On-the-Fly Data Capture Tooling for FHIR/HL7 V2

Paul Lynch, Ye Wang, Ajay Kanduru, Xiaocheng Luan, Lee Mericle, Shennon Lu, and Clement McDonald

Lister Hill Center, National Library of Medicine, NIH

Presented by Clem McDonald, Paul Lynch, Ye Wang, and Ajay Kanduru

HL7 January 2018 Working Group Meeting, New Orleans, LA

February 1st, 2018

Why we built a general forms system

Presenter: Clem

Database systems as a motivating example

- In the 70's pre database systems, I started developing medical record software.
- We had to build a separate data structure for each kind of data, e.g. patient registry, prescriptions, the structures only existed in the programs and were held hostage by them.
- Had to write separate programs for storing data and for reporting content of each such data using these software data structures.
- Then came database systems.
- They magic they provided was the stored data structure definition.
- They freed the data structure from the programs.

Motivating example 2

- The programs then read the data structure from the database definition. This made it easier to keep different programs aligned and modify them.
- Also made it feasible to develop general tools, like report generators, and data input tools, because they would just read the data definition and create a new personality for each data definition.

Motivating example 3

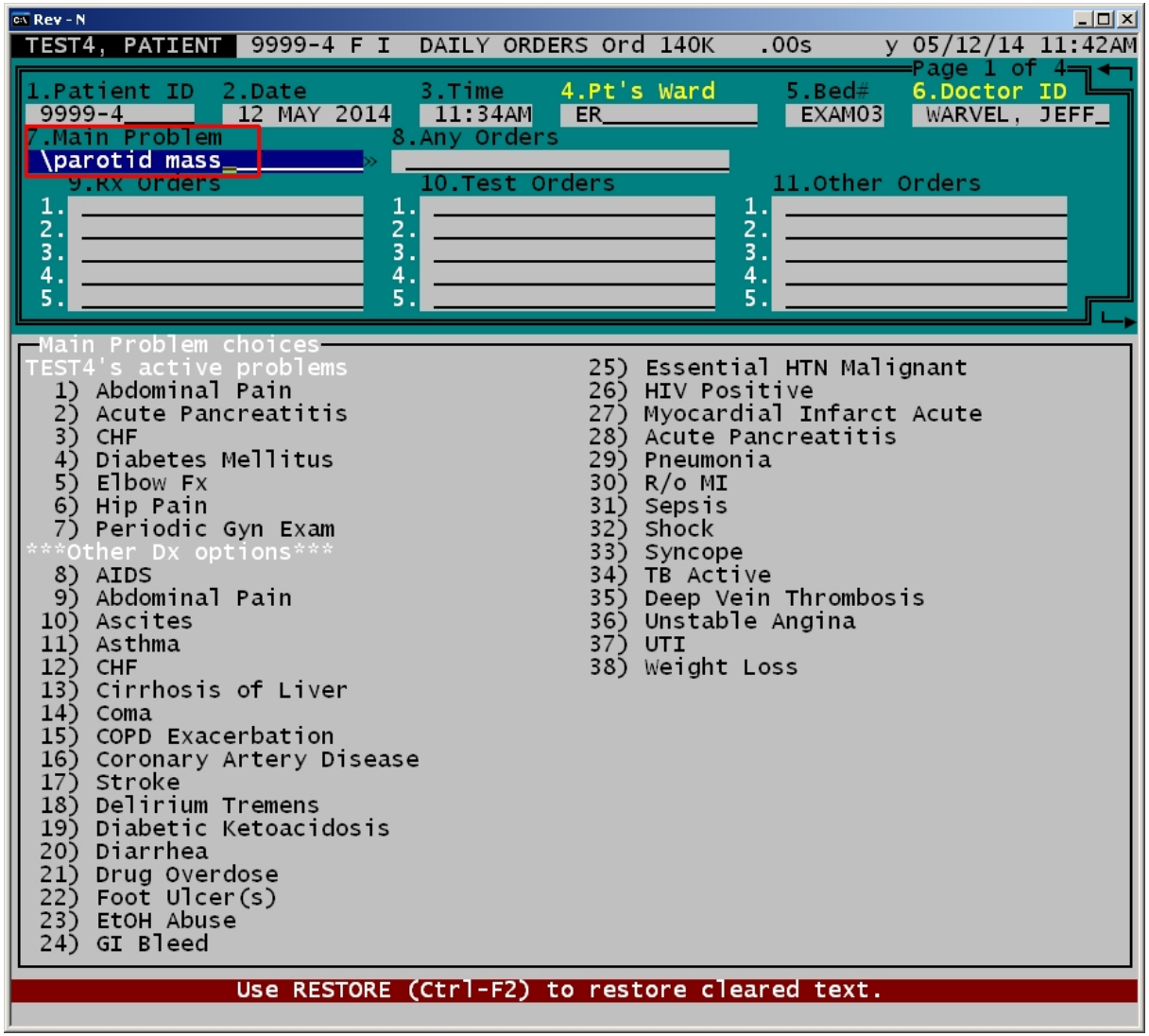
- We developed our own sort-of-relational database system in the mid 70's.
- The software work that remained was development of data entry forms, data display forms and special routines process the data that was entered.
- We thought that for sure, the industry would develop a stored form definition equivalent of stored database definitions and eliminate a big chunk of the remaining work of developing clinical applications X-forms looked like the answer, but it didn't happen.

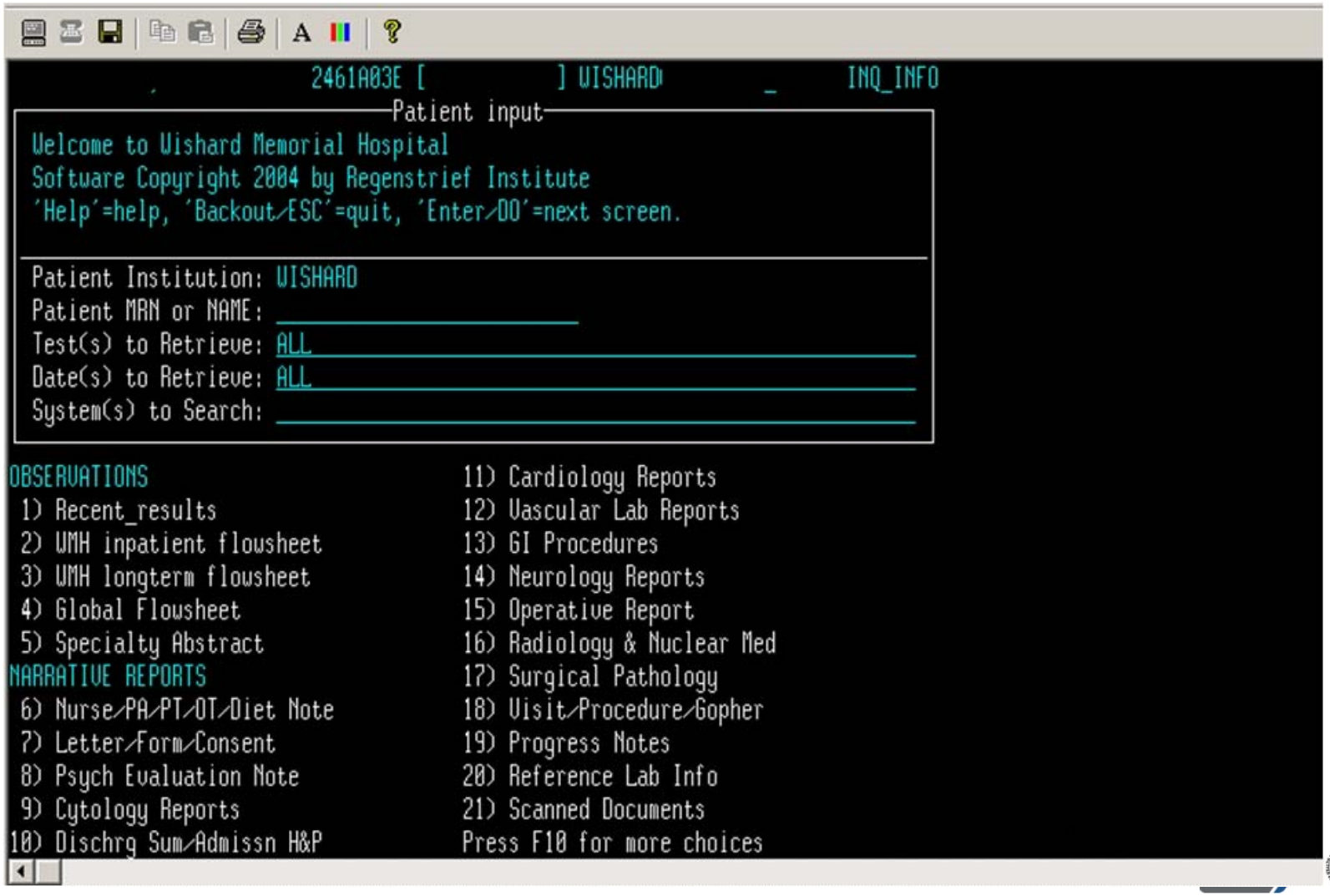
So we built our own

- In the Mid 80's we started building a PC based physician order entry and decision support work station called the Gopher. Wrote it in Revelation (a Pick operating system), which had its own built in database system.
- It was the basis of many papers in the New England Journal and JAMA including one that showed that such systems save money.
- To ease development we developed our own forms generator.
- All we had to do was record the attributes that applied to the form, over all, and the attributes (data type, answer lists, etc.) for each question/variable on the form AND a software routine that defined the next action to be performed.

Our own form development experience

- This made software development fast and “easy”. We could respond overnight to changes needed as we deployed.
- The form “became” the application - Enter something then do something (Where the do was a routine that would take actions such as store the data, run an arbitrary routine, create a flowsheet,) and all master files, reminder rules etc. were maintained through forms.


Rev - N
TEST4, PATIENT 9999-4 F I DAILY ORDERS Ord 140K .00s y 05/12/14 11:42AM
Page 1 of 4
1.Patient ID 2.Date 3.Time 4.Pt's Ward 5.Bed# 6.Doctor ID
9999-4 12 MAY 2014 11:34AM ER EXAM03 WARVEL, JEFF
7.Main Problem
\parotid mass
8.Any Orders
9.Rx Orders
10.Test Orders
11.Other Orders
1.
2.
3.
4.
5.
Main Problem choices
TEST4's active problems
1) Abdominal Pain
2) Acute Pancreatitis
3) CHF
4) Diabetes Mellitus
5) Elbow Fx
6) Hip Pain
7) Periodic Gyn Exam
Other Dx options
8) AIDS
9) Abdominal Pain
10) Ascites
11) Asthma
12) CHF
13) Cirrhosis of Liver
14) Coma
15) COPD Exacerbation
16) Coronary Artery Disease
17) Stroke
18) Delirium Tremens
19) Diabetic Ketoacidosis
20) Diarrhea
21) Drug Overdose
22) Foot Ulcer(s)
23) EtOH Abuse
24) GI Bleed
25) Essential HTN Malignant
26) HIV Positive
27) Myocardial Infarct Acute
28) Acute Pancreatitis
29) Pneumonia
30) R/o MI
31) Sepsis
32) Shock
33) Syncope
34) TB Active
35) Deep Vein Thrombosis
36) Unstable Angina
37) UTI
38) Weight Loss
Use RESTORE (Ctrl-F2) to restore cleared text.


2461A03E [ ] WISHARD INQ_INFO
Patient input
Welcome to Wishard Memorial Hospital
Software Copyright 2004 by Regenstrief Institute
'Help'=help, 'Backout/ESC'=quit, 'Enter/DO'=next screen.
Patient Institution: WISHARD
Patient MRN or NAME:
Test(s) to Retrieve: ALL
Date(s) to Retrieve: ALL
System(s) to Search:
OBSERVATIONS
1) Recent_results
2) WMH inpatient flowsheet
3) WMH longterm flowsheet
4) Global Flowsheet
5) Specialty Abstract
NARRATIVE REPORTS
6) Nurse/PA/PT/OT/Diet Note
7) Letter/Form/Consent
8) Psych Evaluation Note
9) Cytology Reports
10) Dischrg Sum/Admissn H&P
11) Cardiology Reports
12) Vascular Lab Reports
13) GI Procedures
14) Neurology Reports
15) Operative Report
16) Radiology & Nuclear Med
17) Surgical Pathology
18) Visit/Procedure/Gopher
19) Progress Notes
20) Reference Lab Info
21) Scanned Documents
Press F10 for more choices

USA.gov

Next stage

- We build some tools for a PHR at NLM
- Those tools evolved into LHC-Forms
  - A form generator that could make every LOINC panel into an on-the-fly data capture form.
  - It could convert a form definition into an working web form in an instant.
  - It has lots of sophisticated form capabilities including tie-ins to the most common external coding systems.

Impediments

- It was not a standard.
- Users could stitch such forms into their web pages, but it had no natural usage ecosystem.
- Real users of the form would have to figure out how to tie to a patient registry, orders, provider tables, authentication, etc. Tall barriers to its use.

Then along came FHIR

- FHIR eliminated these impediments.
- FHIR SDC is a "standard" which our tools could implement and render as executable forms.
- SMART on FHIR provided a real world ecosystem of authentication tools, patient registries etc., in which LHC forms could live and be useful. So we support SMART on FHIR and LHC-Forms lives on its App gallery - https://apps.smarthealthit.org/.
- We also built tools that can interact directly with FHIR servers and can use native Google and Facebook logons for authentication. Ye Wang will show you the details later.

NIH USA.gov

Forms definitions

- FHIR forms
  - FHIR Questionnaire - defines a simple questionnaire without special features such as skip logic.
  - FHIR SDC Questionnaire – an extension on Questionnaire, with lots of additional features like skip logic, more sophisticated error checking, etc.
- IHE
  - Has an SDC form - but it is different from, and formulated in a more complicated fashion than the FHIR SDC form https://ihe.net/uploadedFiles/Documents/QRPH/IHE_QRPH_Suppl_SDC.pdf
- LHC-Forms has a native form definition based on LOINC panels which we can transform into either of the FHIR Questionnaires and possibly into IHE questionnaire in future.

Introduction & Overview

- Goals and rationale for standards-based data capture
- LOINC panels/Forms – provide 2000 pre-built forms
- Form fields can be tied to External look up tables (coding FHIR systems/master files)
- Can define form in
  - Spreadsheet specification
  - Through form builder program (will present)
- Can generate FHIR SDC form definition
- SDC Questionnaire specification
- Can deliver completed form as
  - FHIR QuestionnaireResponse
  - FHIR DiagnosticReport

Brief over view of FHIR forms

- Questionnaire/QuestionnaireResponse
  - A resource for storing form definitions themselves
    - Questionnaire - https://www.hl7.org/fhir/questionnaire.html
    - Questionnaire, structured data capture (SDC) - https://www.hl7.org/fhir/DSTU2/sdc/questionnaire-sdc.html
  - Questionnaire response a resource for storing the content of a completed form
    - For Questionnaire - https://www.hl7.org/fhir/questionnaireresponse.html
    - For Questionnaire SDC - http://hl7.org/fhir/us/sdc/sdc-questionnaireresponse.html
- DiagnosticReport
  - With standard codes for the variables also *possible* to generate a FHIR DiagnosticReport bundle directly form data capture forms

LHC tools you will hear about today

- LHC Clinical Table Search Service (CTSS)
  - Provides Autocomplete (and soon simple Lucene Search) for commonly needed codes systems such as ICD diagnoses, US national provider files (NPI), Rx.norm drug ingredients, LOINC test and measure terms
  - Includes some secret sauce
  - Can be tied into LHC form fields, used directly off LHC web site, or tied into user application
- LHC-Forms - form definitions and the widget
- LHC-Forms Builder - a tool to create LHC-Forms
  - LHC-Forms can also be defined/edited directly in their native forms (JSON) or via a spread sheet definition
- UCUM - units validator & converter

Secret Sauce in Clinical table linker

- Via the URL, users can specify what fields in the external table are searched, and which answers are returned as part of selection grid, and which are stored in the index value field as hidden variables
- Other fields in the form can use these hidden variables as answer lists, default values, or help messages.
- Explore the tables we have created so far at:

  https://clin-table-search.lhc.nlm.nih.gov/
- Tables include IDC9, ICD10, genetic tables (ClinVar, dbSNP, etc.), RxTerms, NPI (coming soon), etc.

LHC-Forms: Short Introduction

- Strong tie to Meaningful Use standards
  - Coding systems LOINC, RxNorm, Snomed CT, Standard genetic coding systems, UCUM
  - All LOINC panels are also LHC-Forms
- Outputs : FHIR DiagnosticReport, FHIR QuestionnaireResponse, V2 message, JSON

Lots of rich survey instruments and assessments available as LOINC forms > 2000 and thus LHC forms

- Most CMS clinical forms (OASIS, MDS 3.0 nursing home assessments, dialysis reports) are represented as LOINC panels/forms.
  - CMS is unifying all of its assessments as LOINC forms with LOINC codes- some which will use the LOINC codes for existing CMS data items where possible and obtain new ones when needed.
- CDC standard forms for birth and death certificates are LOINC forms
- Lots and lots of psychosocial surveys
  - PHQ, PROMIS
  - Lots of clinical test panels and nursing assessment forms
  - Ventilator variables, 12 lead EKG, Pulmonary function tests (Mayo), cardiac echoes (DICOM), Lab panels, Ophthalmology measures (NEI), etc.

LHC-Forms Builder: Short Introduction

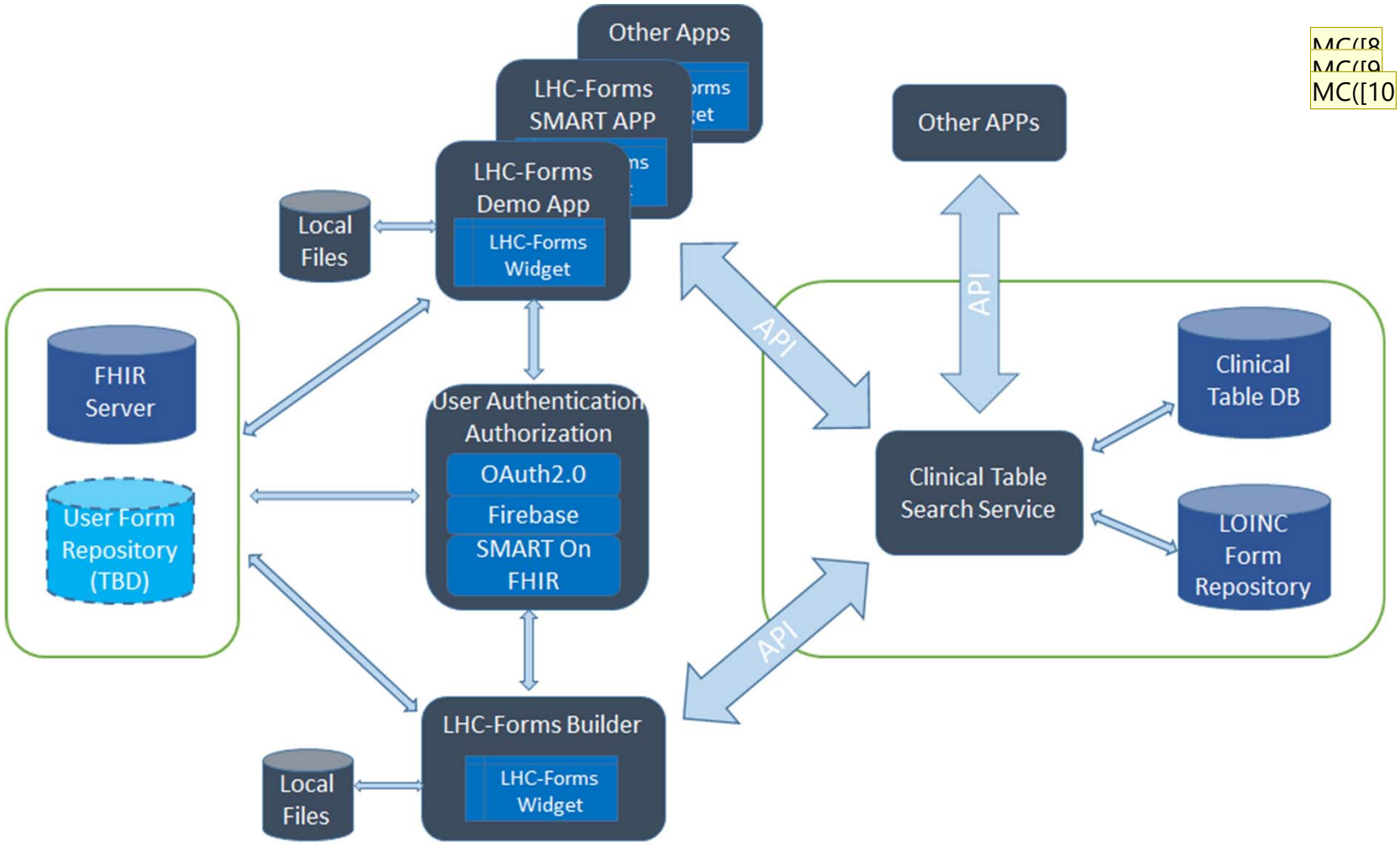
- Website: https://lhc-formbuilder.lhc.nlm.nih.gov/
- Can start with one or more existing LOINC panels, and add to or edit it.
- Can start from scratch using existing LOINC variables or user invented variable.
- JSON Schema defined for validating forms.
- Docker implementation of one of our applications, plan to support Docker for all of them.

