FHIR® North – Canada's Connectathon

What's a FHIR Connectathon?

A FHIR connectathon is a gathering of implementers interested in testing to see whether they can implement and successfully interoperate using some portion of the HL7® FHIR® specification[1]. It is less formal than IHE connectathons in that there is no independent testing or certification process. The objective is to give implementers experience working with the FHIR specification outside a production environment and to provide feedback that improves FHIR.

Why are we holding a FHIR Connectathon in Canada?

FHIR is attracting significant interest world-wide. Although the standard is still evolving, it's being used in production in multiple countries. There have been numerous connectathons held in the U.S., Europe, South America and Australasia. It seems time to give Canadian developers a chance to take it out on the road.

How will the FHIR North Connectathon work?

This connectathon has three tracks (see next page). Implementers choose which track(s) they wish to implement and whether they want to participate as a client, server or both. Depending on the complexity of the track, a participant may choose to perform some or most of their solution development up-front, though some degree of bug fixing, tweaking or enhancing on-site is typical. For the "introduction" track for client systems, all development can happen on-site. During the connectathon, implementers work to test the ability of their systems to execute their chosen scenario with other participants as well as with public test servers.

Who can participate?

At past connectathons we've had a wide range of participants – active developers already familiar with FHIR, individuals who have never seen the FHIR specification and architects who have not coded in years. Almost all have been able to produce an interoperable implementation over the course of a day.

How do I prepare?

At minimum, come prepared with a development environment – FHIR provides reference implementations in Java, C#, Delphi/Pascal, Javascript and Objective C. Ideally, browse through the FHIR specification, download the appropriate reference implementation or one of the open source implementations and try using with one of the many public test servers. The more coding you have done up front, the more you can focus on edge cases and debugging

---

[1] "HL7" and "FHIR" are registered trademarks of Health Level Seven International

Sponsored By:

FHIR® North – Canada's Connectathon

What's in it for me?

A chance to meet and ask questions with some of the world's leading FHIR experts, a chance to see whether FHIR really lives up to the hype and a chance to shape the specification.

What are the details?

April 29th at Mohawk College (day before Apps for Health). Registration fee is $45 which covers entry, lunch and coffee breaks. A pizza dinner is sponsored by Gevity Consulting. Register at
http://www.mohawkcollegeenterprise.ca/en/event_list.aspx?groupId=3

Track 1: Patient

A. Register a new patient

- **Action:** (Patient Demographics consumer) creates a new patient and save to Patient Service. The client can assign the Id.
- **Precondition:** Patient does not exist in service prior to action
- **Success Criteria:** Patient created correctly on server (use browser to inspect Patient)
- **Bonus point:** The Patient resource has an extension

Note: the requirement for the client to assign the Id has been relaxed. However, if the server assigns the Id, then the client will need to be able to retrieve the Id from the server response or by a patient query.

B. Update a patient

- **Action**: (Patient Demographics consumer) updates the patient created in scenario #1 and updates to Patient Service. The patient is retrieved by Id.
- **Precondition**: Patient has been created
- **Success** Criteria: Patient updated on server (use browser to inspect Patient)
- **Bonus Point #1:** Update a patient that has extensions, but leaving the extension untouched.
- **Bonus Point #2:** Update a patient that has extensions, and update the extension also.

C. Retrieve Patient history

- **Action:** (Patient Demographics consumer) searches the patient Service for the history of a Patient
- **Precondition:** There is a patient that has at least one update
- **Success Criteria:** Patient's history displayed in interface. (use browser query Patient Service)
- **Bonus point:** The UI allows the user to display previous versions of the Patient

D. Search for a patient on name

- **Action:** (Patient Demographics consumer) searches the patient Service for patients with a given name
- **Precondition:** Patients with that name have been created
- **Success Criteria:** Patients displayed in interface. (use browser query to confirm)

Sponsored By:

FHIR® North – Canada's Connectathon

Some helpful links:

- Java client sample.
- .net client sample.

Track 2 – SMART on FHIR

This track focuses on user-facing apps that launch from an EHR or PHR. SMART on FHIR uses open standards (FHIR, OAuth2, OpenID Connect) to provide a platform for health apps that integrate with existing Health IT systems.

- For an overview, see http://docs.smartplatforms.org/
- Questions? Visit the SMART on FHIR Google Group.

A. Build a SMART Server

Demonstrate a SMART on FHIR server. A successful server will do the following:

1. Support `/Patient` and `/Observation` end points, so an app can retrieve demographics and vital signs.

2. Support the SMART on FHIR launch and authorization, so an app can request permissions and learn patient context via OAuth2.

Use this **quick-start guide for SMART servers** with examples of URLs, parameters, LOINC codes, and data payloads you'll need to get started.

- Other references

    http://fhirblog.com/2014/08/02/smart-on-fhir-part-1/

    http://fhirblog.com/2014/08/12/smart-on-fhir-adding-oauth2/

B. Build a SMART App

Demonstrate a SMART on FHIR client. A successful client will be a Web app or mobile app that can run against our SMART on FHIR testing server. It should:

- Talk to SMART's sandbox server (and any other servers presented at the Connectathon)
- Authorize to access patient data
- Retrieve patient data including demographics, vitals, and labs
- Present some useful view (can be just straight data table) of patient data

Use this **quick-start guide for SMART apps** with code and examples to get you started.

Sponsored By:

GEVITY   MOHAWK COLLEGE   UHN Toronto General Toronto Western Princess Margaret Toronto Rehab

FHIR® North – Canada's Connectathon

You may also want to explore source code for [sample apps on GitHub](#).

You can get started developing from `http://localhost` without registering your app -- but when you want to get your app running on the Web, just [follow our self-service app registration instructions](#).

Track 3 – Questionnaire
This track makes use of the Questionnaire and Questionnaire answers resources to create and populate forms

1. Retrieve a completed QuestionnaireAnswers Instance
- **Action:** Download a completed QuestionnaireAnswers and display to the user
- **Precondition:** Completed QuestionnaireAnswers instances exist on the server
- **Success Criteria:** A rendition of the QuestionnaireAnswers is presented
- **Bonus point:** Allow the user to display all the completed questionnaires for a given user and select one to display

2. Retrieve a Questionnaire definition
- **Action:** Download a Questionnaire and render a user interface from it
- **Precondition:** Questionnaire instances exist on the server
- **Success Criteria:** A user interface matching the Questionnaire is presented
- **Bonus point:** Provide a list of questionnaires for the user to select from
- **Bonus point:** Choose a Questionnaire that references an external (or contained) ValueSet, and use the ValueSet contents to display a selection list

3. Save a QuestionnaireAnswers Instance
- **Action:** Complete a QuestionnaireAnswers instance based on a Questionnaire and save it to a server
- **Precondition:** Questionnaire previously downloaded
- **Success Criteria:** The completed QuestionnaireAnswers can be retrieved and displayed

4. Create a Questionnaire Template
- **Action:** Create a new Questionnaire as a template and save it to a server
- **Precondition:** None
- **Success Criteria:** The Questionnaire can be retrieved and used to generate a completed Questionnaire instance
- **Bonus point:** Implement a template that uses some of the standard extensions - like show/hide...

FHIR® North – Canada's Connectathon

Track 4 – Experimental

This track is wide open – make use of any of the FHIR resources and capabilities as you see fit – so long as you can find at least one communication partner to interact with. To coordinate seeking partners, implementers are encouraged to sign up on the FHIR implementer's Skype chat and/or list server. Instructions can be found on the HL7 wiki: http://wiki.hl7.org?title=FHIR.

Sponsored By: