

# Practicum Sprint #2

## 1. PROJECT SUMMARY

FHIR Implementations Guides are key to interoperable health information. Standardizing data representation and terminology is key to preserving semantic interoperability. FHIR IGs are crafted by groups of stakeholders and domain experts who must adapt their models to meet growing challenges. Example data is a key part of a FHIR IG, but the example data provided often is not comprehensive or becomes out of date. This project will generate example information dynamically based on a HL7 FHIR implementation guide. The project will build on the example data of the existing tooling called Synthea which is a synthetic patient population simulator. The project uses any implementation guide Structure Definition to plug in all information that is missing from the generated Synthea example data bundle.

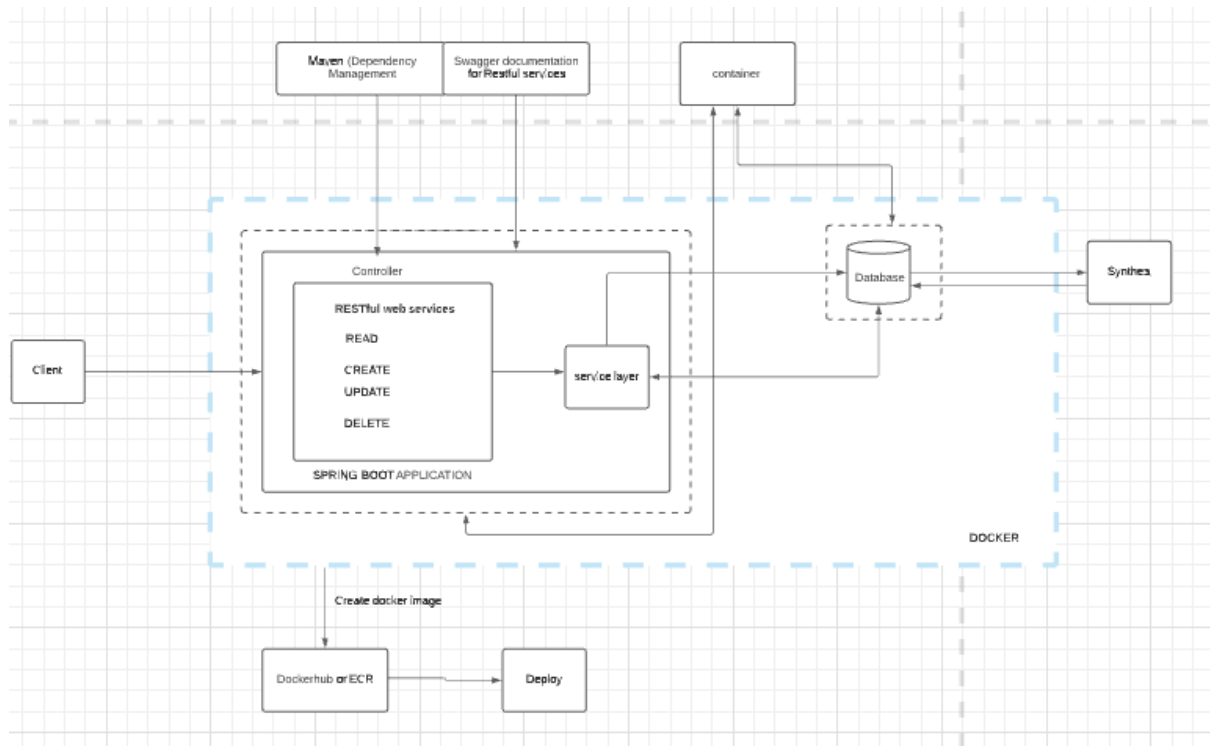
## 2. TOOLS AND TECHNOLOGY

- Programming language **Java v11**
- Use **Spring boot framework (v2.5)** to build Restful services. We will be using **Synthea** as a data resource.
- We are planning to write a unit test to make sure each function is returning the result as expected before using the controller to build a service.
- Instead of documenting each service manually, we will make use of **Swagger**. This makes it tremendously easy for client developers to access and use the API to build front end applications. The same tooling will also be used to test basic CRUD functionality of the API by providing the required JSON payloads.
- **Postman** as the automation tool to test and validate the services functionality
- We will be using **PostgreSQL database (v14)**
- We will be using **Docker** to containerize our application.
- Docker images will then be pushed to either **docker hub or ECR**.
- That image will then be used for the deployment.

## 3. DATA RESOURCES

- FHIR IG: provides information on implementing FHIR for a specific use-case. Generally it adds additional data constraints/guidelines ("Profiles") on top of an existing base version Resources (currently FHIR R4). Here's an example of an IG for a set of profiles called US Core: <https://www.hl7.org/fhir/us/core/examples.html>
- Synthea data: Generated by Synthea. For each synthetic patient, Synthea data contains a complete medical history, including medications, allergies, medical encounters, and social determinants of health.

#### 4. ARCHITECTURE DIAGRAM



#### 5. SCREEN MOCK-UPS FOR SWAGGER UI (NOTE: CRUD OPERATION IN OBSERVATION RESOURCE IMPLEMENTED BY SWAGGER UI)

The generated json data will be documented using Swagger UI as depicted in the following screenshots.

