 등록한 리소스 조회 - Search

#### Modifier

- 검색 방법을 조정하기 위한 수식어
- o :missing
- :exact
- o :contains
- :[type]
- :text, :below, :above 등
- 사용법
  - o {BaseURL}/{Resource}?{[parameter][:modifier]= [value]}

# 감사합니다