UCUM-LHC: Unit Validation and Conversion

- For "Unified Code for Units of Measure" (UCUM)
- Website: https://ucum.nlm.nih.gov/ucum-lhc/
- Library can be downloaded from GitHub or installed with "bower"
- Library supports:
  - Validation of unit expressions
  - Conversion of values between different unit expressions
- Unit codes in UCUM are not always what one would expect, but there are synonyms
  - Some UCUM syntax is not in common vernacular, e.g. [lb_av], but synonyms ([lb_av] = pounds) will guide the users
- Some special syntax: "." = multiplication, * = exponentiation

Overall Technology

- Mixture of Web widgets and applications, namely: Clinical Table Search Service, Form Builder, UCUM validator
- All software - applications and widgets - are written in JavaScript
- Most of development uses Google AngularJS
- All data (including form definition and content of populated form) is represented as JSON
- Most content stored in Elastic Search (with Lucene) https://www.elastic.co/
- Size of LHC forms renderer that loads into browser – about 300K


Other Apps
LHC-Forms SMART APP
LHC-Forms Demo App
LHC-Forms Widget
Local Files
FHIR Server
User Form Repository (TBD)
User Authentication Authorization
OAuth2.0
Firebase
SMART On FHIR
LHC-Forms Builder
LHC-Forms Widget
Local Files
API
API
API
Other APPs
Clinical Table Search Service
Clinical Table DB
LOINC Form Repository

MC([10

Slide 23

MC([8 could you put numbers on the four things we are countnig as "tools" in the previous slide. Would like to be able to refecenc them by number. But... I only count three ALso can I say that the API is the same general structure -in all of the instances shwoing on the grap
McDonald, Clem (NIH/NLM/LHC) [E], 9/12/2017

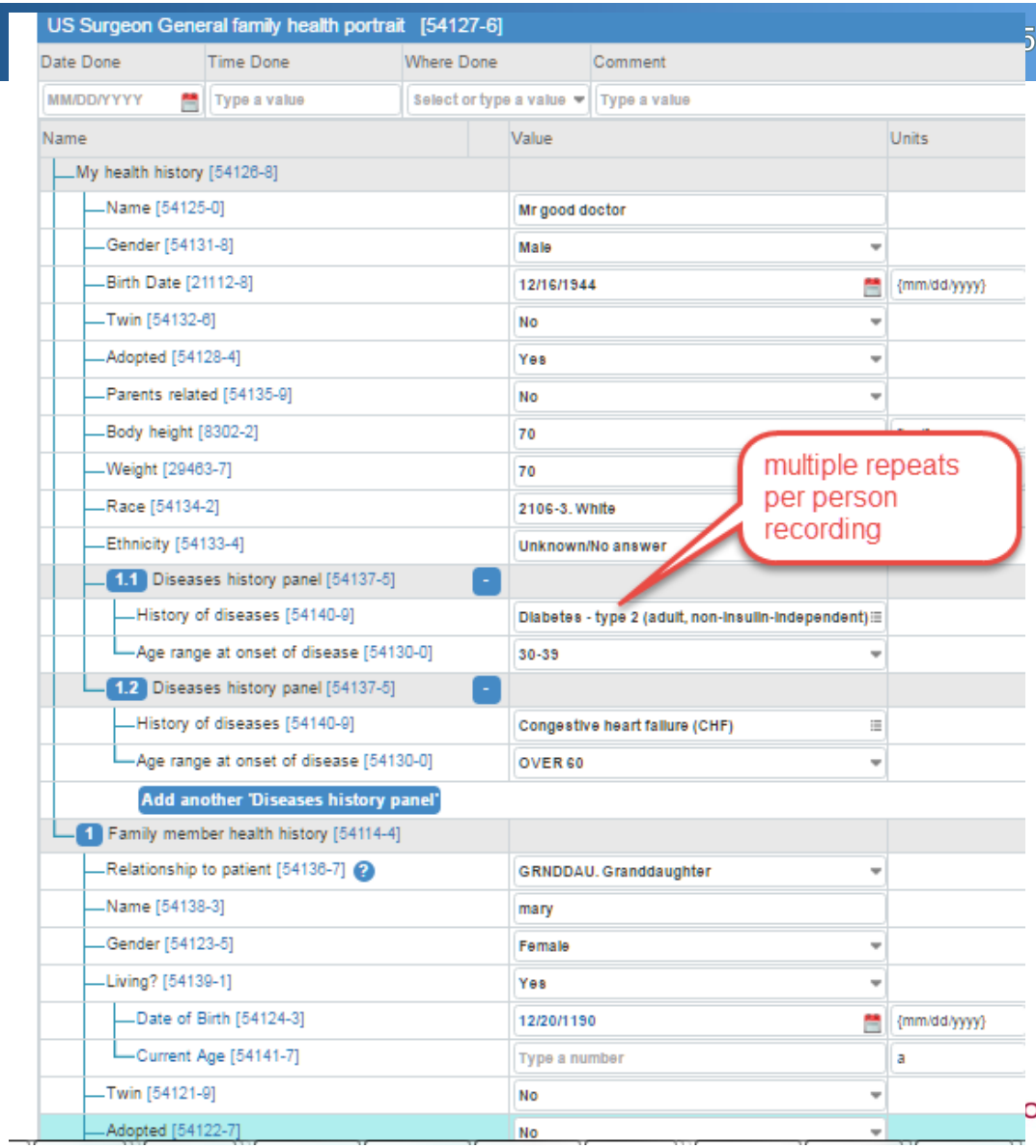
MC([9 Finally is the LOINC form repository just all of teh LOINC terms and panels or is soemthing seperate.
McDonald, Clem (NIH/NLM/LHC) [E], 9/12/2017

MC([10 Instead of other APPs better say, Your Apps or something
McDonald, Clem (NIH/NLM/LHC) [E], 9/12/2017

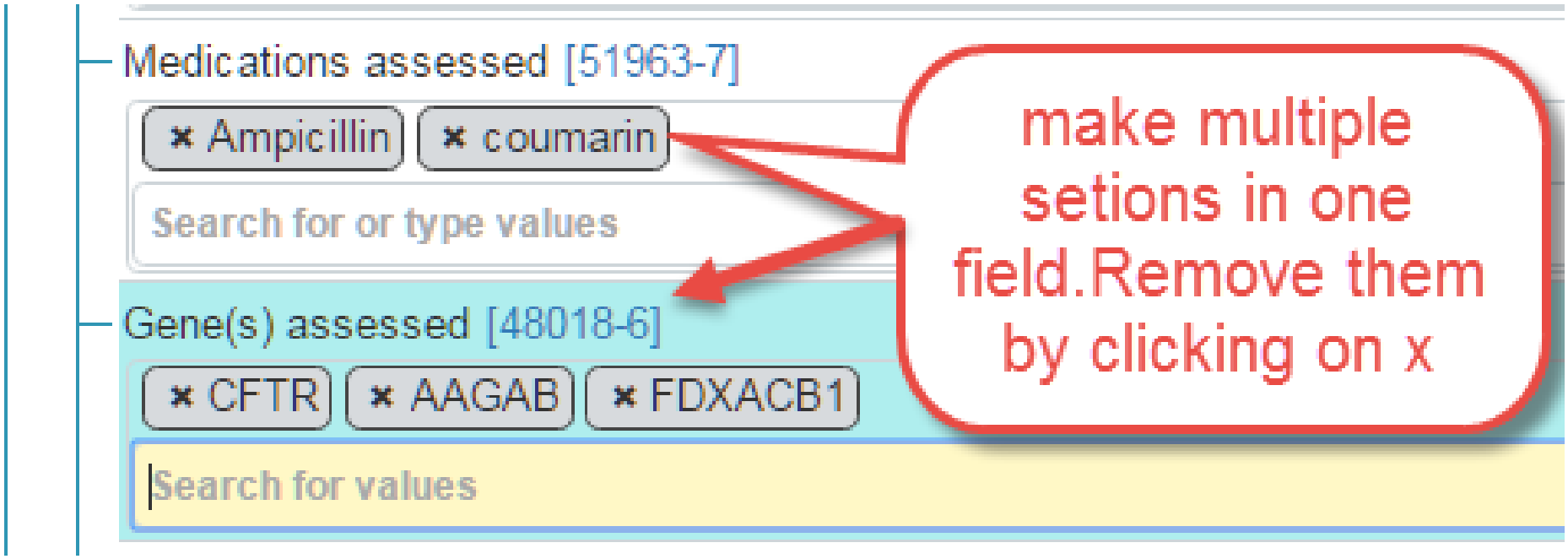
Some example forms as preview

Surgeon Generals family health history as an LHC-Form

Multiple levels of nesting -
A person can have many diagnoses with dates AND many relatives who also have many diagnoses and dates

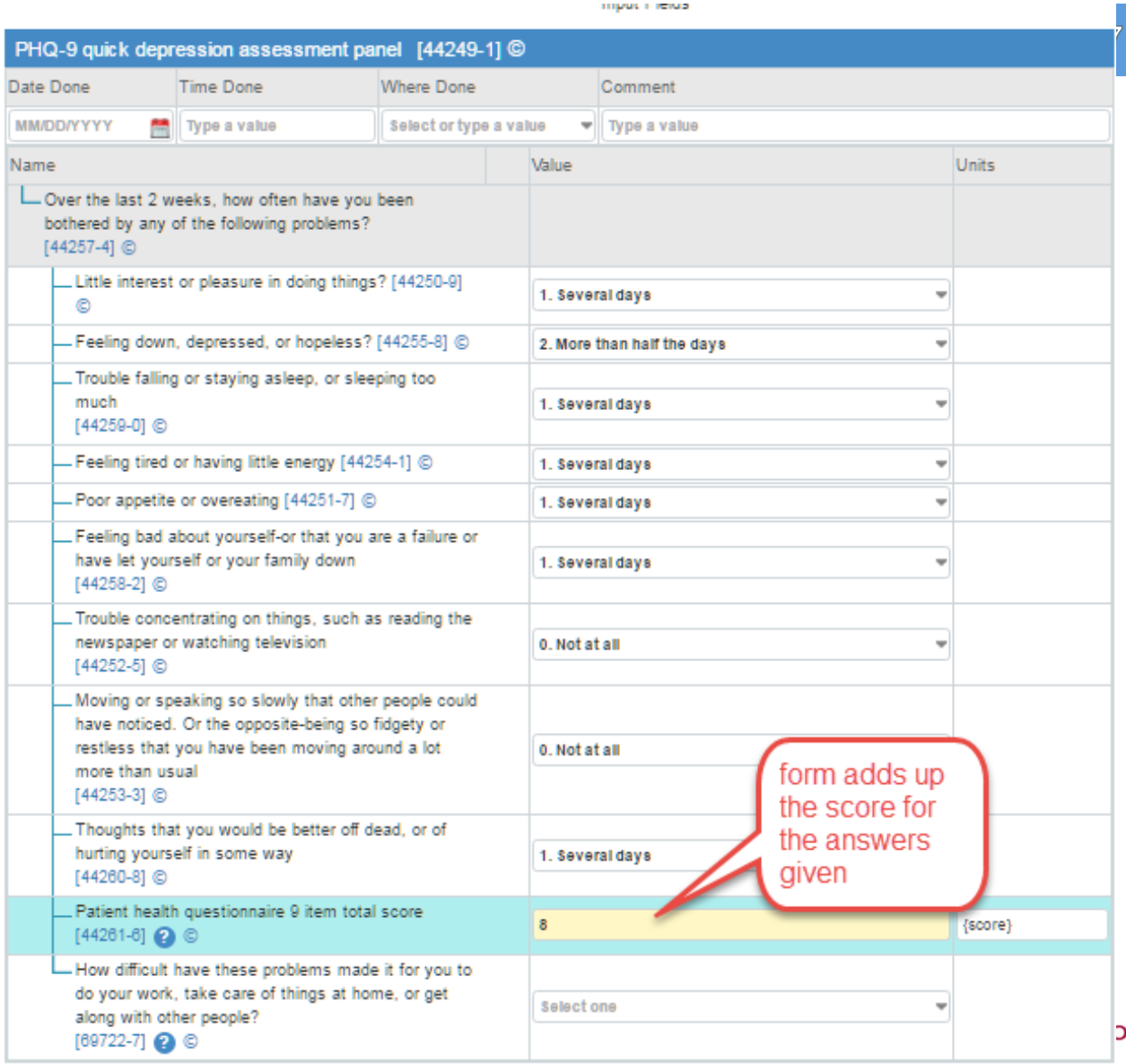


Can enter multiple fields in one field (or as separate fields illustrated with PHR)



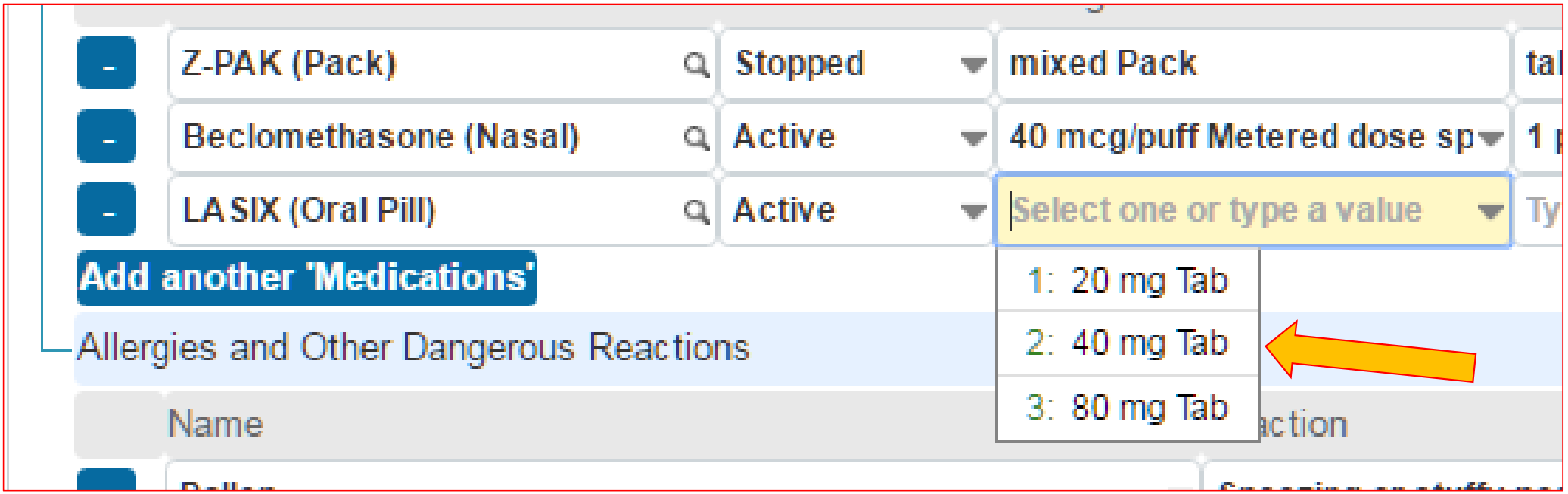
PHQ-9 Depression survey with score computed on the fly

As an LHC-Form



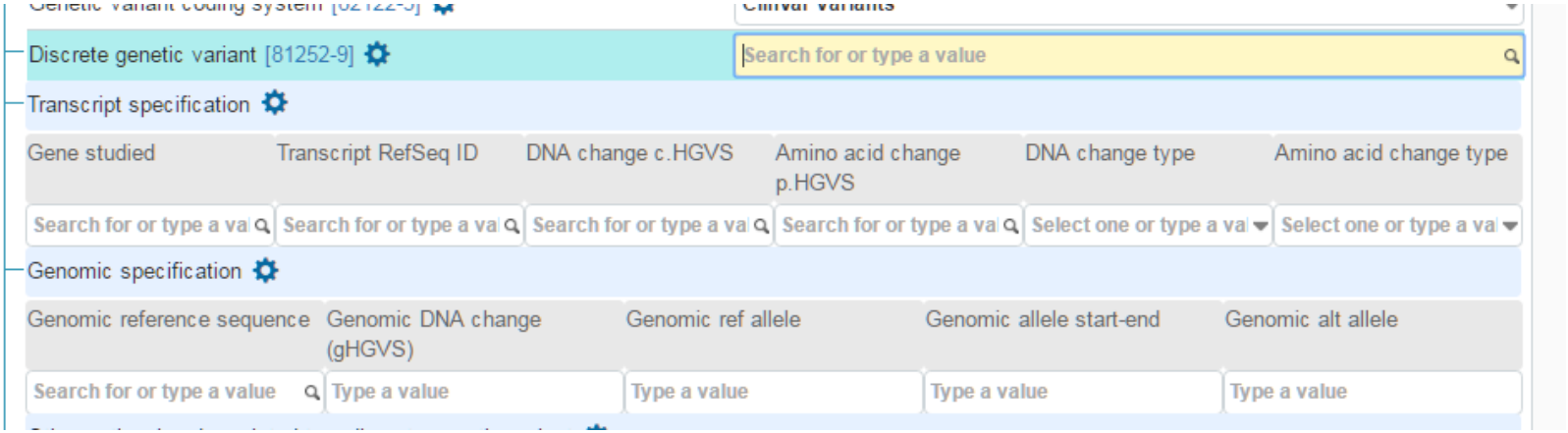
https://lhc-forms.lhc.nlm.nih.gov/

Secret sauce : Code stored in one field can generate code answer lists for succeeding fields- to be illustrated on the PHR form
Enter Lasix oral and get selection menu of available pill sizes

