

2020 Mar 18 FHIR Squad Meeting

Attendees:

Piotr
Casey
Varun
Bashir
Jacinta
Bett
Siddarth
Ian
Daniel
Jen

- Status updates:
 - Bett
 - Implementation of medication request
 - Provenance resource he made an error - create and update resource when nothing is being updated - he fixed that
 - Added support to create a condition - though he will think about these issues
 - figure out how to deploy fhir 2 in a dockerized environment haven't made progress
 - Jacinta
 - 3 issues - optimize db operations for search methods for allergy intolerance - fixed
 - Add search functionality to medication resource
 - Add additional fields for observation resource - relying on medicationrequest and service request resources
 - Ian
 - Made an implementation by sorting people by name
 - Piotr
 - Trying to get the fhir module to work with 2.0.5 OpenMRS platform and modules from isanteplus - refapp 2.6.0 mostly
 - Ran into some problems with the migration issues and refactored the liquibase files and db migrations work now
 - Other problems that he will log involving the condition resource - issues with support in 2.0.5 and emr api module
 - Will post or create an issue about this
 - Bashir
 - Condition search - should have a pr ready this week

- Wanted to ask about 2.2 or less - can send a pr that has some of the search functionality - the db schema is different - should we have smaller or bigger PRs?
 - Prefer smaller
 - Haven't implemented all the search parameters in the ticket but will send the PR for all that we have implemented
- Siddarth
 - 2 issues - obs search
 - Oauth module
- Varun

Varun

Should it work for problems with 2.0 and further

Haven't been able to get it set up correctly for 2.0.5

Bett - was working on the condition resource - couldn't reproduce the error but he wanted to move away from depending on the api, but had to remove a condition mapping - so now you have to have emr api installed for fhir 2 module to work - for you to install emr api there are other dependencies you have to also install

Maybe supporting conditions before it got moved into core is actually not a good idea?

Introducing and dependency on emr api - the core fhir module should be deployable as easily as possible

Before emr api the conditions were stored as obs - you can query from obs

We should really be supporting the latest and greatest rather than legacy stuff - does it make sense to spit out the condition support as a special module that has special dependencies

Another option is that the condition support of the fhir module can be only for 2.2 and above - when you make the api call for condition related fhir resources , but by default it can not be supported - sticky about that is that the conformance metadata - if there is a condition resource provider it will say its going to fix it.

Ultimate vision is that there should be a module that can be run completely separate from openmrs

What aspects of condition search require emr api?

Brief period - conditions were served as obs , then moved to conditions class in emr api - module for classes that haven't made it into core but are candidates.

For 2.2 it shouldn't be required - the problem is that i don't know if anybody is actually running 2.2 in production.

Is there in general a way we want to provide or turn of support for a given resource.

Core fhir module should only support what we can out of core - then we can add additional submodules that support others

What to do in the short term? Comment out the api 2.1 module and build from there

Piotr will post an error log and then we can hopefully figure out what lingering issues there are

If we are going to split out the conditions stuff - it shouldn't take too long to do that

Piotr will keep going to figure out what his errors are - can't currently get it to run without bean and injection errors.

Piotr will document his findings and send it out

Bashir - what would be the difference to split it out?

Ian - if we are splitting it out, we are splitting out the entire resource support

As part of the long term vision there should be different modules that build on top of fhir module

This would be a different case than those sort of things - wouldn't necessarily want specific resources for which we have clear mappings to be stored in submodules

Bett - i think that the best approach would be to not remove that
Add support for conditions but 2.2+

Ian - concerned about the condition resource provider and it can't find the service its relying on

Bett- but i thought conditional resources work well with that?

Lets see if we can target 2.2+ support for conditions without having to move the whole thing out
- if we can turn off that support for any platform that can't support it - have there be no errors

Bashir

Work on the google cloud healthcare team - there is fhir store for people to bring their healthcare data to google cloud and there is an export

Specifically working on NBU next billion users - most of the team focuses on developed world - his soft team works specifically on healthcare tools in the developing world - a month ago we

converged on openmrs - it is very commonly used - what we are trying to achieve is to see how we can help - the primary goal is to make the dhc tools

Specific idea - if we have everything in openmrs being represented as fhir resources - then it opens up some opportunities for openmrs - one of the needs that he mentioned is analytics needs

Bring all the data together to a data warehouse and run analytic queries - export fhir resources into BigQuery

Is this a need?

If we have openmrs data represented as fhir resources - then maybe it can be a way to tackle the sync problem

A lot of interest in this- Bashir is not here to advocate for google apis but if you have an export you can feed into fhir store and feed into BigQuery - piotr is trying to propose this pipeline and pilot