1. **Project abstract**:

*Please summarize your submission briefly, including information about the problem you’re trying to solve, how you have solved/are solving it and how your solution worked in practice.*

LibreHealth is a community that develops and implements health information technology applications to improve health outcomes. The platform consists of three projects. [LibreHealth EHR](https://librehealth.io/projects/lh-ehr/) is an open source ambulatory EHR designed for easy use and customizable for variety of settings. The second component [LibreHealth Radiology](https://librehealth.io/projects/lh-radiology/) is a radiology information system. The third component is the [LibreHealth Toolkit](https://librehealth.io/projects/lh-toolkit/) which is an application programming interface (API) created to augment the other two components and permit new healthcare applications. It runs on Windows, Linux and OSX and was initially written with the open source tools: [Hibernate](https://www.tutorialspoint.com/hibernate/), [Spring](https://en.wikipedia.org/wiki/Spring_Framework) and [MySQL](https://www.mysql.com/). This project describes how the platform was reconfigured to more efficiently handle FHIR resources.

1. **Project rationale, impact and innovation**:

*Describe the problem/issue your submission is trying to address. What is the problem? Who does the problem affect? What is the impact (in terms of reduction of morbidity/mortality, number of people/patients affected, etc.) of solving the problem, both in the long as well as in the short term? In what ways is your project innovative?*

LibreHealth toolkit was originally written using Hibernate and Spring and used hard coded FHIR resources. It was also tightly coupled with a single database. Due to this it was hard to maintain the project as the FHIR resources could not be updated easily along with the latest specification and it was hard to use the toolkit with other persistence mechanisms.

This project provides an entirely reactive and modular solution to the above problems. As it uses Spring Data Repositories, there is no need to revamp the whole codebase whenever the FHIR specification is updated. In addition to this, it does not hard code the FHIR resources, rather it retrieves them as a dependency and maps them to the database supported types using a conversion mechanism.

1. **Project design and implementation**:

*Describe how you designed and implemented your project. How did your design address the problem you described above? How did you implement your solution and what requirements must be met to be able to do so? What challenges did you have to overcome?*

As a part of this project I built an entirely reactive Spring Data app which uses [Cassandra](https://www.mysql.com/) as its NoSql database to provide support for create, read, update and delete (CRUD) operations using its REST API for the most recent FHIR standard [HAPI FHIR DSTU3](http://hapifhir.io/download.html) structures.

The app supports CRUD operations for the FHIR resources: Patient, Observation and Encounter and partial *search* operation for Patient with support for more resources to come.

As I was required to work with the HAPI FHIR structures directly without cloning them or modifying them locally, I needed to figure out a way to make Cassandra directly load the HAPI structures as tables. For this purpose, I created custom models which directly extended the original HAPI structures and annotated various parts of those classes to implement certain functions such as defining the primary key, ignoring certain fields etc.

One of the key advantages is that the original HAPI structures can be easily and implicitly casted to the original structures wherever needed without losing any data.

To properly save the data Cassandra needed to understand the complex data types of HAPI structures. The solution was to implement various converters which act as a layer between the models and the persistence layers and works to convert the complex data types to primitive data types which can be easily understood by Cassandra when writing the data and can just as easily be converted back to the complex data types at the time of reading the data.

1. **Project evaluation and sustainability***:*

*How did you evaluate your project? What kind of qualitative and quantitative data did you gather, and what conclusions do you draw from them? Did your project achieve its goals in terms of implementation and impact? How will your project be sustained?*

Since this project is Work In Progress, they are not evaluated yet.

1. **Does your solution have paying customers?**

No as this project is work in progress.

1. **When was your solution conceived?**

April, 2018

1. **When was your solution implemented?**

August, 2018

1. **How many users does your solution have or how many patients have been impacted by it (please indicate time frame)?**

No implemented project (WIP)

1. **Website/URL**(any additional information/video/etc. about the project)

<https://forums.librehealth.io/t/project-implement-spring-data-to-librehealth-toolkit/2341/102>

1. **Logo/headshot and promotional photo**



1. **Twitter project summary**

<https://twitter.com/librehealthio?lang=en>

1. **Any other information about the project we should know about?**

Link for the reactive lh-toolkit repo

<https://gitlab.com/yashdsaraf/reactive-lh-toolkit>