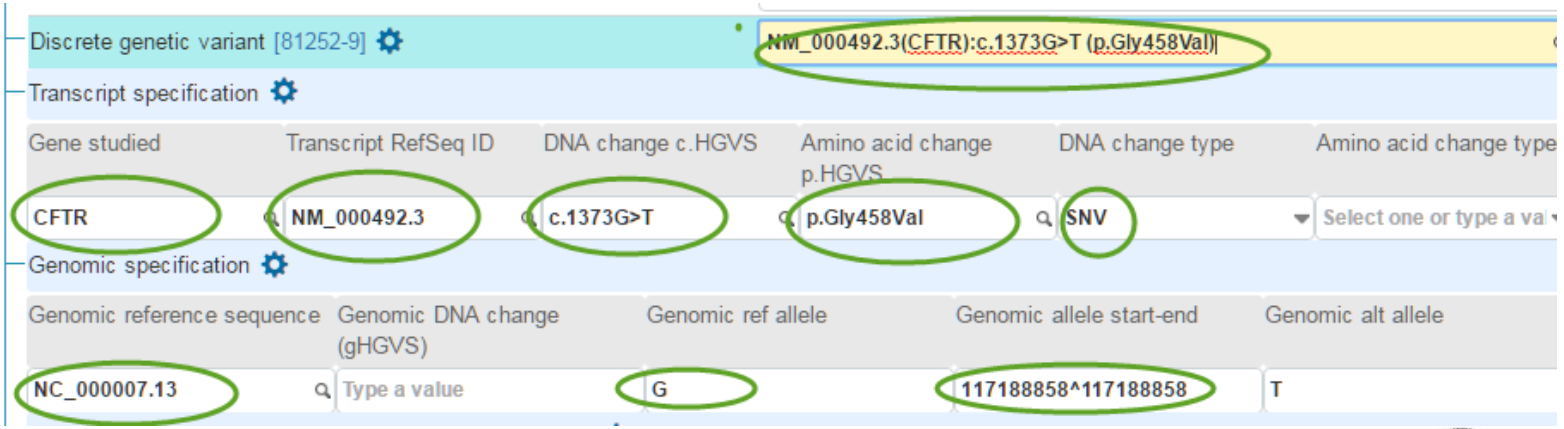


Secret sauce: record for Id in one field generates defaults for others

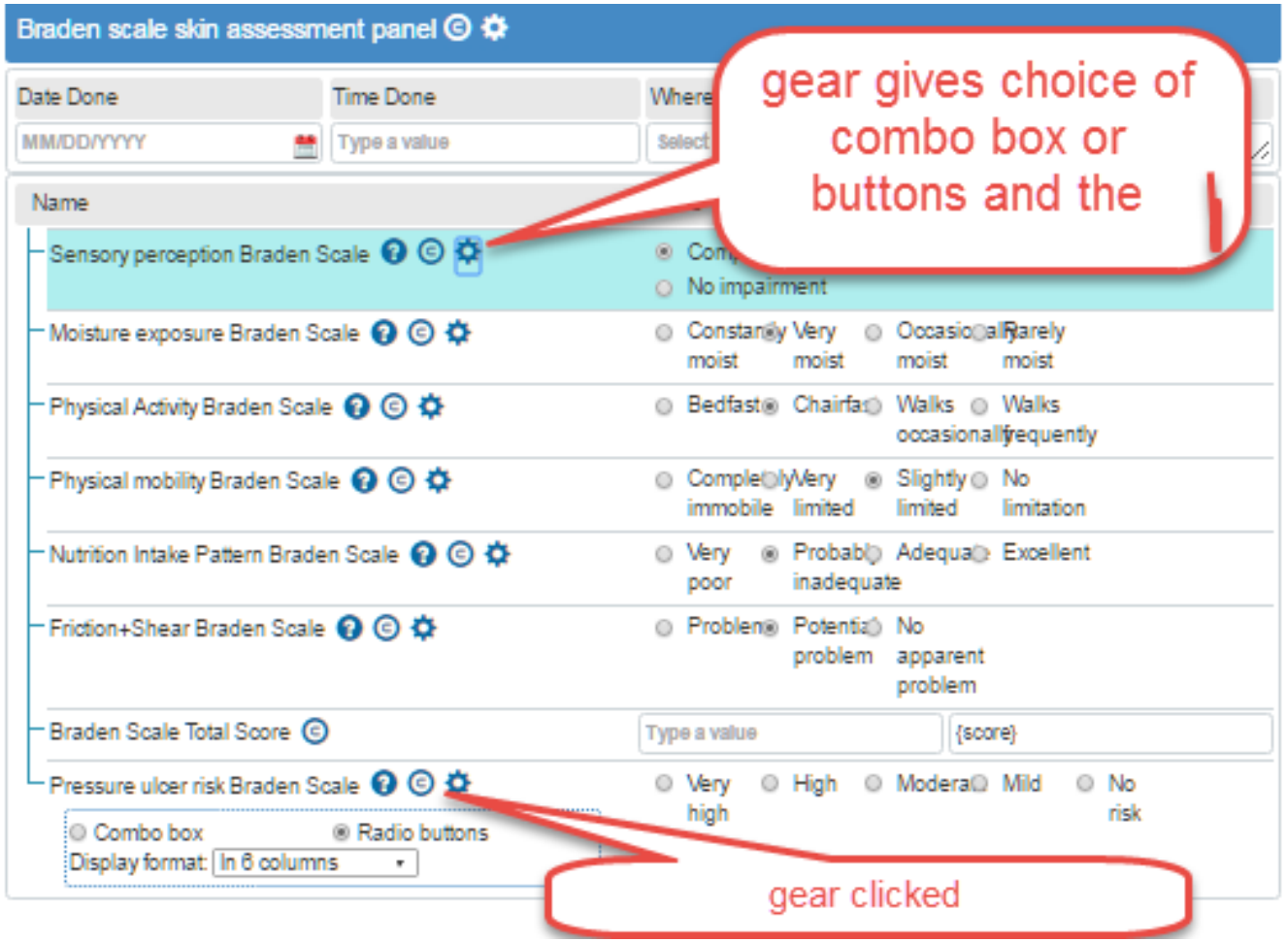
Before



After

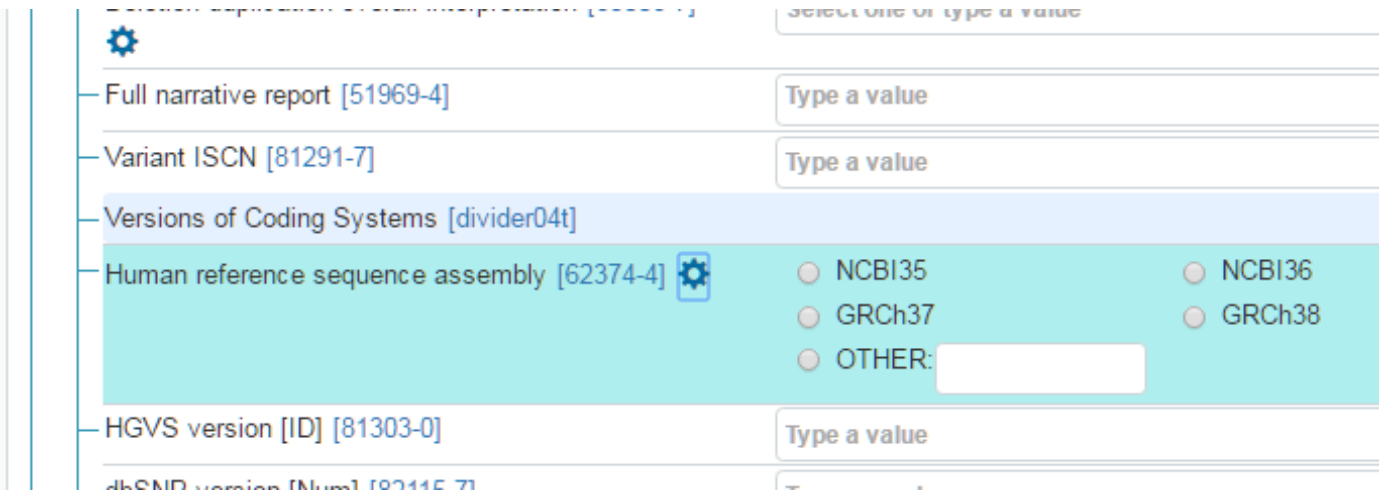


Choices of combo boxes, radio buttons, check boxes and grids when appropriate

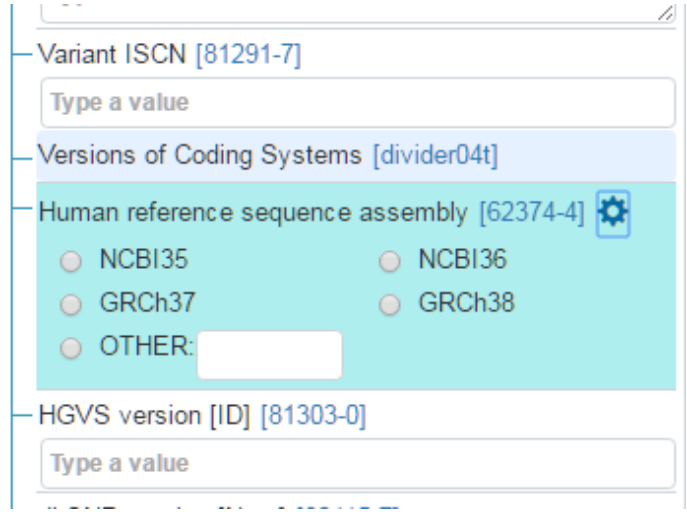


Responsive design-example from HL7 V2 genetics reporting form

Wide screen



Skinny screen



Making Applications from forms – Implemented PHR close to what it was in previous Ruby on Rails applications (will demo)

Save To File | Use "Label Above" Style | Use "Label on Left" Style | Show HL7 Message

☐ Display Question Code ☑ Show Help/Description ☐ Keyboard Navigation On Input Fields Total # of Questions: 44

Personal Health Record

Medical Conditions

Medical condition	Status	Started	Stopped	Description/Comment
Chest pain	Active	04/20/2016	MM/DD/YYYY	Sounds anginal. Worrse with exertion, but young and no family history
Pneumonia - bronchial	Inactive	03/17/2017	04/22/2016	Treated wish Zpack on ambulatorybais
Hay fever (allergic rhinitis)	Active	03/20/2012	MM/DD/YYYY	Every spring
bac	Select one or t	MM/DD/YYYY	MM/DD/YYYY	Type a value

Add another 'Medical Conditions'

Medications

Medication name	Status	Strength	Instructions	Started	Stopped	Why stopped	Resupply
Z-PAK (Pack)	Stopped	mixed Pack	take until gone	03/17/2016	04/22/2016	Finished the prescription	MM/DD/YYYY
Beclomethasone (Nasal)	Active	40 mcg/puff Metered dose sp	1 puff twice day in season	03/20/2012	MM/DD/YYYY	Select one or type a value	MM/DD/YYYY

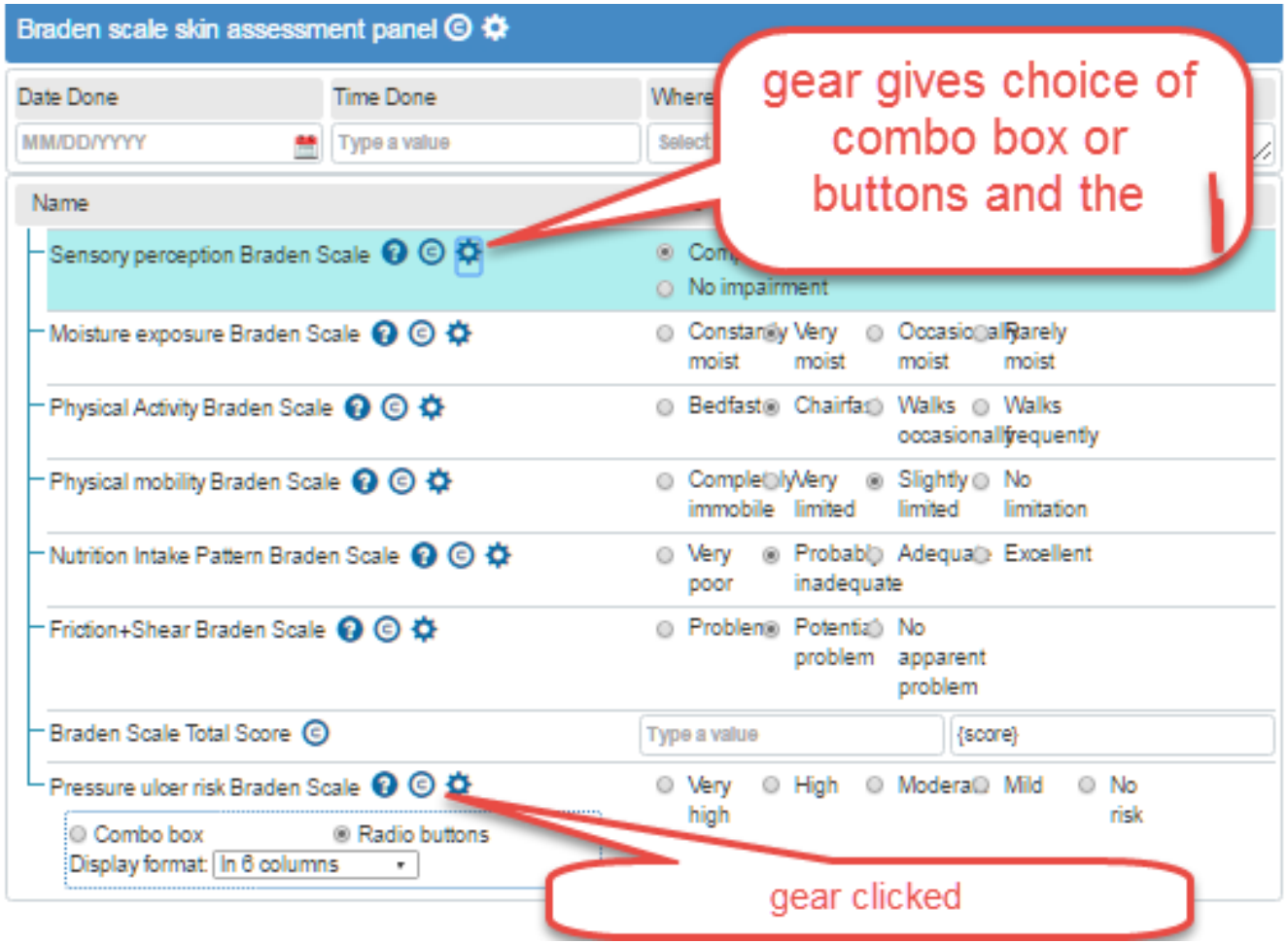
Add another 'Medications'

Allergies and Other Dangerous Reactions

Name	Reaction	Started	Comment
Pollen	Sneezing or stuffy nose	03/15/2017	Worse when maple trees bloom
Select one or type a value	Select one or type a value	MM/DD/YYYY	Type a value

Add another 'Allergies and Other Dangerous Reactions'

Choices of combo boxes, radio buttons, check boxes and grids when appropriate



HL7 message generated from completion of genetics form- only top 2/3ds showing (Hand out)

OBR|1|X1000-0^HL7 Genetic Test Panel for Variants - 20160322^LN|
OBX|1|CNE|XXXXX-12^Choose kind of mutations targeted^LN|1|C01^Simple variants^LN~C02^Complex variants^LN|
OBX|2|CNE|XXXXX-10^Choose region of interest specification^LN|1|C01^Specific targeted mutations^LN|
OBX|3|CWE|51967-8^Genetic disease assessed^LN|1|C0010674^Cystic fibrosis^LN|
OBX|4|CNE|48002-0^Genomic source class^LN|1|LA6683-2^Germline^LN|
OBX|5|CNE|48018-6^Gene(s) examined^LN|1|1884^CFTR^LN|
OBX|6|CNE|51968-6^Genetic analysis overall interpretation^LN|1|LA6576-8^Positive^LN|
OBR|2|XXXXX-9^Simple variants^LN|
OBX|1|CNE|XXXXX-5^Variant ID^LN|1.1|44487^35823^LN|
OBX|2|ST|XXXXX-13^Variant name^LN|1.1|NM_000492.3(CFTR):c.1400T>C (p.Leu467Pro)|
OBR|3|XXXX1-1^Variant specifics^LN|
OBX|1|CNE|XXXXX-19^Type^LN|1.1.1|SNP^SNP^LN|
OBX|2|CNE|48018-6^Gene^LN|1.1.1|44487^CFTR^LN|
OBX|3|CNE|48013-7^NM_RefSeq^LN|1.1.1|44487^NM_000492.3^LN|
OBX|4|CNE|41103-3^DNA change^LN|1.1.1|44487^c.1400T>C^LN|
OBX|5|CNE|48005-3^AA change^LN|1.1.1|44487^p.Leu467Pro^LN|
OBX|6|ST|XXXXX-17^NC/NG_RefSeq^LN|1.1.1|NC_000007.13|
OBX|7|ST|69547-8^Ref allele^LN|1.1.1|T|
OBX|8|ST|X0029^Allele loc^LN|1.1.1|117199525|
OBX|9|ST|69551-0^Alt allele^LN|1.1.1|C|
OBX|3|CWE|X1001-0^Cytogenetic location^LN|1.1|44487^7q31.2^LN|
OBX|4|CWE|X1002-0^Possible associated phenotype^LN|1.1|C0010674^Cystic fibrosis^LN|
OBR|4|XXXXX-9^Simple variants^LN|
OBX|1|CNE|XXXXX-5^Variant ID^LN|2.1|44494^35830^LN|
OBX|2|ST|XXXXX-13^Variant name^LN|2.1|NM_000492.3(CFTR):c.1684G>A (p.Val562Ile)|
OBR|5|XXXX1-1^Variant specifics^LN|
OBX|1|CNE|XXXXX-19^Type^LN|2.1.1|SNP^SNP^LN|
OBX|2|CNE|48018-6^Gene^LN|2.1.1|44494^CFTR^LN|
OBX|3|CNE|48013-7^NM_RefSeq^LN|2.1.1|44494^NM_000492.3^LN|
OBX|4|CNE|41103-3^DNA change^LN|2.1.1|44494^c.1684G>A^LN|
OBX|5|CNE|48005-3^AA change^LN|2.1.1|44494^p.Val562Ile^LN|
OBX|6|ST|XXXXX-17^NC/NG_RefSeq^LN|2.1.1|NC_000007.13|

Clinical Table Search Service

Presenter: Paul

Clinical Table Search Service: Introduction

- https://clin-table-search.lhc.nlm.nih.gov/
- Purpose: A web service for autocompleting selection lists used in filling in forms
  - Example use: A field with a list of drug names that autocompletes as you type
- Provides access to pre-defined 20+ data tables, e.g.:
  - Medical conditions
  - ICD-10-CM
  - Drug names (and associated strength/form lists)
  - LOINC questions (with coded answer lists, and forms)
  - UCUM (Unified Codes for Units of Measure) -- for lists of units
  - National Provider Index (doctors and practices)
  - Genetic tables (ClinVar, COSMIC, snps, etc.)

Clinical Table Search Service: Introduction

- Definition: "A **web service** is a service offered by an electronic device to another electronic device, communicating with each other via the World Wide Web." (Source: Wikipedia)

- Output is intended for programs, not people
- A program sends a URL to the web service, perhaps with a query, and gets back data (typically XML or JSON)

- For the Clinical Table Search Service (CTSS):
  - a JavaScript program in a webpage sends a URL with what the user typed, and
  - gets back a JSON structure with data for matching items that can be shown in a list and selected.

Clinical Table Search Service: Introduction

Web service example:

- URL sent by program: https://clin-table-search.lhc.nlm.nih.gov/api/conditions/v3/search?terms=arm
- Returned data (hard to read, but not by a program):

[7,["2958","29959","23374","9917","9918","9893","9909"],null,[["Arm pain"],["Shoulder or upper arm injury"],["Elbow pain"],["Shoulder strain"],["Shoulder sprain"],["Humerus fracture"],["Biceps tendon rupture"]]]

Clinical Table Search Service: Introduction

- Usages:
  1. AJAX calls from external software to the APIs for our tables
  2. LHC-Forms can use the CTSS for autocompleting lists from any of the pre-defined tables.
  3. Stand alone searches off our web site (to see what's there, using the demos on the documentation pages)

- Hands-on exercise at end of this section:
  - We'll modify some starting code to create an autocompleting field
  - We'll add some code to auto-populate an adjacent field with data from the selected item
  - If you get stuck, there is a functioning parallel example with a different API that you can modify.

Clinical Table Search Service: Introduction

- Each data table has its own set of fields
  - Example: The medical conditions table has codes, a primary name, a consumer name, and a synonyms field.
- The service allows you to specify, via the URL:
  - Which fields to search
  - Which field(s) you want returned as:
    - a code field for the record
    - fields to display in columns of the user choice grid, e.g. patient name AND address
    - extra fields of data (returned as a hash for later use, e.g. by LHC-Forms as defaults, menus, help text, etc., for subsequent fields.)
  - More on this later

Clinical Table Search Service: Introduction

- Early in its history and still a pilot
- Have not advertised or promoted
- Content is updated on a monthly basis (except dbVar)
  - Data update date for each table is shown on main page
- In use by several production websites and (probably) apps. As of 2017/12:
  - 67 referring sources (using or linking to us)
  - 12% direct access (e.g. non-web programs)

Clinical Table Search Service: Website

- https://clin-table-search.lhc.nlm.nih.gov/
- Documentation and demos of the web service
- Main page shows list of data tables
  - Clinical/administrative tables in first half
  - Genetic tables grouped together is second half of page
- Link to detailed documentation for each table on left
  - Individual page per table also defines the fields in each table and provides details for how to construct query URLs
  - Also includes information on the data source and processing we performed
- Center column of web page grid has input field for demonstrating the autocomplete search of that table plus:
  - Link to source code for demo autocompleter
  - Link to a page per table for customizing the search & display fields

U.S. National Library of Medicine

Clinical Table Search Service

A web API service for use with autocomplete-lhc & LHC-Forms

Clinical Table Search Service (formerly "lforms-service") is a web service which software programs can use for querying clinical data tables. The API for each table is designed to work with our form field autocompletion package, but can be used by other programs as well. Please let us know if there are other clinical data tables you would like to see here.

