GSoC 2020 : Improve FHIR Search

Primary mentor	Ian Bacher
Backup mentor	Reagan patrick Makoba
Assigned to	Varun Gupta

Abstract

FHIR is an emerging standard for healthcare interoperability. While OpenMRS has been an early adopter of the FHIR standard, there is still work to do to get our FHIR implementation working properly. This project focuses on improving our implementation of FHIR's search capabilities. FHIR defines a very detailed API for searching that is fundamental to providing the correct data to any front-end applications. Currently, the OpenMRS FHIR module supports some basic search functionality, but it lacks capabilities to search across multiple properties or support advanced search operations such as _include. This project to extend this module to include support for these more complex search operations.

Project Champions

- Ian Bacher

Skills Needed

- Good Java skills
- Familiarity with SQL
- Bonus points for familiarity with Hibernate and especially the Criteria API
- Bonus points for knowledge of how to write efficient queries and how to optimize queries

Objectives

- Implement the FHIR search API across core resources (Patient, Encounter, Observation, Practitioner, Person, etc.), including the ability to sort results and to chain queries
- Implement proper paging using the HAPI FHIR IPagingProvider and the Hibernate Criteria API
- Implement many of the default parameters for all resources, including _id, _lastUpdated
- Implement some advanced features, including _include, _elements, _summary, _list and _count
- Implement support for Lucene for the versions of OpenMRS where the Lucene index is available for a given resource

Extra Credit

- Implement the ability to search using the _filter special syntax
- Implement the ability to search via GraphQL

Extra Extra Credit

- Integrate FHIR search with ElasticSearch

Getting Started

- Read up on FHIR. Good introductions can be found in this post, in this slide show, or this video
- Read up on the HAPI FHIR library which we use for FHIR support, especially the part on search
- Read up on the Hibernate Criteria API
- Look through the FHIR module source code, especially the search implementation for Observation

Objectives Completed During GSoC 2020:

- Implement the FHIR search API across core resources (Patient, Encounter, Observation, Practitioner, Person, etc.), including the ability to sort results and to chain queries (Completed)
    1. The FHIR module currently has a good amount of search added for all the resources, at least good enough for the initial version of the module. The core resources like Patient, Encounter, Observation, Practitioner, Person, Location have a lot of search parameters in place for a very specific search. The support for chained search has also been implemented for the reference parameters.
    2. The sorting functionality has been added wherever meaningful. We can also sort using multiple sort parameters by specifying them in the _sort parameter. The sorting can also be done both ways - ascending as well as descending.
    3. One of the key fixes done was to ensure that search returned distinct search results, which was earlier not the case as same resources were being returned multiple number of times due to the joins between the hibernate tables.

- Implement proper paging using the HAPI FHIR IPagingProvider and the Hibernate Criteria API (Completed)

1. Paging support has been added for all the resources which have a resource provider. This allows us to return only a fixed and desired number of search results on a single page and the link to the next page, if it exists.

- Implementation of default search parameters (Completed)
    1. The _id search parameter has been implemented to search for a particular resource based on the specified UUID. This is of the most significance when used alongside other advanced parameters like _include and _revinclude or when the user wants to retrieve information related to only a specific resource.
    2. The _lastUpdated search parameter has been added for all the resources based on the last time they were modified, or based on the time when they were created if they are immutable (they were never modified).

- Implementation of advanced search parameters (Completed)
    1. The advanced parameters _elements, _summary and _count are by default supported by the FHIR server.
    2. The support for _include was added for resources wherever possible according to our present search capability. _include is used to include associated resources for the resources matching the specified search criteria.
    3. The support for _revinclude was added as well, again according to our present search capability. _revinclude allows the user to retrieve resources of a particular type that refer to the resources matching the specified search criteria.
    4. _include and _revinclude support was added by including the additional resources once the search for the matching resources was complete and the resources to be displayed on a particular page was decided.

Resources:

- FHIR Module source code - https://github.com/openmrs/openmrs-module-fhir2
- HAPI FHIR Documentation
- The official HL7 guide

Future Work:

- Implement support for Lucene for the versions of OpenMRS where the Lucene index is available for a given resource
- Implement the ability to search using the _filter special syntax
- Implement the ability to search via GraphQL
- Integrate FHIR search with ElasticSearch