**2022-05 Questionnaire**

- **Short Description**
  - Questionnaires and forms permeate healthcare. This track will continue testing implementation of the US Core Questionnaire profiles, the Structured Data Capture (SDC) implementation guide and the Patient Reported Outcomes (PRO) implementation guide, which all provide guidance on how to define and complete questionnaires in healthcare, including the ability to automatically populate forms and the ability to extract resource information from completed forms.

- **Long Description**
  - Track objectives include:
    - Representing a wide variety of forms using FHIR and verifying that the SDC implementation guide supports the needed representation capabilities, including flow control logic and calculations
    - Testing authoring application abilities to create and edit such forms
    - Test the ability of questionnaire completion tools abilities to handle different types of forms
    - Test the ability of applications to automatically complete forms using existing data
    - Test the ability of systems to handle rendering 'adaptive' forms (forms where a third-party system decides on the next question based on prior answers)

- **Type**
  - Test an Implementation Guide

- **Submitting Work Group/Project/Accelerator/Affiliate/Implementer Group**
  - FHIR-I

- **Track Lead(s)**
  - Paul Lynch; Brian Postlethwaite (maybe); Joee Garcia (maybe)

- **Track Lead Email(s)**
  - plynch@nih.gov

- **Related Tracks**
  - Role 1 - Forms Repository Server (SDC Form Manager)
  - Role 2 - Data Provider
  - Role 3 - Populate Server
  - Role 4 - SDC Form Receiver
  - Role 5 - Form Renderer (SDC Form Filler)
  - Role 6 - SDC Form Designer

- **Scenarios**
  - Scenario 1 - Create a new Form with Observation or Expression-based pre-population fields inside
  - Scenario 2 - Provide data for pre-population based on a Search Bundle
  - Scenario 3 - Perform a pre-population of provided form
  - Scenario 4 - Validate a form provided by a user that has been pre-populated
  - Scenario 5 - Render a form with data that has been been pre-populated
  - Scenario 6 - Terminology questionnaire choice column extension
  - Scenario 7 - AnswerExpression as another alternative to answerOption or answerValueSet
  - Scenario 8 - Observation-based extraction
  - Scenario 9 - StructureMap-based extraction
FHIR Version

- Current build (experimental)
- FHIR R4 (primary)

Specification(s) this track uses

- http://build.fhir.org/ig/HL7/sdc
- http://build.fhir.org/ig/HL7/patient-reported-outcomes/
- http://fhir.org/guides/argonaut/questionnaire/1.0.0/

Artifacts of focus

- http://hl7.org/fhir/Questionnaire
- http://hl7.org/fhir/QuestionnaireResponse
- http://hl7.org/fhir/ValueSet

Expected participants

Zulip stream

https://chat.fhir.org/#narrow/stream/179255-questionnaire

Track Kick Off Call

4/28, 5 p.m.-6 p.m. U.S. Eastern Time (GMT-4)  Link: https://hi7-org.zoom.us/j/8583747934?pwd=VnZiWFZsNm1TSk5QWmZlbVJyQVZGdz09.

Track Details

System Roles used in the scenarios

Note that software for many of these roles is already written (see "Software and Servers" below), so you can try scenarios involving roles you have not written yourself.

Role 1 - Forms Repository Server (SDC Form Manager)

A server that holds the FHIR Questionnaires to be used for rendering and pre-population (forms should have the SDC pre-population information in them)

Role 2 - Data Provider

A FHIR Server that provides data for use in the pre-population of a form, and also clinical context (preferably through smart on fhir app launch)

For best results the server should also support the search batch processing, reducing the number of calls to the server to retrieve data to populate with.

Role 3 - Populate Server

A server that implements the SDC $populate operation, with FHIRPath or Observation based data extraction processing. Note that Form Renderers (see below) may handle population as well, so population can be either done server-side or client-side.

Role 4 - SDC Form Receiver

A server that will receive a FHIR QuestionnaireResponse that has been submitted by a user, and subsequently validated.

Role 5 - Form Renderer (SDC Form Filler)

A user interface for displaying a Questionnaire/Questionnaire response, permitting a user to complete the form and submit the answers.

Role 6 - SDC Form Designer

Tools that build and edit form definitions.