Connectable Clinical Tables

What follows is a list of currently available tables with links to instructions for the APIs for programmatic access, and an example autocompleter for each table which uses that table's API.

Some of the data sets that we serve are stable (e.g., ICD-9-CM), while others are updated more frequently (e.g., ClinVar). We strive to keep our data up-to-date with respect to the data sources; in most cases it should be less than a month behind. Please refer to the "Details" column below for more information on a specific API's data version or date.

Table Description	Try It		Details
ICD-10-CM	Code or name	Source code Customize	From CDC Data version: 2018
ICD-9-CM diagnoses	Code or name	Source code Customize	From CMS Data version: 32 (2014-10-01)
ICD-9-CM procedures	Code or name	Source code Customize	From CMS Data version: 32 (2014-10-01)
LOINC questions and forms Contains over 2000 medical forms and their associated questions)	LOINC item	Source code Customize	From LOINC. Note that use of LOINC is subject to the LOINC Terms of Use and in some cases additional copyrights. Form definitions for the forms can also be retrieved and rendered with the LHC-Forms display widget. Data version: 2.61 (2017-06-23)
Major surgeries and implants About 280, hand-edited with synonyms; a Regenstrief Institute derivative	Procedure	Source code Customize	From the NLM PHR & Regenstrief Institute Data version: 2017-10-31
Medical conditions Over 5000, hand-edited with synonyms; a Regenstrief Institute derivative	Condition	Source code Customize	From the NLM PHR & Regenstrief Institute Data version: 2017-10-31
NPI - individuals	Provider name or NPI	Source code Customize	National Provider Identifier (NPI) records for individuals, from CMS Data version: 2017-12-12
NPI - organizations	Provider name or NPI	Source code Customize	National Provider Identifier (NPI) records for organizations, from CMS Data version: 2017-12-12
Prescribable drug ingredients A subset of RxTerms/RxNorm ingredients	Ingredient	Source code	From RxTerms/RxMix Data version: 201801
RxTerms drug names & strength lists	Drug name	Source code	From RxTerms

NIH USA.gov

Clinical Table Search Service: Website

- Overview of some Data Tables
  - Most popular (December)
    - Medical conditions: Hand-edited with synonyms; a Regenstrief Institute derivative. Usage = 63%
    - ICD-10-CM: Usage = 19%
    - RxTerms: Drug names & strength lists, derived from RxNorm. Usage = 12%
    - UCUM: Unified Codes for Units of Measure. Usage = 3%
  - Also have LOINC terms, along with form definitions (more later)
    - Can retrieve answer lists for LOINC questions

Clinical Table Search Service: Medical Conditions

- Medical conditions (>5000) list originated from Regenstrief Institute
- Content selected for importance and most carry synonyms as well as associated billing codes and links to MedlinePlus
- Medical conditions table fields
  - Display-text fields: primary_name, consumer_name
  - Code fields: key_id, icd10cm_codes (CSV), icd10cm (JSON), term_icd9_code, term_icd9_text
  - word_synonyms - synonyms for any word in the term
  - synonyms - for the term as a whole
  - info_link_data - links to MedlinePlus
- Not all fields are populated for every record
  - key_id, our internal ID, exists for every record, as do the display-text fields.
  - Can specify a preferentially ordered list of codes to return

https://clin-table-search.lhc.nlm.nih.gov/apidoc/conditions/v3/doc.html

Presenter: Paul

Clinical Table Search Service: ICD-10-CM and ICD9_CM

- ICD-10-CM - a medical coding system for classifying diagnoses and reasons for visits (>71,000 terms) – Required for billing since 2015
- Not as useful for problems as the Medical conditions table because
  - Terms are sometimes quite long
  - Not consumer oriented
- ICD-9 CM –shorter table used for almost 2 decades before ICD10 introduced

https://clin-table-search.lhc.nlm.nih.gov/apidoc/icd10cm/v3/doc.html

Presenter: Paul

Clinical Table Search Service: RxTerms

- RxTerms is derived from RxNorm (>9000 terms)
  - Intended for prescription writing in two steps. Enters drug name and route- e.g., Oral penicillin, in first field, and then pick delivery form and strength in a 2nd field that displays only the choices that apply to that drug
- RxTerms table fields
  - DISPLAY_NAME - The drug name and route combination
  - STRENGTHS_AND_FORMS - A (JSON) list of strength and form combination strings (e.g., "2mg Tab") for the drug.
  - RXCUIS- These are codes for the DISPLAY_NAME + strength-form combination (so one for each STRENGTHS_AND_FORMS entry)
  - DISPLAY_NAME_SYNONYM - Synonyms for aiding searching
    - In RxTerms this is specific to a strength as well as a name, but we did not preserve that association.

https://clin-table-search.lhc.nlm.nih.gov/apidoc/rxterms/v3/doc.html

Presenter: Paul

Clinical Table Search Service: LOINC

- LOINC - a universal code system for medical tests and measurements, from Regenstrief Institute (>84,000 terms)
  - Contains coded questions with coded answers lists
  - Questions are organized into forms (e.g. lab panels), which can be retrieved as JSON structures
- LOINC table fields:
  - text - The text of the question or form name
  - LOINC_NUM - The LOINC code of the question or form
  - datatype - Indicates whether it is a list, a string, a number, etc.
  - units - For numeric data types, this will a list of units (usually just one)
  - answers - The coded answer list for list datatypes
- https://clin-table-search.lhc.nlm.nih.gov/apidoc/loinc/v3/doc.html

Presenter: Paul

Clinical Table Search Service: Demo page

- https://clin-table-search.lhc.nlm.nih.gov/demo.html
- Useful for seeing the contents of the data tables in response to autocompletion.
- Can select which fields are searched and which are displayed
- Can switch between data tables
- Shows the URL used for the selected search and display fields

Presenter: Paul

Clinical Table Search Service: Demo page

Autocompletion Demo for the Alleles API

This page allows you to try the Alleles API and see the effects of different choices for search and display fields (specified with the "sf" and "df" query string parameters, as described on the API documentation page) on returned data, which is shown in the autocompletion list below.

Example autocompleter

The following autocompleter was constructed using the fields specified below.

Search for value

> Show developer info

Database: alleles

The "Search" checkboxes control which data fields the autocompleter searches, and the "Display" checkboxes control which data fields show up in the autocompletion list. If none are checked, a default selection will be used.

Field	Search All Default	Display All Default	Description
AlternateAllele	☐	☐	The value of the AlternateAllele field in the source file.
AlternateAllele_lbl	☐	☐	The value of the AlternateAllele field in the source file, but prefixed with "Alt=".
AlleleID	☐	☐	The ID of the allele as taken from the AlleleID column of the source file.
AminoAcidChange	☐	☐	This is the amino acid change (starting with "p.") parsed from the Name field.
Chromosome	☐	☐	The chromosome number, taken from the Chromosome field in the source file, but prefixed with "chr".
ChromosomeAccession	☐	☐	The chromosome accession number, taken from the ChromosomeAccession field in the source file.
Cytogenetic	☐	☐	The cytogenetic location of the allele, taken from the "Cytogenetic" field in the source file.
dbSNP	☐	☐	The "rs" ID number from dbSNP, taken from the "RS# (dbSNP)" field in the source file.
GeneID	☐	☐	The gene ID from NCBI's gene database.
GeneSymbol	☐	☐	This is the GeneSymbol field listed in the source file. It is the symbol for the gene that overlaps the variant.
GenomicLocation	☐	☐	This is an HL7-style concatenation of the Start and Stop fields, i.e., Start^Stop.
hgnc_id	☐	☐	A unique ID provided by the HGNC for each gene with an approved symbol. Although standard HGNC IDs are of the format HGNC:n, where n is a number, we have removed the "HGNC:" prefix, so that these values are just numbers.
HGVS_c	☐	☐	The "HGVS (c.)" field from the source file. (The "RefSeq cDNA-based HGVS expression".)
HGVS_p	☐	☐	The "HGVS (p.)" field from the source file. (The "RefSeq protein-based HGVS expression".)
Name	☐	☐	This is the "Name" field (a description of the allele) from the source file.

USA.gov

Clinical Table Search Service: Demonstrations

- https://clin-table-search.lhc.nlm.nih.gov/
- Example query for the medical conditions API
  - Base URL for API + "terms" parameter with user input

https://clin-table-search.lhc.nlm.nih.gov/api/conditions/v3/search?terms=heart

  - Format of URL is the same for each data table
  - URL contains an API version number to provide stability for existing users when we change features.

Presenter: Paul

Clinical Table Search Service: Demonstrations

- Example output for the medical conditions API
  https://clin-table-search.lhc.nlm.nih.gov/api/conditions/v3/search?terms=heart
- Returns a JSON structure, designed for compactness

Available result count

Array of condition codes

Placeholder for extra data about each record

- [32,["2212","365","2895","8411","3978","3867","8777"],null,[["Coronary artery disease (CAD)"],["Congestive heart failure (CHF)"],["Palpitations"],["Atrial myxoma"],["Heart murmur"],["Heart attack (myocardial infarction)"],["Sinoatrial block"]]]

Array of display strings, corresponding to codes

Clinical Table Search Service: Configuration

- API Parameters
  - terms (required, added by autocompleter)
  - maxList (desired number of results)
  - sf = “search fields”
  - df = “display fields”
  - cf = “code field”
  - ef = “extra fields”
  - q = a query string to further constrain results (Lucene syntax)

Presenter: Paul

Clinical Table Search Service: Configuration

- Example using query parameters:

https://clin-table-search.lhc.nlm.nih.gov/api/conditions/v3/search?sf=consumer_name,synonyms&df=consumer_name,icd10cm_codes&terms=ar

search query (autocompletion)

search fields

display fields (to be shown to the user)

- When not supplied, default settings are used for sf & df

Clinical Table Search Service: Autocomplete-lhc

- Autocompletion package written for the format output by CTSS
  - https://lhncbc.github.io/autocomplete-lhc/
  - Can also use CTSS with other autocompleters or on its own
  - Advantage: No need to write code to parse the output
  - Pre-built, ready-to-use version hosted on CTSS website:
    - https://clin-table-search.lhc.nlm.nih.gov/autocomplete-lhc-versions
- Demo of RxTerms API & autocomplete-lhc
  - https://plnkr.co/edit/sVFMu8v3ZINJMMu6bfhW
  - Uses autocomplete-lhc hosted on CTSS website
  - Just a few lines of code create an autocompleting field hitting CTSS

Presenter: Paul

Clinical Table Search Service: Autocomplete-lhc

- Features (demo at http://lhncbc.github.io/autocomplete-lhc/)
  - Built-in notion of coded list items (but codes are optional)
  - Provides methods for retrieving the "extra" data for a selected item
  - Optionally numbered lists, and can pick by number
  - Longer lists handled with a "see more" link and two-column flow
  - Supports a table-format (multi-fielded list items) with column headings
  - Long lists can be broken up with non-clickable heading items
  - Comes with an AngularJS directive
  - Accessible to screen readers

Presenter: Paul

Clinical Table Search Service: Exercise

- Build a units (UCUM) lookup showing the unit code and its name.
- Template: https://jsfiddle.net/lforms/rgt28krv/
- Steps:
  1. Look at example of an NPI lookup: https://jsfiddle.net/lforms/01gwLmpb/
  2. Look at https://clin-table-search.lhc.nlm.nih.gov/apidoc/ucum/v3/doc.html for field names for the UCUM code and name.
- Extra credit: Fill in a form field for the unit's "LOINC property"
  1. Look at an NPI example: https://jsfiddle.net/lforms/fa96usvf/
  2. Add an input field for "LOINC property"
  3. Add a the listener for selection events on the unit field to assign the corresponding loinc_property value to its input field

Clinical Table Search Service: Exercise

- Answers:

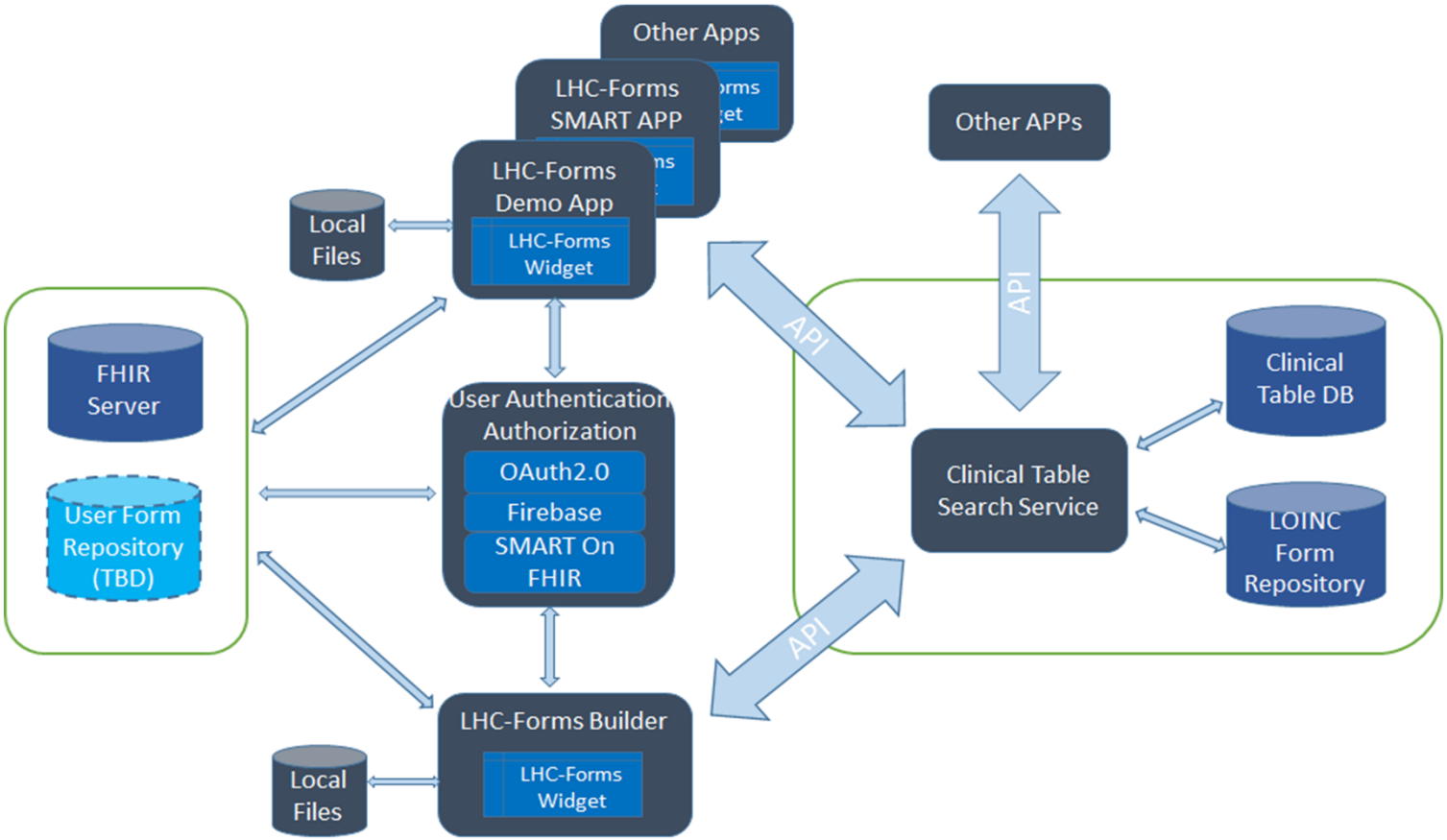
https://jsfiddle.net/lforms/eutkj3fs/

LHC-Forms: An Intelligent Form Widget

- A widget developed in JavaScript for web, mobile and desktop Apps
  - AngularJS
  - NodeJS
  - JSON
- Advanced features
  - Skip Logic
  - Repeats
  - Display control
  - Search field/autocomplete
  - Formula/calculation method
  - Data restrictions/validations
  - Data control
  - Customizable templates
- Data exports
  - HL7 v2
  - FHIR DiagnosticReport
  - FHIR Questionnaire
  - FHIR QuestionnaireResponse

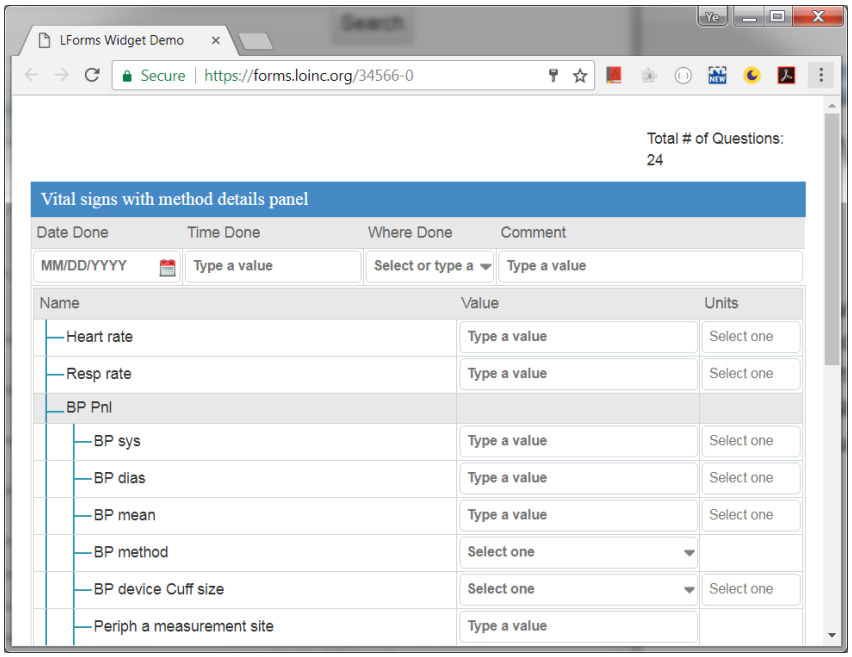
Presenter: Ye

LHC-Forms: System Design



Five sites where LHC-Forms is implemented

Regenstrief LOINC website (1)

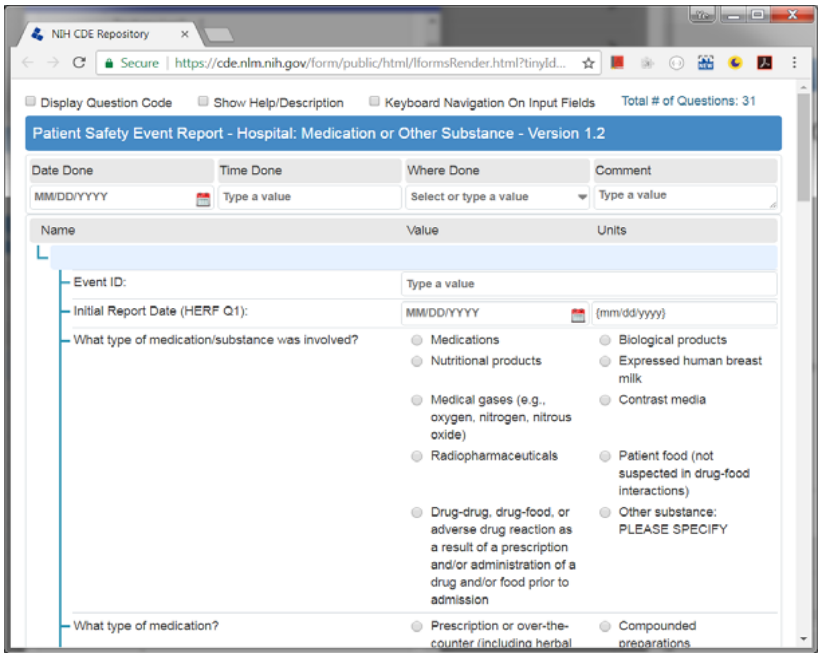


https://search.loinc.org

- Logical Observation Identifiers Names and Codes (LOINC) is a common language (a set of identifiers, names, and codes) for identifying health measurements, observations, and documents.
- Clicking on a link button for LOINC items that are forms or panels yields a LHC form based on that panel
- Login is required.