Observation The Observation FHIR resource type	
GET	/Observation/{id} read-instance: Read Observation instance
PUT	/Observation/{id} update-instance: Update an existing Observation instance, or create using a client-assigned ID
DELETE	/Observation/{id} instance-delete: Perform a logical delete on a resource instance
PATCH	/Observation/{id} instance-patch: Patch a resource instance of type Observation by ID
GET	/Observation/{id}/_history/{version_id} vread-instance: Read Observation instance with specific version
GET	/Observation search-type: Search for Observation instances
POST	/Observation create-type: Create a new Observation instance
GET	/Observation/_history type-history: Fetch the resource change history for all resources of type Observation
GET	/Observation/{id}/_history instance-history: Fetch the resource change history for all resources of type Observation
GET	/Observation/\$lastn

Users will have the option to validate and test the API services from the UI itself without having to import it to the postman tool. As shown in the screenshot below, from the swagger UI, users can pass the required parameters to create the observation resource. If anything

goes wrong, appropriate code will be returned along with an error or successful message. This serves well for both testing and documenting purposes.

POST

/Observation create-type: Create a new Observation instance

Parameters

No parameters

Request body

application/fhir+json

{  
  "resourceType": "Observation"  
}

Execute

The example of returned response:

Responses

Curl

curl -X 'POST' \  
  'http://hapi.fhir.org/baseR4/Observation' \  
  -H 'accept: application/fhir+json' \  
  -H 'Content-Type: application/fhir+json' \  
  -d '{  
    "resourceType": "Observation"  
  }'

Request URL

http://hapi.fhir.org/baseR4/Observation

Server response

Code

Details

201

Undocumented

Response body

{  
  "resourceType": "Observation",  
  "id": "2645658",  
  "meta": {  
    "versionId": "1",  
    "lastUpdated": "2021-10-16T23:45:38.267+00:00"  
  }  
}

Download

Response headers

connection: keep-alive  
content-encoding: gzip  
content-location: http://hapi.fhir.org/baseR4/Observation/2645658/\_history/1  
content-type: application/fhir+json; charset=utf-8  
date: Sat, 16 Oct 2021 23:45:38 GMT  
etag: W/"1"  
last-modified: Sat, 16 Oct 2021 23:45:38 GMT  
location: http://hapi.fhir.org/baseR4/Observation/2645658/\_history/1  
server: nginx/1.18.0 (Ubuntu)  
transfer-encoding: chunked  
x-powered-by: HAPI FHIR 5.6.0-PRE7-SNAPSHOT/2581cd1446/2021-10-04 REST Server (FHIR Server; FHIR 4.0.1/R4)  
x-request-id: KTg5L8c6f3mcbVwo

Responses

Code

Description

Links

200

No links

## 6. REFERENCES

- HL7 International - US Realm Steering Committee. (n.d.). *US Core Implementation Guide*. USCore. Retrieved October 16, 2021, from <https://www.hl7.org/fhir/us/core/all-examples.html>.
- Cangioli, G. (Ed.). (2021, July 21). *International Patient Summary Implementation Guide*. HL7.FHIR.UV.IPS\International patient summary implementation guide - FHIR V4.0.1. Retrieved October 16, 2021, from <https://build.fhir.org/ig/HL7/fhir-ips/>.

## 7. PROJECT TASKS

### Sprint 1:

- Form a team
- Identify the list of projects which we are interested in and submit it for confirmation.

### Sprint 2:

- Initial discussion about the project.
- Complete Design
- Implementation plan

### Sprint 3:

- Create a database
- Confirm the relevant APIs to be used for development

### Sprint 4:

- Develop the project
- Integrate the API's and validate the output.

### Sprint 5:

- Continue with development and dockerize
- Add Unit Tests

### Sprint 6:

- Process the data from Synthea

### Sprint 7:

- Fine tune the processing to fit the specific messaging formats
- Performance improvements

### Sprint 8:

- Test for various scenarios with positive, negative and edge cases

### Sprint 9:

- Peer review
- Deploy / Commit the code

### Sprint 10:

- Project Demo and Retrospective

## 8. PROJECT TIMELINE

Task Name	Duration ①	Start	Predecesso...	Assigned To	Finish	% Complete	Status
<b>Sprint #1</b>	11d	09/06/21			09/20/21	100%	Complete
Form a team							
List of interested projects							
<b>Sprint #2</b>	7d	10/08/21			10/18/21	60%	In Progress
Complete Design							
Implementation Plan							
<b>Sprint #3</b>	5d	10/18/21			10/22/21		Not Started
Create database							
Confirm the relevant APIs to be used for development							
<b>Sprint #4</b>	6d	10/25/21			11/01/21		Not Started
Develop the project							
Integrate the API's and validate the output							
<b>Sprint #5</b>	6d	11/01/21			11/08/21		Not Started
Continue with development and dockerize							
Add unit tests							
<b>Sprint #6</b>	6d	11/08/21			11/15/21		Not Started
Process data from Synthea							
<b>Sprint #7</b>	6d	11/15/21			11/22/21		Not Started
Fine tune data from Synthea to fit the specific messaging formats							
Performance Improvements							
<b>Sprint #8</b>	6d	11/22/21			11/29/21		Not Started
Test for various scenarios with positive, negative and edge cases							
<b>Sprint #9</b>	6d	11/29/21			12/06/21		Not Started
Peer review the code							
Deploy / Commit the code							
<b>Sprint #10</b>	6d	12/06/21			12/13/21		Not Started
Project Demo and Retrospective							

## 9. NEEDS / RISKS

- One risk right now is an incorrect project timeline. Since we are still lacking information, the timeline for the project could initially be incorrectly calculated and set, and the functional features of the final product were not determined in a timely manner. Thus, we are not locked into our current timeline, and it is subject to change as we gather more information about the project.