Software & Servers

- SDC Implementations: Includes software for "Form Designer", "Form Filler (Renderer)", and Populate Server
- Public Test Servers: For "Form Repository Server", "Data Provider", and "Form Receiver"

Scenarios

The following are some suggested scenarios to try. Feel free to make up your own to exercise other interesting areas of the SDC Implementation Guide.

Scenario 1 - Create a new Form with Observation or Expression-based pre-population fields inside
This will be used to get experience creating forms, including data extraction information, and fhirpath expressions

**Action:** Create a new form and save it to a FHIR Server (Such as the Telstra Health Test Server - which will validate the definition)

**Precondition:** (none)

**Success Criteria:** Form was able to validate and be used by a populate server, retrieving data from several FHIR Servers (PMS)

**Bonus point:** Search bundle was external to the form

**Bonus point:** Data extraction includes multiple launch contexts

**Bonus point:** Multiple queries in the search bundle

**Bonus point:** Multiple groups with complex fhirpath extract expressions

**Scenario 2 - Provide data for pre-population based on a Search Bundle**

**Action:** Return the results a search for data as defined by a form pre-population definition

**Precondition:** A Form with pre-populate rules, a context to perform the searching

**Success Criteria:** able to return the results of the search

**Bonus point:** able to handle a bundle as well as individual search interactions

**Scenario 3 - Perform a pre-population of provided form**

**Action:** Implement the fhirpath based $populate operation

**Precondition:** A form definition with fhirpath based data extraction rules in it, and the set of required resources to complete the population (data context)

**Success Criteria:** A draft QuestionnaireResponse resource is created with all the answers as completed

**Bonus point:** pre-populate a definition with repeating sections

**Bonus point:** pre-populate a definition with multiple search bundle queries

**Scenario 4 - Validate a form provided by a user that has been pre-populated**

**Action:** perform QuestionnaireResponse validation against a defined Questionnaire definition, including SDC extenions.

**Precondition:** A form that has SDC validation extensions in it

**Success Criteria:** The basic FHIR validations, and Questionnaire Type validations are performed

**Bonus point:** SDC extensions are reported

**Bonus point:** SDC complex validation extensions are reported

**Scenario 5 - Render a form with data that has been been pre-populated**

**Action:** Ability to render the QuestionnaireResponse, and then submit it to a designated location (for validation etc)

**Precondition:** A QuestionnaireResponse with completed answers in it.

**Success Criteria:** Each completed answer should display with the data as extracted, and SDC extensions should also be honored, specifically enable when.

**Bonus point:** Fhirpath based enable when logic

**Scenario 6 - Terminology questionnaire choice column extension**

When using questionnaire-choiceColumn to indicate additional columns to show for an item's answerValueSet, there are some challenges in R4 around getting the data for the additional columns. It looks like you first need to do an $expand (maybe with filter) to get a list of N items, and then do an additional N CodeSystem $lookup queries to get the properties requested by the questionnaire-choiceColumn instances for each item. In R5, the properties can be requested with the $lookup, so it will be easier. The task here would be to see if the answer list can be rendered reasonably quickly in R4.

https://chat.fhir.org/#narrow/stream/179255-questionnaire/topic/questionnaire-choiceColumn

**Scenario 7 - AnswerExpression as another alternative to answerOption or answerValueSet**

Under some scenarios you need to have a more dynamic set of options than are possible to define with a static set, this extension permits you to provide an expression to define that set of possible values.
Examples:

- https://sqlonfhir-r4.azurewebsites.net/fhir/Questionnaire/coding-sampler
- https://github.com/lhncbc/lforms/blob/master/test/data/answerExpressionTest.json

**Scenario 8 - Observation-based extraction**

Create an Questionnaire in which some questions are marked for extraction into Observation resources. Write or use a tool to do the extraction.

**Scenario 9 - StructureMap-based extraction**

Write a Questionnaire and a StructureMap for extracting data from it, or write the code that implements the $extract operation.

**TestScript(s)**

None at this point, however will have a test service available for all roles, that you can switch various roles for testing, or provide your own questionnaires too. If someone wants to write TestScripts, they'd be welcome 😊

**Security and Privacy Considerations**

No expectation for OAuth or TLS in the test environment (though such technologies might well be needed in a production environment)