Presenter: Ye

NIH common data elements (CDE) website (2)

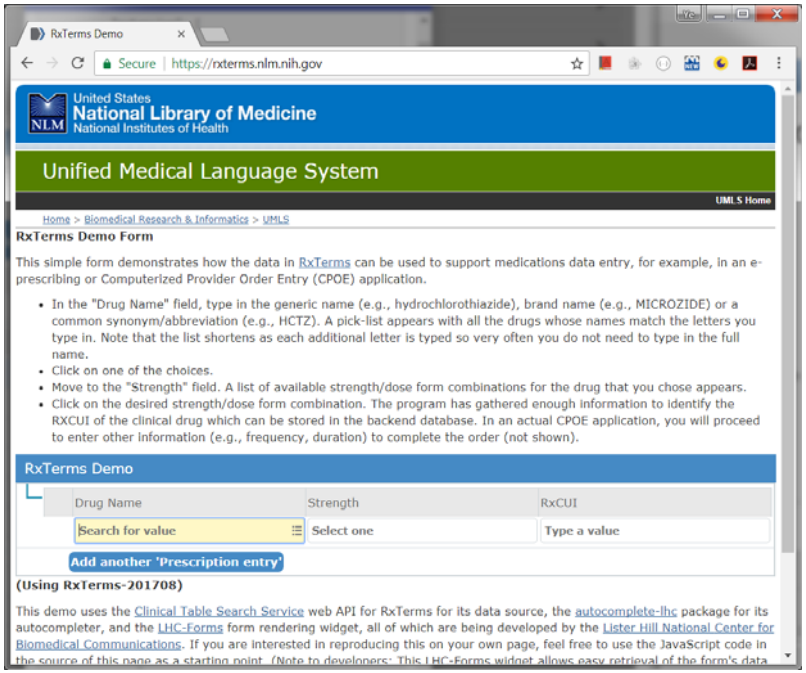


https://cde.nlm.nih.gov

- The NIH Common Data Elements (CDE) Repository has been designed to provide access to structured human and machine-readable definitions of data elements.
- Forms at CDE has an option to be displayed by LHC-Forms widget.

Presenter: Ye

NLM's RxTerms website (3)

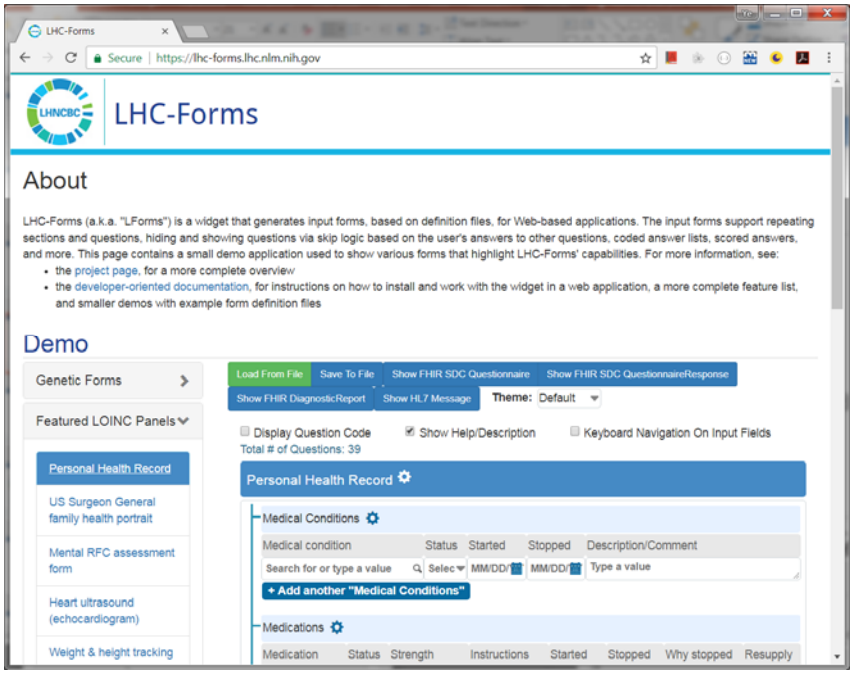


https://rxterms.nlm.nih.gov

- RxTerms is a drug interface terminology derived from RxNorm (the U.S. terminology standard for clinical drugs) which can be easily used for prescription writing or medication history recording.
- A demo application for searching RxTerms is implemented in LHC-Forms

Presenter: Ye

LHC's website (4)

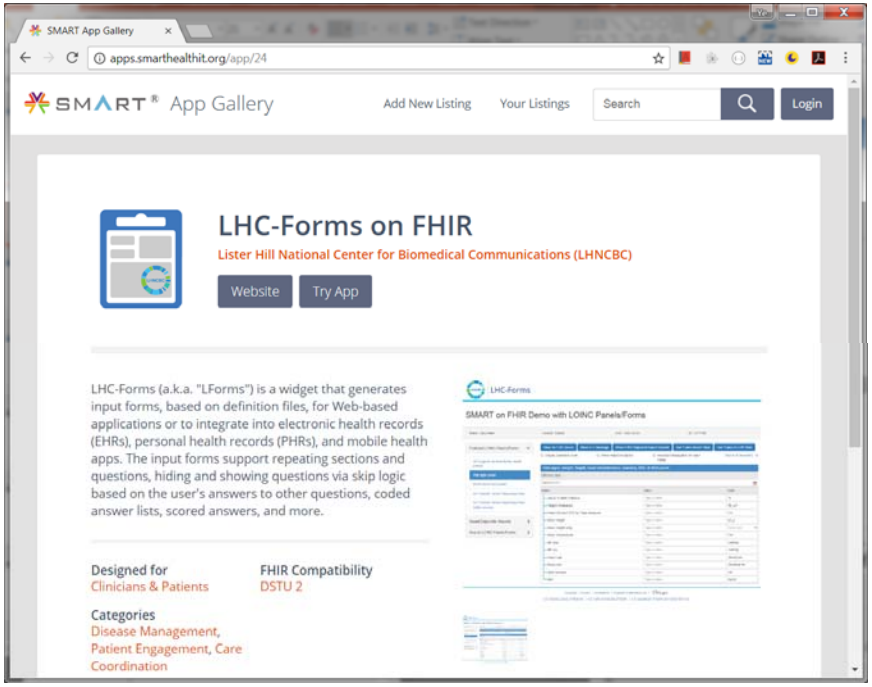


https://lhc-forms.lhc.nlm.nih.gov

- A public demo site that contains an application to show various forms that highlight LHC-Forms' capabilities.
- A demo application (/demo-app) utilizes a HAPI FHIR server and Google Firebase service.

Presenter: Ye

SMART on FHIR platform (5)



http://apps.smarthealthit.org/

- SMART Health IT is an open, standards based technology platform that enables innovators to create apps that seamlessly and securely run across the healthcare system (https://smarthealthit.org/an-app-platform-for-healthcare/about/)
- A registered app utilizes SMART On FHIR platform.

Presenter: Ye

LHC-Forms in operation

- Surgeon General's Family History
- PHQ-9
- Personal Health Record
- Clinical genomics HL7 V2 report
- Large CMS form

LHC-Forms and FHIR

LHC-Forms and FHIR Questionnaire

- Most common features are compatible.
  - Form structure, code, name, data types and etc.
  - Most FHIR Questionnaire resources could be loaded into LHC-Forms widget and displayed as actionable forms
  - LHC-Forms form definition data can be exported as FHIR Questionnaire
  - And User data can be exported as FHIR QuestionnareResponse
- LHC-Forms has some advanced features
  - Skip logic, Display control and etc. support more functions
  - Formula, Data control and etc. are not available in FHIR Questionnaire yet.
  - Would lead to some extensions in Questionnaire
- There are a list of fields in Questionnaire are not yet supported in LHC-Forms

LHC-Forms Features (1)

Features	LHC-Forms	Questionnaire	SDC Questionnaire
Name	name, question	name, title; text	name, title; text
Code	code, codeSystem; questionCode, questionCodeSystem	code (Coding)	code (Coding)
Copyright	copyrightNotice	copyright	copyright
Section/Item	item.items	item.item	item.item
Question cardinality	questionCardinality	-	questionnaire-minOccurs, questionnaire-maxOccurs
Question repeats	questionCardinality	repeats	repeats
Answer cardinality	answerCardinality	-	-
Answer repeats	answerCardinality	-	-
Answer required	answerCardinality	required	required

LHC-Forms Features (2)

Features	LHC-Forms	Questionnaire	SDC Questionnaire
Answer list	answers	option, options	option, options
Search field/autocomplete	externallyDefined	options	options
Data types	dataType	type	type
Unit list	units	-	questionnaire-unit
Readonly/editable	editable	readOnly	readOnly
Skip logic	skipLogic	enabledWhen	enabledWhen
Coding instructions/helps	codingInstructions, codingInstructionsFormat	-	entryFormat
Calculation method/formula	calculationMethod	-	-
Default/Initial value	defaultAnswer	initial[x]	initial[x]

LHC-Forms Features (3)

Features	LHC-Forms	Questionnaire	SDC Questionnaire
Display control	displayControl	-	questionnaire-itemControl, questionaire-choiceOrientation
Data control	dataControl	-	-
Restrictions/validations	restrictions (minExclusive, minInclusive, maxExclusive, maxInclusive, length, minLength, maxLength, pattern)	maxLength	maxLength, minLength, regex, minValue, maxValue, maxDecimalPlaces, maxSize,
Display template	template, templateOptions	-	-

LHC-Forms specs:
https://github.com/lhncbc/lforms/blob/master/form_definition.md

Questionnaire Fields not supported in LHC-Forms

- Form Level:
  - experimental
  - purpose
  - lastReviewDate
  - useContext
  - contact
  - publisher
  - approvalDate
  - effectivePeriod
  - jurisdiction
- Item Level:
  - definition
  - prefix

Data Types

Data Types	QuestionnaireItemType	LHC-Forms dataType	
Group	group	SECTION	
Display	display	TITLE	
Question	question	(item)	
Boolean	boolean	BL	not fully supported yet
Decimal	decimal	REAL	
Integer	integer	INT	
Date	date	DT	
Date Time	dateTime	DTM	not supported yet
Time	time	TM	
String	string	ST	
Text	text	TX	
Url	url	URL	
Choice	choice	CNE	
Open Choice	open-choice	CWE	
Attachment	attachment	BIN	not supported yet

USA.gov

Additional Data Types in LHC-Forms

Data Types		**LHC-Forms dataType**	
Numeric Range		**NR**	
Date - Year		**YEAR**	
Date - Month		**MONTH**	
Date - Day		**DAY**	
Email		**EAMIL**	
Phone		**PHONE**	
Ratio		**RTO**	not supported yet

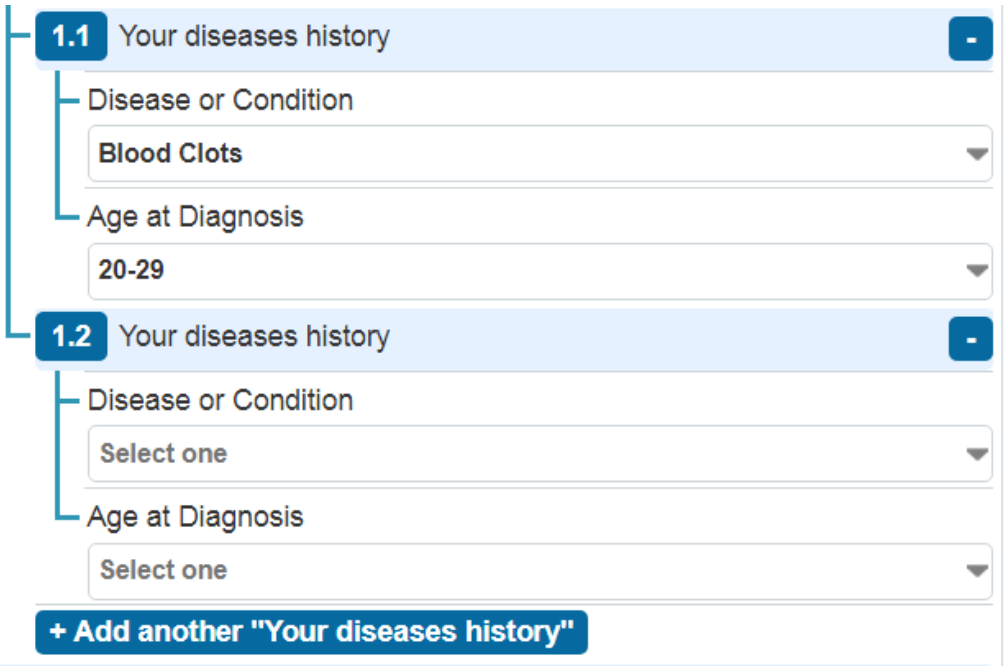
LHC-Forms Advanced Features

LHC-Forms Advanced Features

- http://lhncbc.github.io/lforms/demos.html
  - Repeats
  - Skip Logic
  - Formula/calculationMethod
  - Restriction/validation
  - dataControl
  - displayControl
  - Autocomplete --Search field & Answer list
  - template & templateOptions

Presenter: Ye

LHC-Forms Features: Repeats (1)



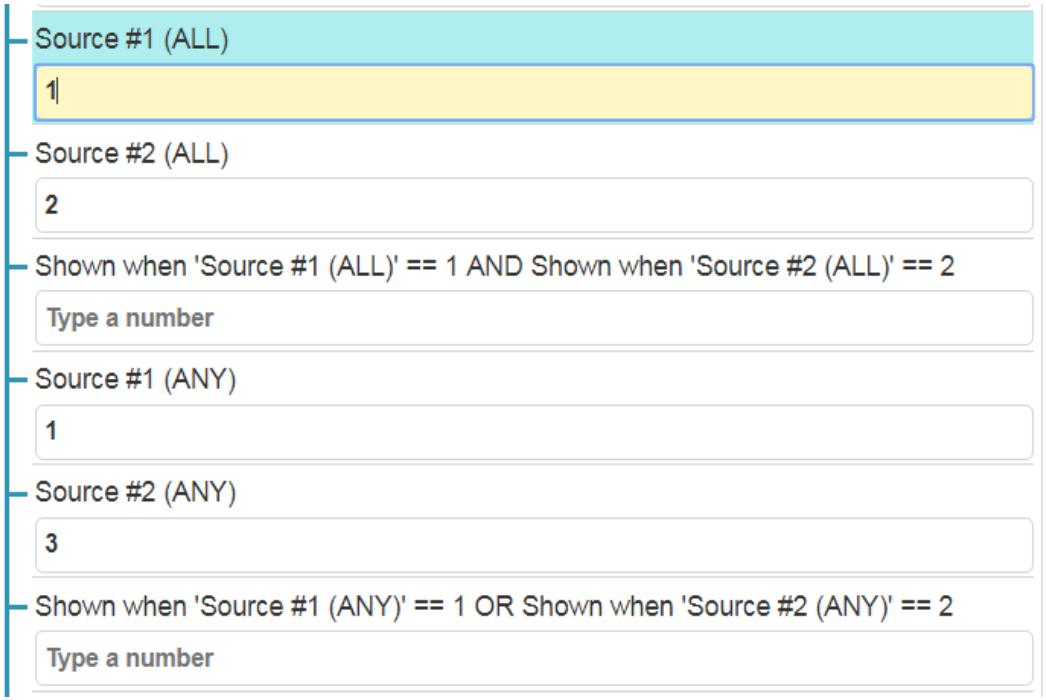
- A section or an item could have multiple instances on the form.

LHC-Forms Features: Repeats (2)

```
{
   questionCardinality": {"min": "1", "max": "*"}
}
```

Presenter: Ye

LHC-Forms Features: Skip Logic (1)



- A feature that controls if an item or a section is shown on a form based on the current values of other items.
- Support <, >, =, >=, <=
- Support AND, ANY

Presenter: Ye

LHC-Forms Features: Skip Logic (2)

```
// skip logic, with logic ALL
{ "questionCode": "slALLSource1", "dataType": "INT", "question": "Source #1
(ALL)" },
{ "questionCode": "slALLSource2", "dataType": "INT", "question": "Source #2
(ALL)" },
{ "questionCode": "slALLTargetItem", "dataType": "INT", "question": "Shown
when 'Source #1 (ALL)' == 1 AND Shown when 'Source #2 (ALL)' == 2 ",
     "skipLogic": {
        "conditions":[
            {"source": "slALLSource1", "trigger": {"value": 1}},
            {"source": "slALLSource2", "trigger": {"value": 2}}],
        "action": "show",
        "logic": "ALL"},
},
```

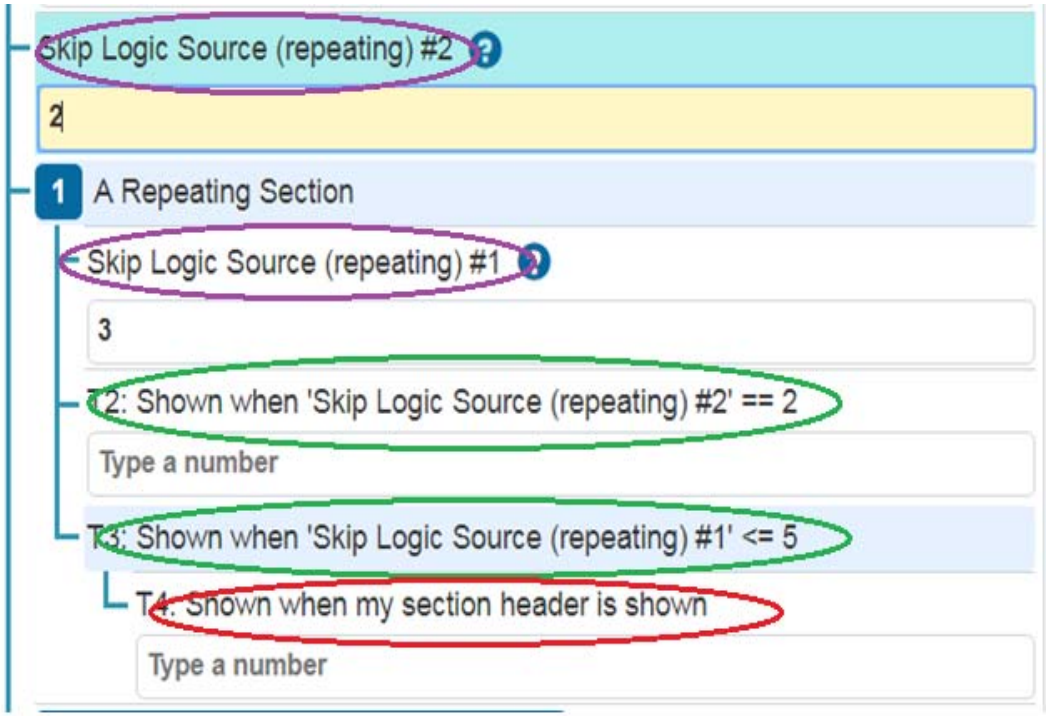
Presenter: Ye

LHC-Forms Features: Skip Logic (3)

```
// skip logic, with logic ANY
{ "questionCode": "slANYSource1", "dataType": "INT", "question": "Source #1
(ANY)" },
{ "questionCode": "slANYSource2", "dataType": "INT", "question": "Source #2
(ANY)" },
{ "questionCode": "slANYTargetItem", "dataType": "INT", "question": "Shown
when 'Source #1 (ANY)' == 1 OR Shown when 'Source #2 (ANY)' == 2 ",
     "skipLogic": {
        "conditions":[
            {"source": "slANYSource1", "trigger": {"value": 1}},
            {"source": "slANYSource2", "trigger": {"value": 2}}],
        "action": "show",
        "logic": "ANY"},
}
```

Presenter: Ye

LHC-Forms Features: Skip Logic (4)



- Decedents are controlled by ancestors or siblings.
- Children cannot control parents.
- If a parent item is shown, all decedents are shown, unless they are controlled by other skip logic rules.

Presenter: Ye

LHC-Forms Features: Skip Logic (5)

```
“items”: [
   { "questionCode": "X-001",
     "question": "Favorite dessert (try ice cream)",
     "dataType": "ST" },
   { "questionCode": "X-002",
     "question": "Ice cream flavor?",
     "dataType": "ST",
     "skipLogic": {
        "conditions":[
            {"source": "X-001",
             "trigger": {"value": "ice cream"}
            }],
        "action": "show"}
   }]
```

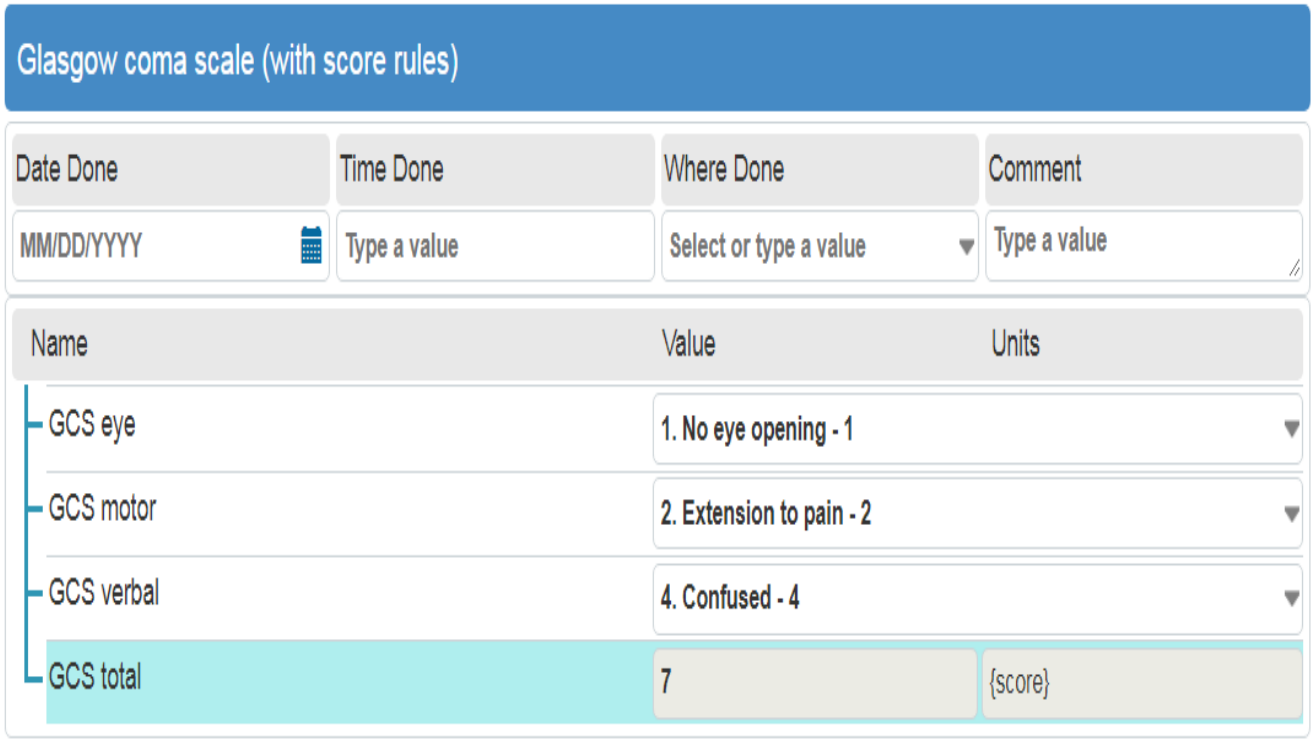
Presenter: Ye

LHC-Forms Features: Skip Logic (6)

```
{ "questionCode": "X-006",
  "question": "What is your favorite food? (shown if previous is > 10)",
  "dataType": "REAL",
  "skipLogic": {
     "conditions":[{
        "source": "X-005",
        "trigger": {
            "minExclusive": "10"}
        }],
     "action": "show"}
}
```

Presenter: Ye

LHC-Forms Feature: calculationMethod (1)



- Pre-built Functions, such as TOTALSCORE, BMI.
- Calculate value for an item based on other items' values

LHC-Forms Features: calculationMethod (2)

```
{ "questionCode": "9270-0",
  "dataType": "CNE",
  "question": "GCS verbal",
  "answers": [
    {"label": "1", "code": "LA6557-8", "text": "No verbal response (>2 yrs); no vocal response (<=2 yrs)",
"score": 1, "other": null},
    {"label": "2", "code": "LA6558-6", "text": "Incomprehensible sounds", "score": 2, "other": null},
    {"label": "3", "code": "LA6559-4", "text": "Inappropriate words", "score": 3, "other": null},
    {"label": "4", "code": "LA6560-2", "text": "Confused", "score": 4, "other": null},
    {"label": "5", "code": "LA6561-0", "text": "Oriented", "score": 5, "other": null}
  ]
},
{ "questionCode": "9269-2",
  "question": "GCS total",
  "calculationMethod": {
      "name": "TOTALSCORE"}
}
```

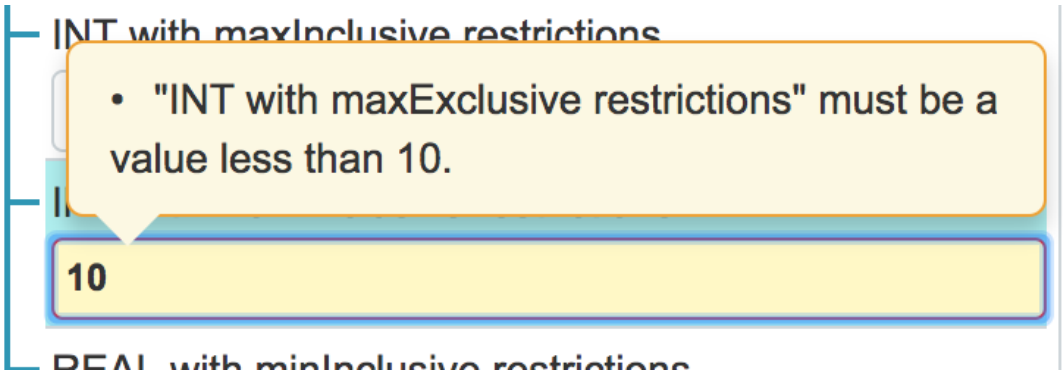

NIH USA.gov

LHC-Forms Feature: calculationMethod (3)

```
{ "questionCode": "8302-2",
  "question": "Height",
  "dataType": "REAL",
  "units": [
     {"name": "inches", "default": true}, {"name": "centimeters"}]
},
{ "questionCode": "29463-7",
  "question": "Weight",
  "dataType": "REAL",
  "units": [
     {"name": "lbs", "default": true}, {"name": "kgs"}]
},
{ "questionCode": "39156-5",
  "question": "Body mass index (BMI) [Ratio]",
  "calculationMethod":{
     "name":"BMI",
     "value":["29463-7","8302-2"]}
}
```


NIH USA.gov

LHC-Forms Feature: restrictions/validation (1)



- Validation is applied to dataTypes automatically.
- Additional restrictions are available to check numeric values, string size, and patterns.

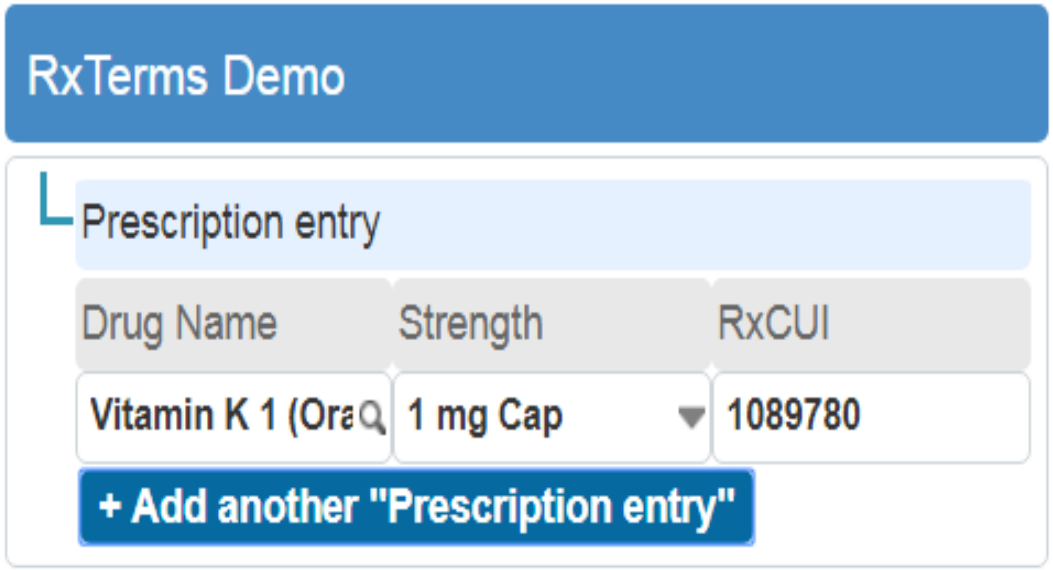
Presenter: Ye

LHC-Forms Features: restrictions/validation (2)

```
{ "questionCode": “int1",
  "dataType": “INT",
  "question": “INT with maxExclusive restrictions",
  “restrictions”: {“maxExclusive”: 10}
}
```

Presenter: Ye

LHC-Forms Features: dataControl (1)



- An attribute of an item, such as “answers”, “units”, tool tips, could be reset based on another item’s value.
- It supports a direct copy of the attribute value, new construction of an object or an array.

Presenter: Ye

LHC-Forms Features: dataControl (2)

```
"dataControl": [{
    "source": {
        "sourceType": "INTERNAL",
        "sourceItemCode": "nameAndRoute"
    },
    "construction": "ARRAY",
    "dataFormat": {
        "code": "value.data.RXCUIS",
        "text": "value.data.STRENGTHS_AND_FORMS"
    },
    "onAttribute": "answers"
  }]
```

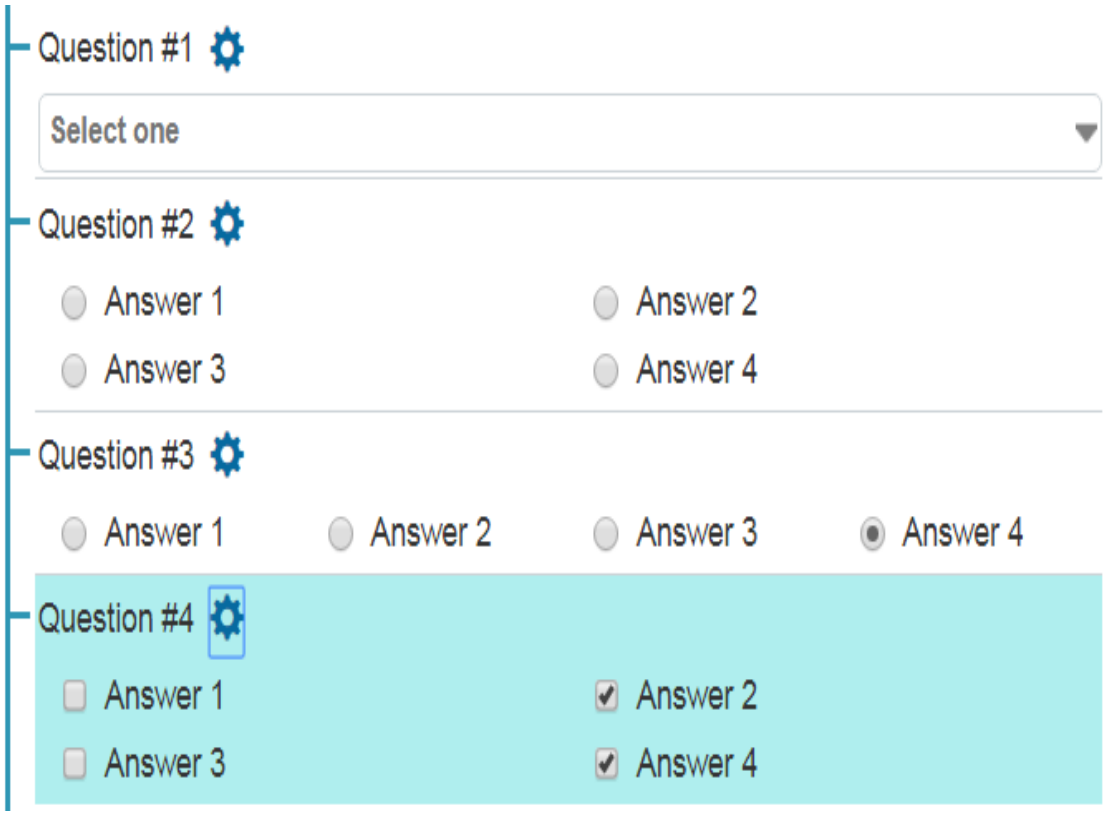
Presenter: Ye

LHC-Forms Features: dataControl (3)

```
"dataControl": [{
    "source": {
        "sourceType": "INTERNAL",
        "sourceItemCode": "strengthAndForm"
    },
    "construction": "SIMPLE",
    "dataFormat": "value.code",
    "onAttribute": "value"
  }]
```

Presenter: Ye

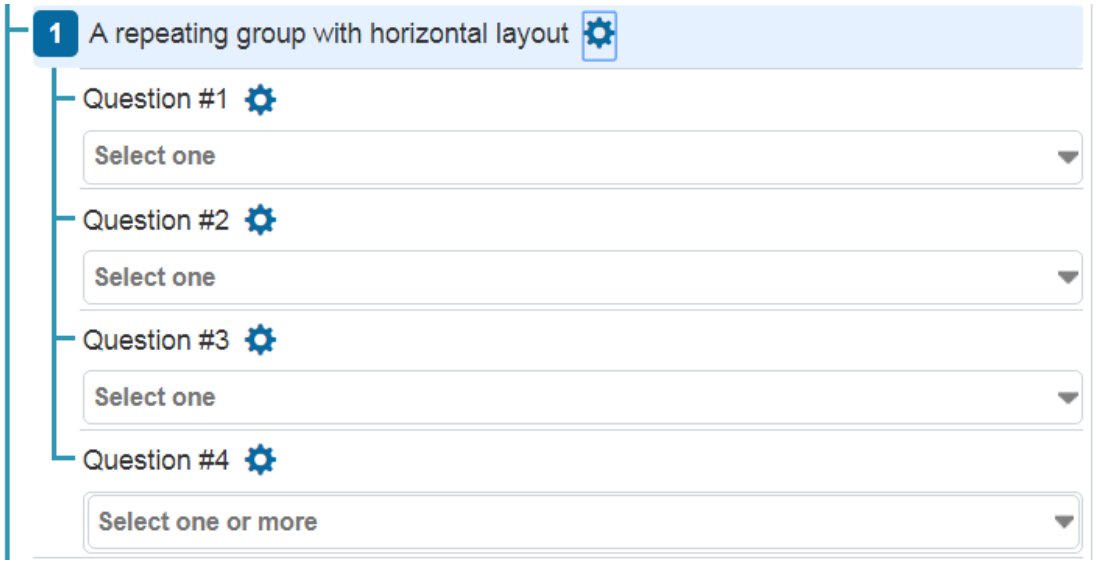
LHC-Forms Features: displayControl (1)




```
"displayControl": {
   "answerLayout":{
      "type":"COMBO_BOX"}}
"displayControl": {
   "answerLayout":{
      "type":"RADIO_CHECKBOX",
      "columns":2}}
"displayControl": {
   "answerLayout":{
      "type":"RADIO_CHECKBOX",
      "columns":4}}
"displayControl": {
   "answerLayout":{
      "type":"RADIO_CHECKBOX",
      "columns":2}}
```

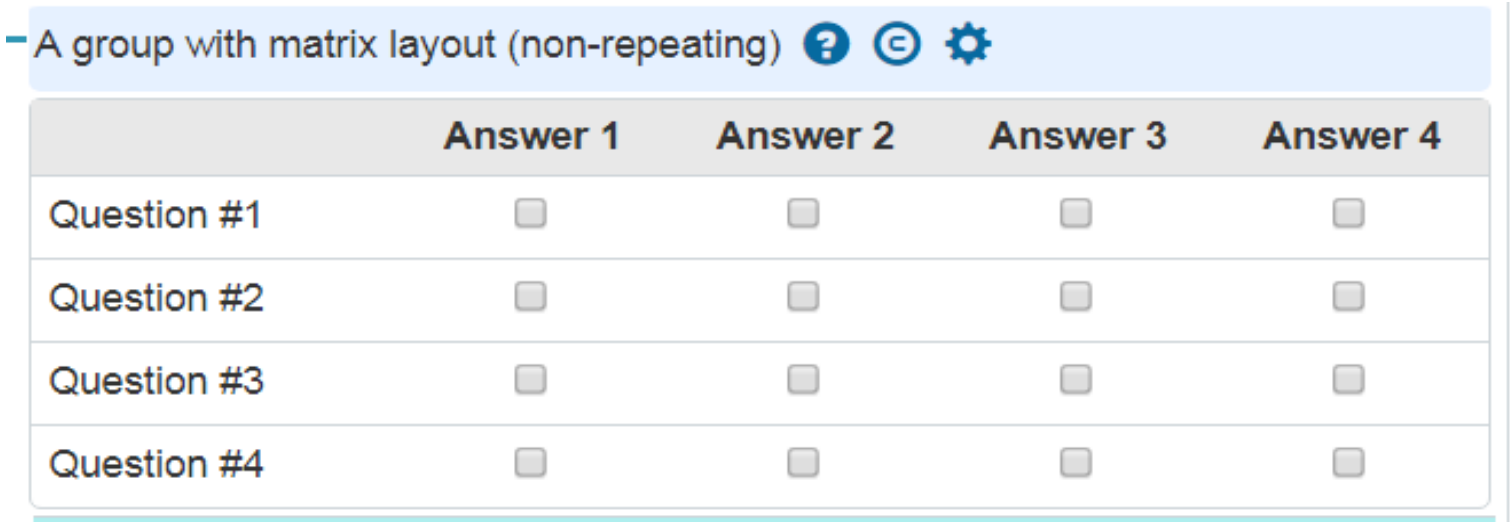
Presenter: Ye

LHC-Forms Features: displayControl (2)




```
"displayControl": {"questionLayout":"vertical"}
```

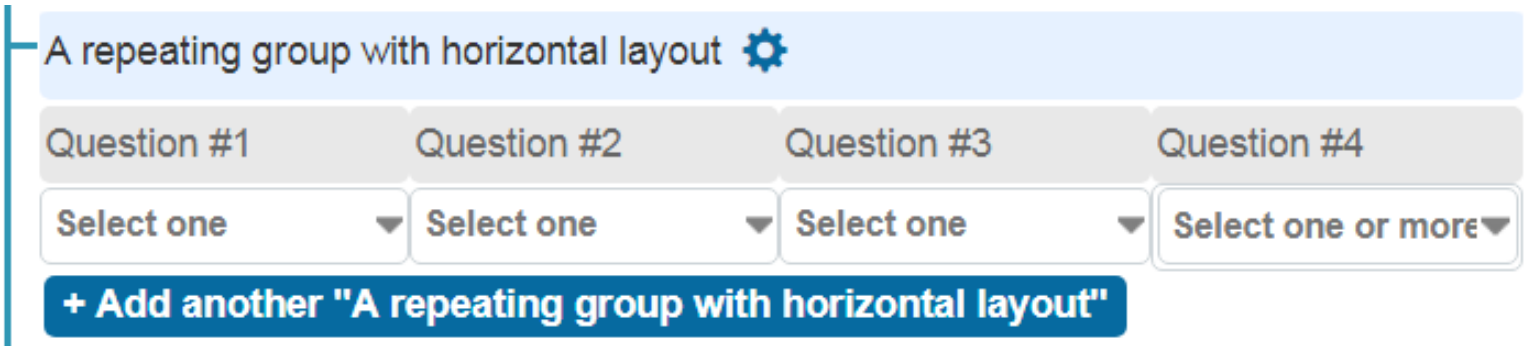
LHC-Forms Features: displayControl (3)




```
"displayControl": {"questionLayout":“matrix"}
```

Presenter: Ye

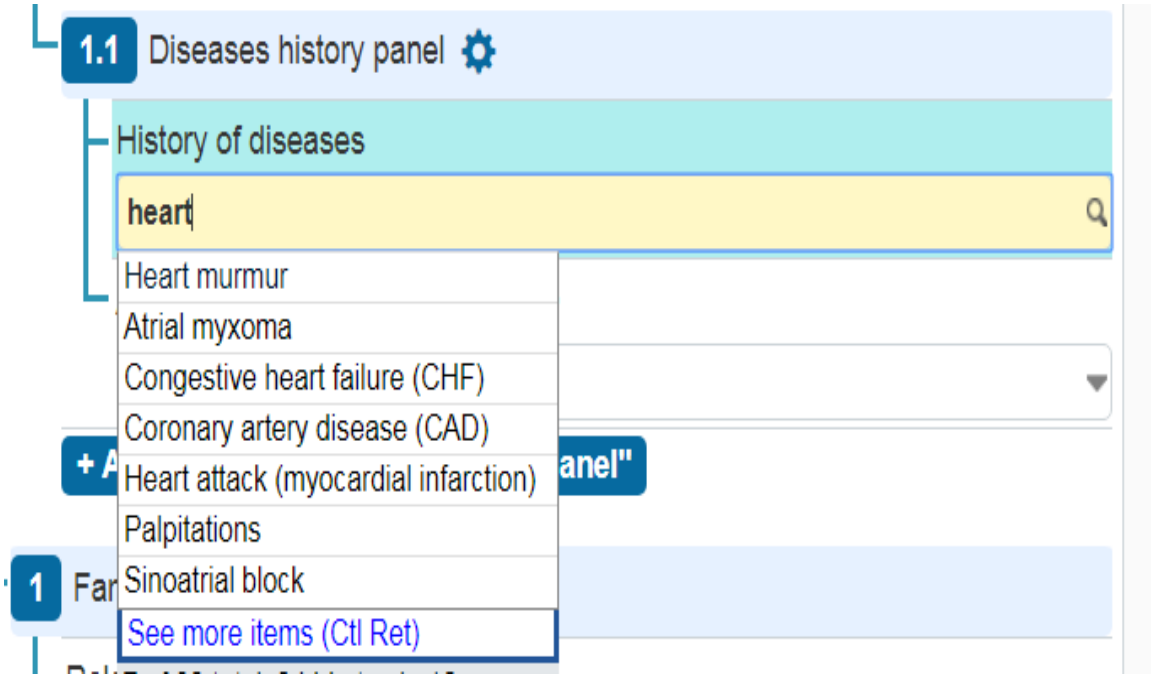
LHC-Forms Features: displayControl (4)




```
"displayControl": {"questionLayout":"horizontal"}
```

Presenter: Ye

LHC-Forms Features: Autocomplete—Search Field (1)



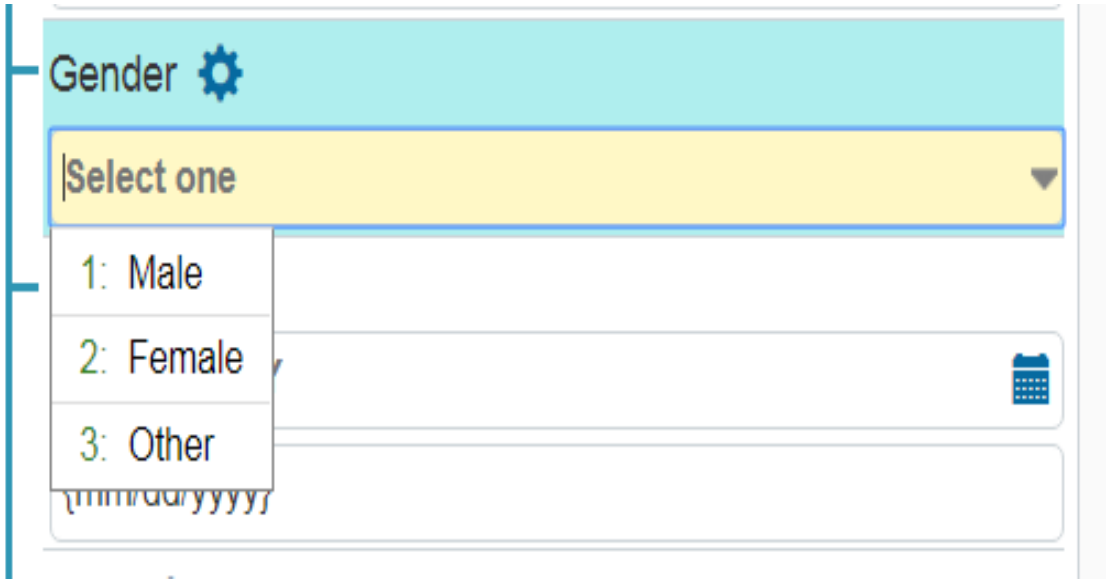
- Uses autocomplete-lhc widget
- Uses LHC CTSS search service

LHC-Forms Features: Autocomplete—Search Field (2)

```
{ "questionCode":"54116-9",
  "dataType":"CNE",
  "question":"History of diseases",
  "externallyDefined": “https://clin-table-search.lhc.nlm.nih.gov/api/conditions/v3/search”
}
```

Presenter: Ye

LHC-Forms Features: Autocomplete—Answer List (1)



- Uses autocomplete-lhc widget

Presenter: Ye

LHC-Forms Features: Autocomplete—Answer List (2)


```
{ "questionCode":"54123-5",
  "dataType":"CNE",
  "question":"Gender",
  "answers":[
     {"label":null,"code":"LA2-8","text":"Male","other":null},
     {"label":null,"code":"LA3-6","text":"Female","other":null},
     {"label":null,"code":"LA46-8","text":"Other","other":null}]
}
```


Presenter: Ye

LHC-Forms Features: template & templateOptions

showQuestionCode
showCodingInstruction
tabOnInputFieldsOnly
hideFormControls
hideUnits
showFormOptionPanel
showFormOptionPanelButton
allowMultipleEmptyRepeatingItems
allowHTMLInInstructions
useAnimation
defaultAnswer
displayControl
viewMode
defaultAnswerLayout
showFormHeader
formHeaderItem

LHC-Forms specs:
https://github.com/lhncbc/lforms/blob/master/form_definition.md

Presenter: Ye

LHC-Forms: Exporting data

- Available functions:
  - LForms.Util.**getUserData**(...) -- returns user data with options to include form definition data.
  - LForms.Util.**getFormData**(...) -- returns the form definition with user data. It has everything needed to redisplay the form in its filled-in state.
  - LForms.Util.**getFHIRData**(...) -- returns FHIR Questionnaire, QuestionnaireResponse or DiagnosticReport resources.
  - LForms.Util.**getHL7Data**(...) -- returns HL7 v2 OBR and OBX segment data.

- Documentation
  - http://lhncbc.github.io/lforms/index.html#retrieving-user-entered-data

Presenter: Ye

LHC-Forms: Installation for Development

- Install as a bower package
  - bower is a package manager for web application development (https://bower.io/)
  - “bower install lforms”
- Pre-built, ready-use packages on CTSS website (https://clin-table-search.lhc.nlm.nih.gov/lforms-versions)
  - Download the package
  - Include the package in webpages directly
  - Documentations: http://lhncbc.github.io/lforms/

Presenter: Ye

LHC-Forms: Resources

- LHC-Forms Project website
  https://lhncbc.nlm.nih.gov/project/lhc-forms
- LHC-Forms public demo application
  https://lhc-forms.lhc.nlm.nih.gov/
- Github pages
  https://github.com/lhncbc/lforms
  http://lhncbc.github.io/lforms/

Presenter: Ye

LHC-Forms: Exercise (1)

- LHC-Forms feature demo on Github
  - http://lhncbc.github.io/lforms/demos.html
- LHC-Forms demo website
  - https://lhc-forms.lhc.nlm.nih.gov/
- LHC-Forms on FHIR SMART app
  - https://apps.smarthealthit.org/app/24

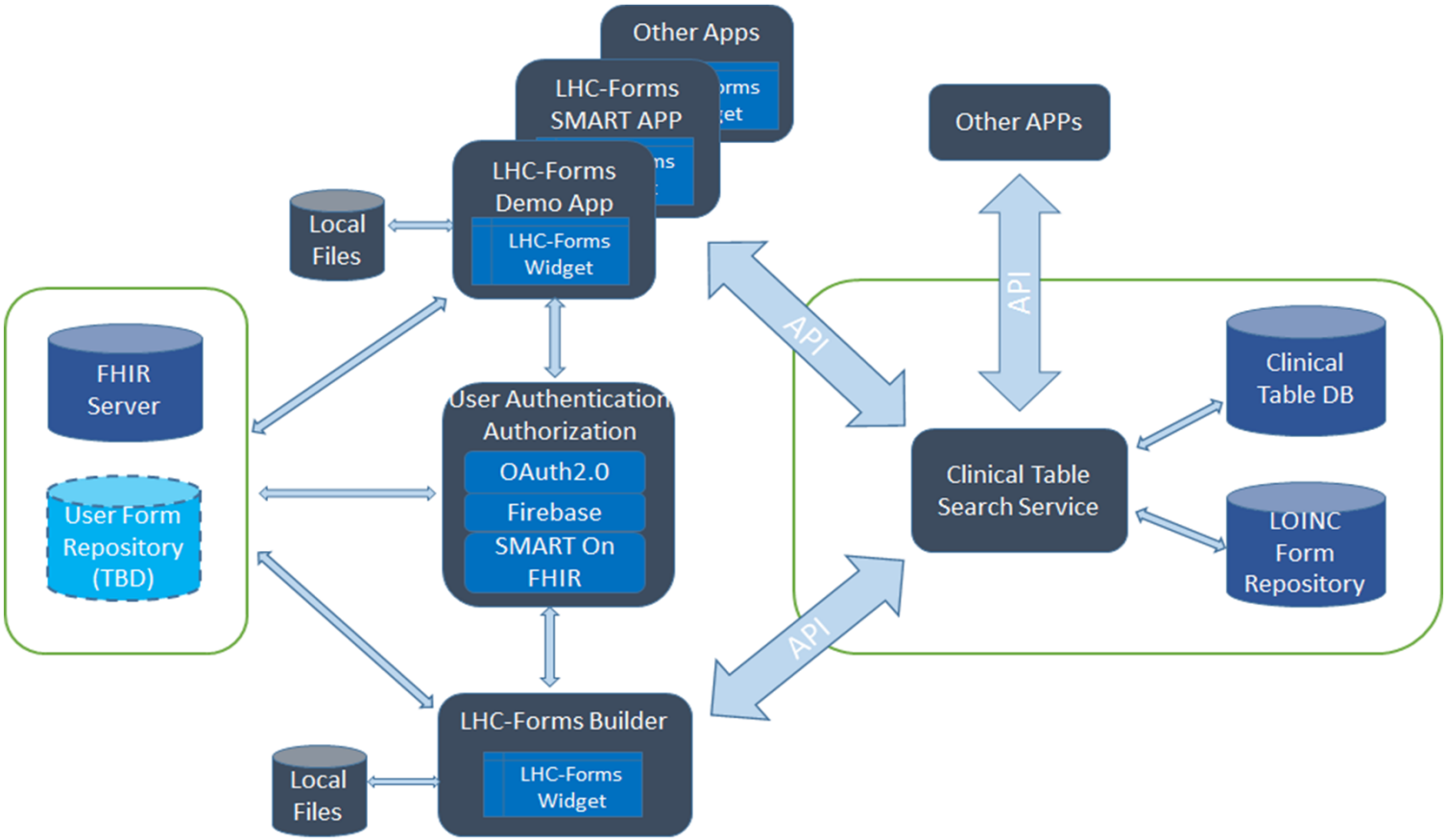
Presenter: Ye

LHC-Forms: Exercise (2)

- A demo application with Form builder and Demo app:
  - LHC-Forms form builder, to create/edit forms
  - LHC-Forms demo-app, to use those forms for user input data
  - Forms are stored as FHIR Questionnaire in a HAPI FHIR server
  - User data are stored as FHIR QuestionnaireResponse in the same HAPI FHIR server
  - User authentication/authorization is handled by Google firebase
- LHC-Forms Form Builder
  - https://lhc-formbuilder.lhc.nlm.nih.gov/
  - requires login: lforms/demo
- LHC-Forms FHIR Demo App
  - https://lhc-forms.lhc.nlm.nih.gov/demo-app

Presenter: Ye

LHC-Forms: System Design



LHC-Forms Builder

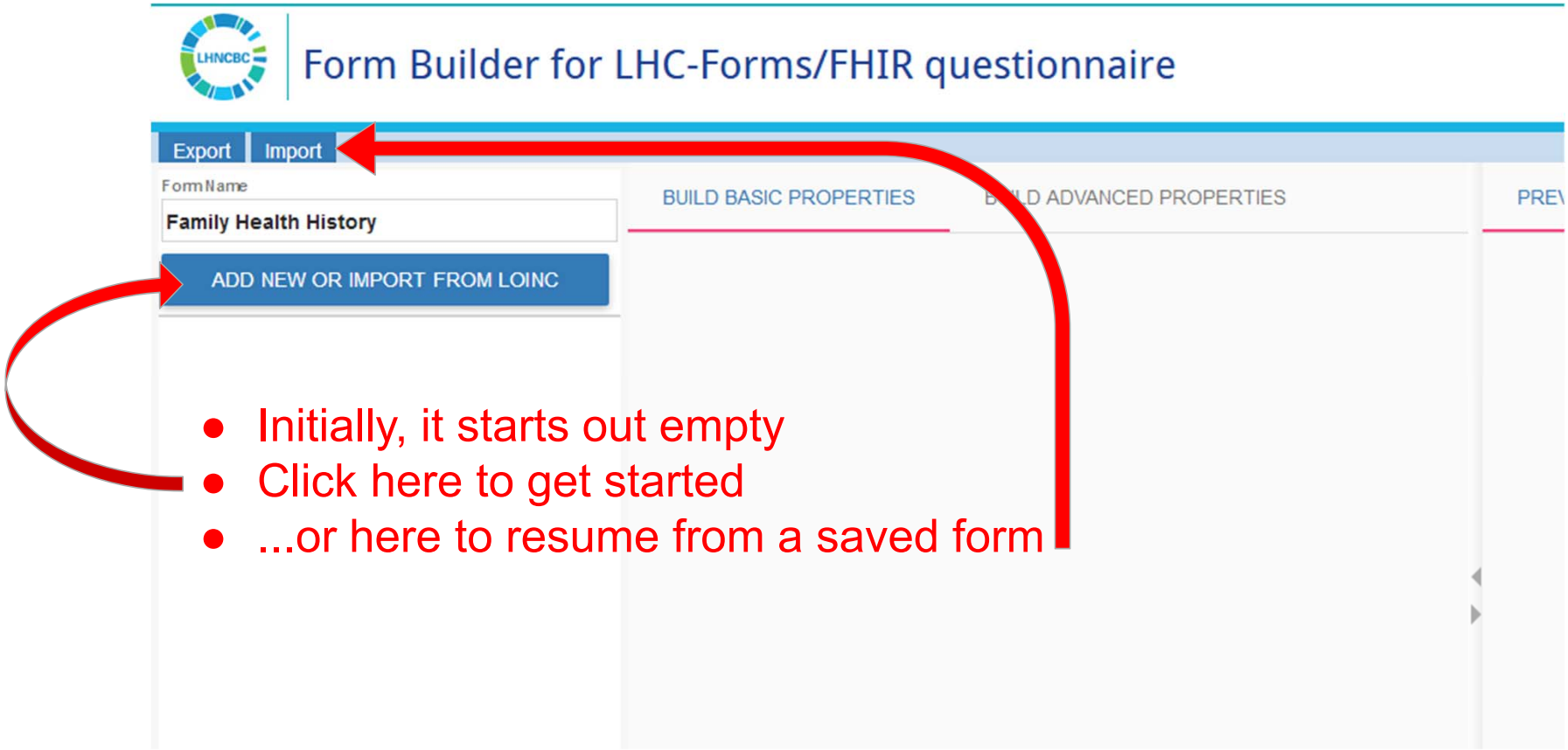
Presenter: Ajay

LHC-Forms Builder: Demo

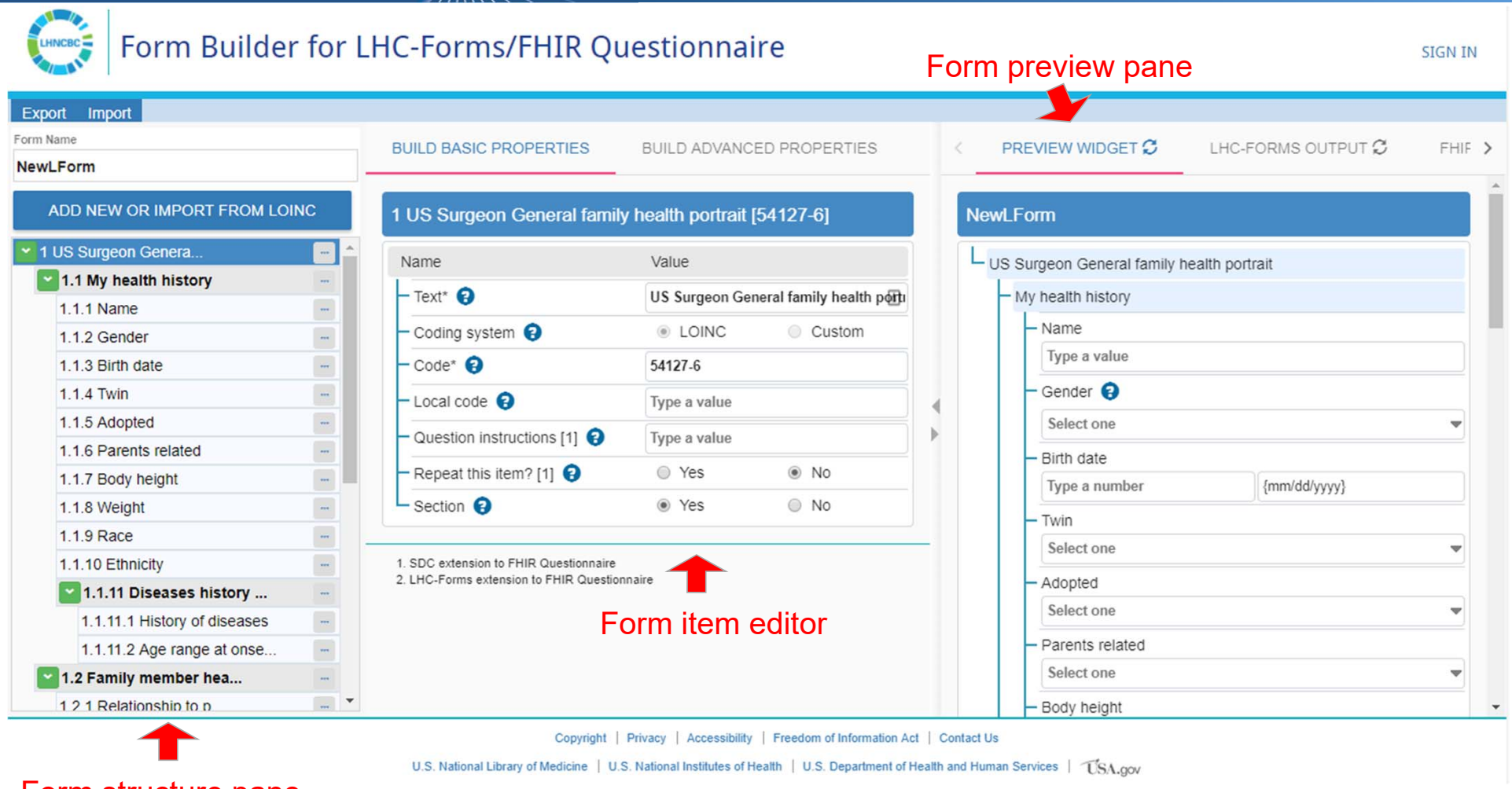
- Website: https://lforms-formbuilder.nlm.nih.gov/
- Builds forms for the LHC-Forms render, and can import or export
- Can import or export FHIR Questionnaires
- Currently behind a password, but we hope to remove that soon
- Can start with one or more existing LOINC panels, and add to or edit it.
- Can start from scratch, and import LOINC questions (with their answer lists) to make new panels

Presenter: Ajay

LHC-Forms Builder: Demo



Presenter: Ajay



Presenter: Ajay

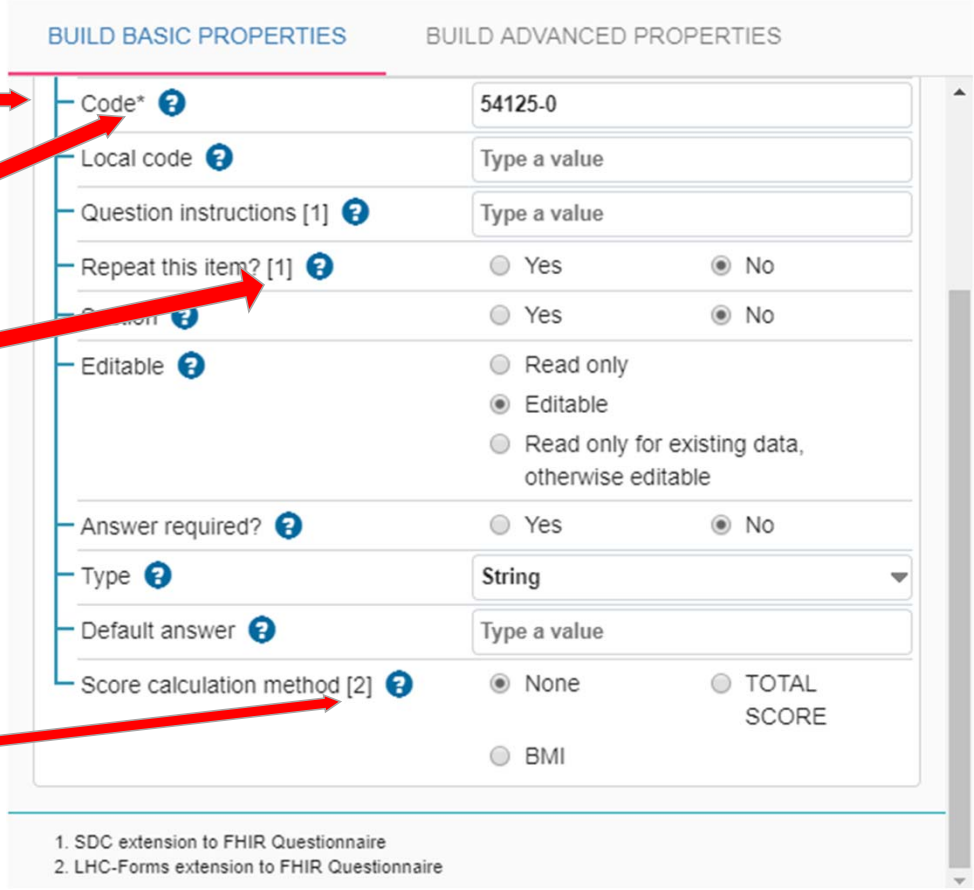
Form Item Editor

Fields without foot notes are defined in Standard FHIR Questionnaire

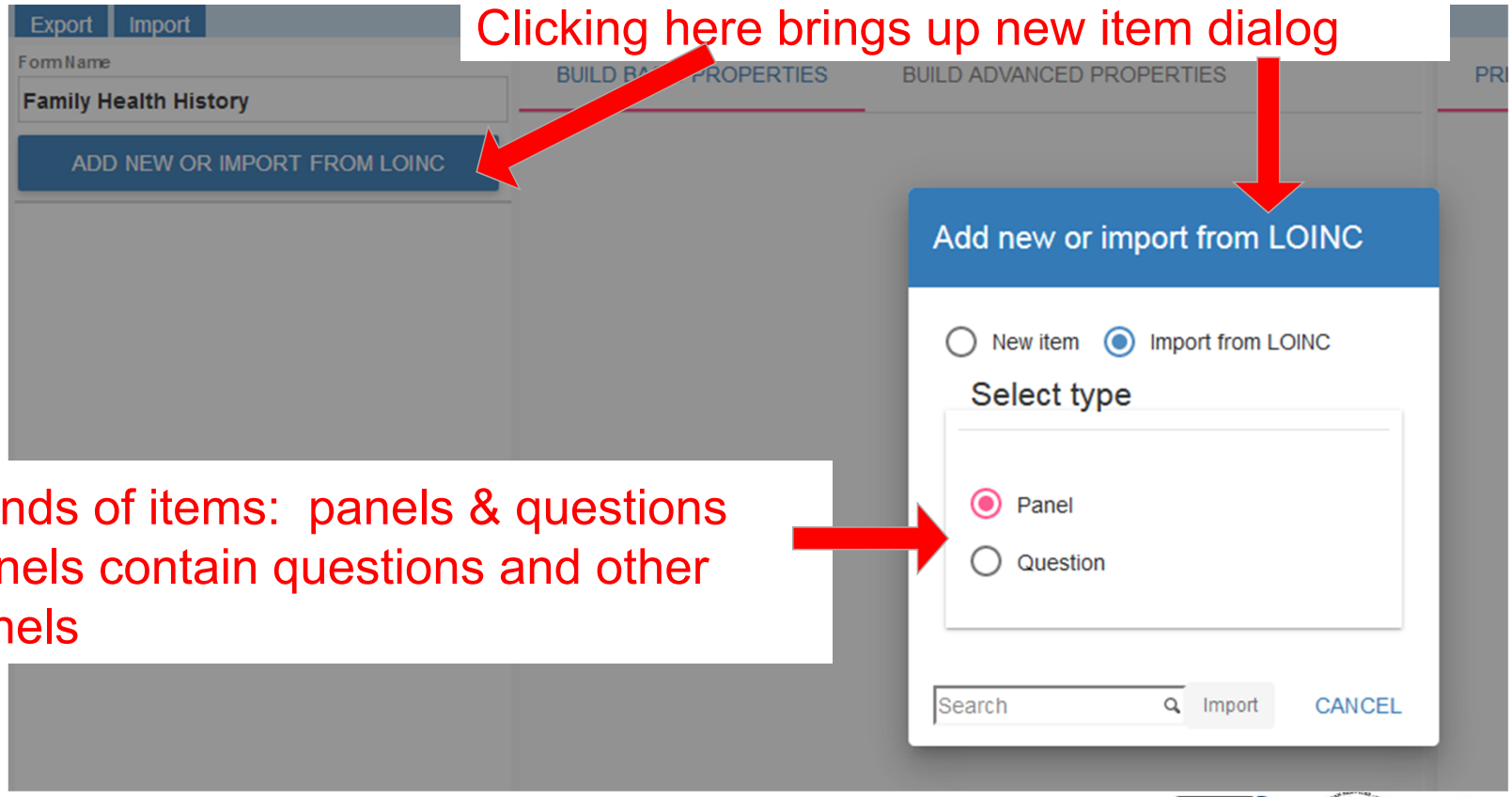
Fields with * are required

Fields with foot note [1] are SDC extensions to FHIR Questionnaire

Fields with foot note [2] are LHC-Forms extensions to FHIR Questionnaire



LHC-Forms Builder: Demo



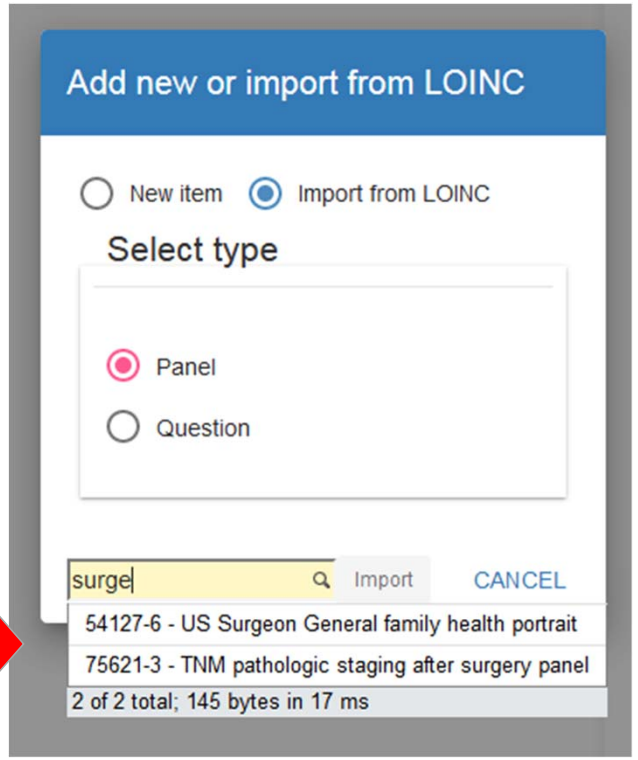
Two kinds of items: panels & questions

- Panels contain questions and other panels

Presenter: Ajay

LHC-Forms Builder: Demo

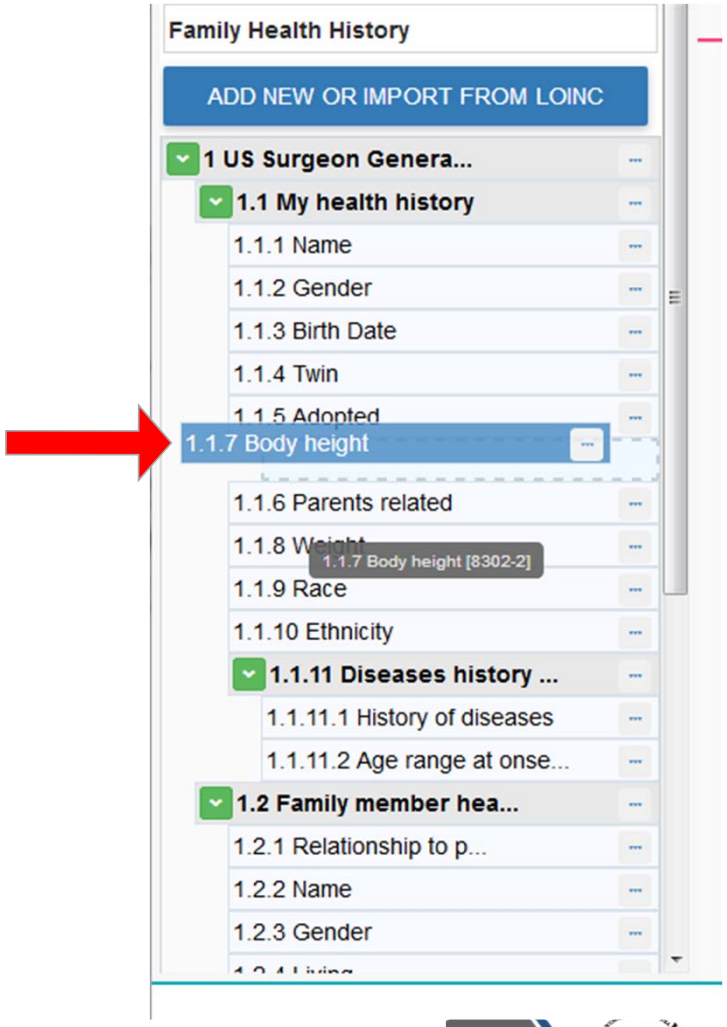
Autocompleting to bring in a LOINC panel



Presenter: Ajay

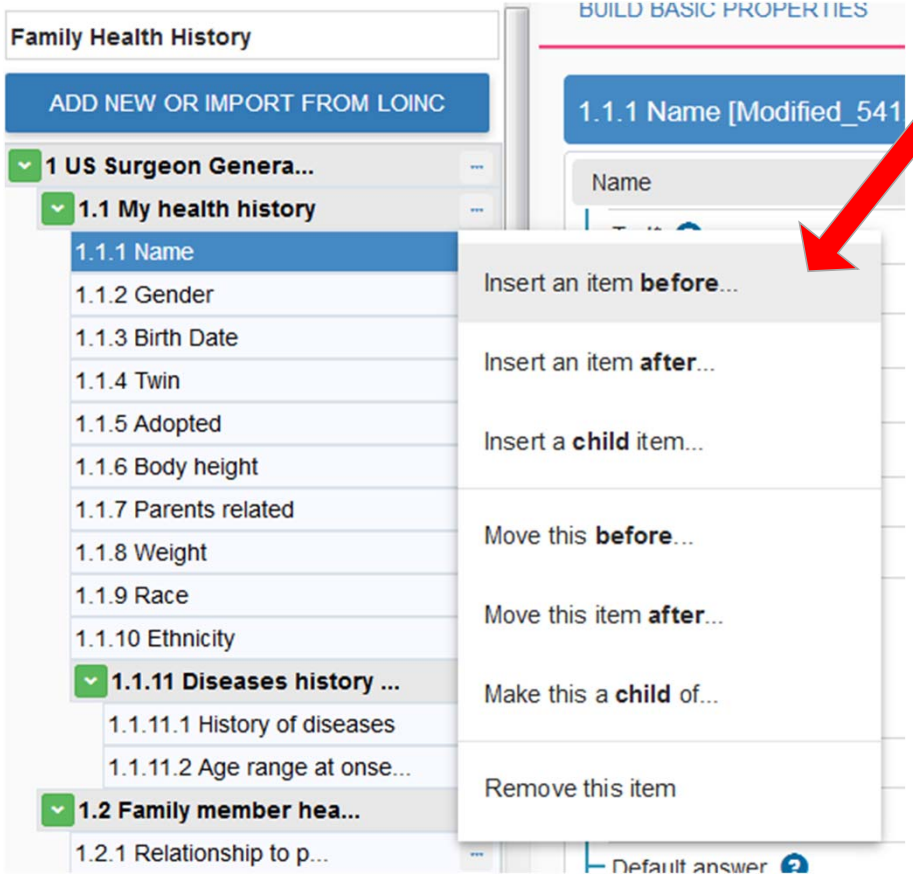
LHC-Forms Builder: Demo

The form structure pane allows for drag & drop rearranging of items, including making one item a child of another.



Presenter: Ajay

LHC-Forms Builder: Demo



Each item also has a context menu:

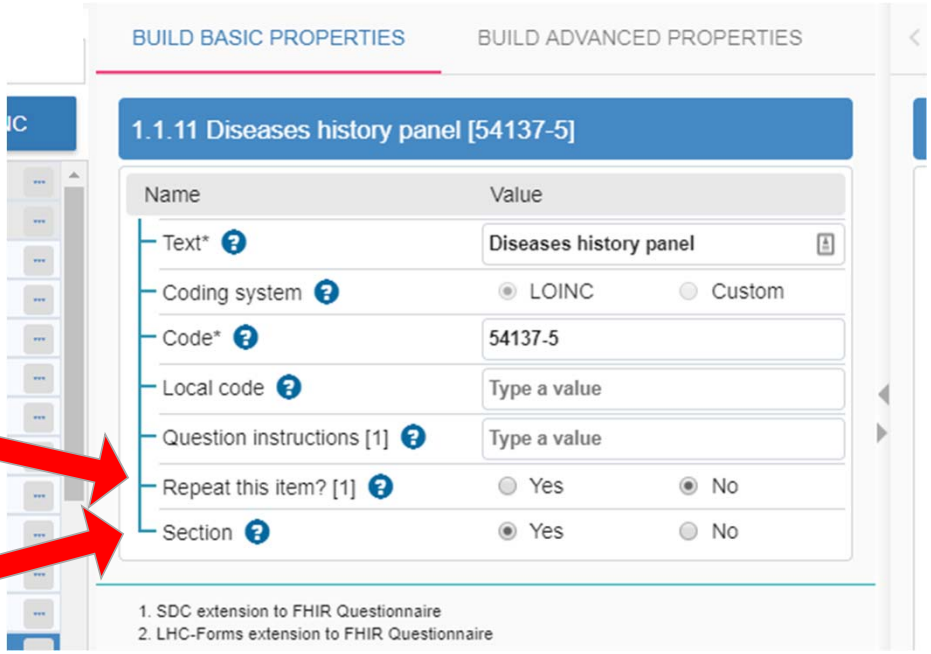
- Allows movement of items without drag and drop (for accessibility)
- Allows creation or import of new questions and panels next to that item
- Allows deletion of item
  - … currently without any pesky warnings to get in your way

Presenter: Ajay

LHC-Forms Builder: Demo – Item Properties

The item editing pane (the middle section) is divided into “basic” and “advanced” properties, to reduce the complexity for simple use case.

- Both sections and questions can repeat
- For sections, question options are hidden

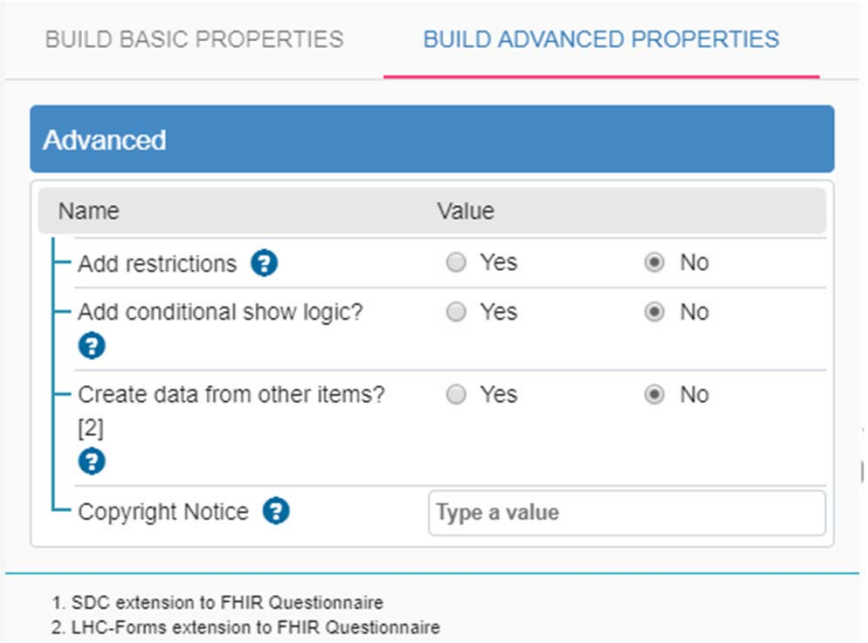


Presenter: Ajay

LHC-Forms Builder: Demo – Item Properties

Advanced properties for a section item:

- The complexities of the first three options are hidden until you select “Yes”.

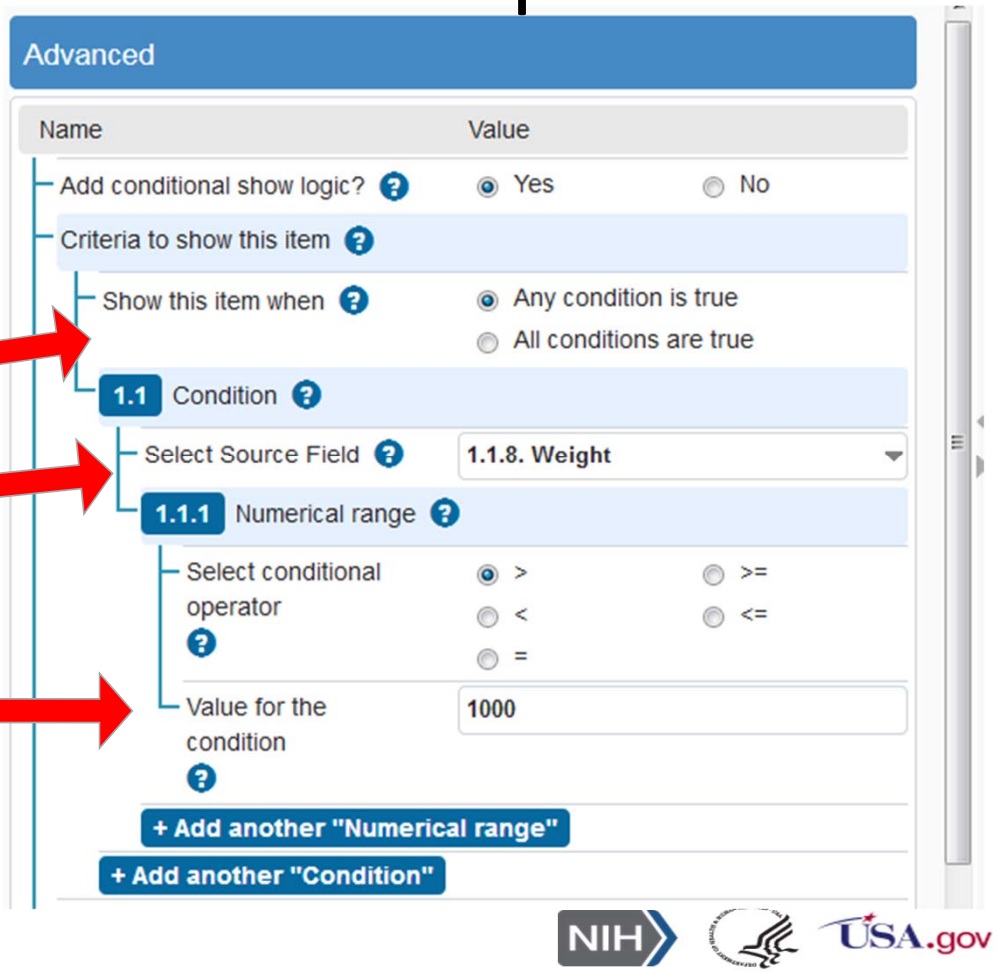


Presenter: Ajay

LHC-Forms Builder: Demo – Item Properties

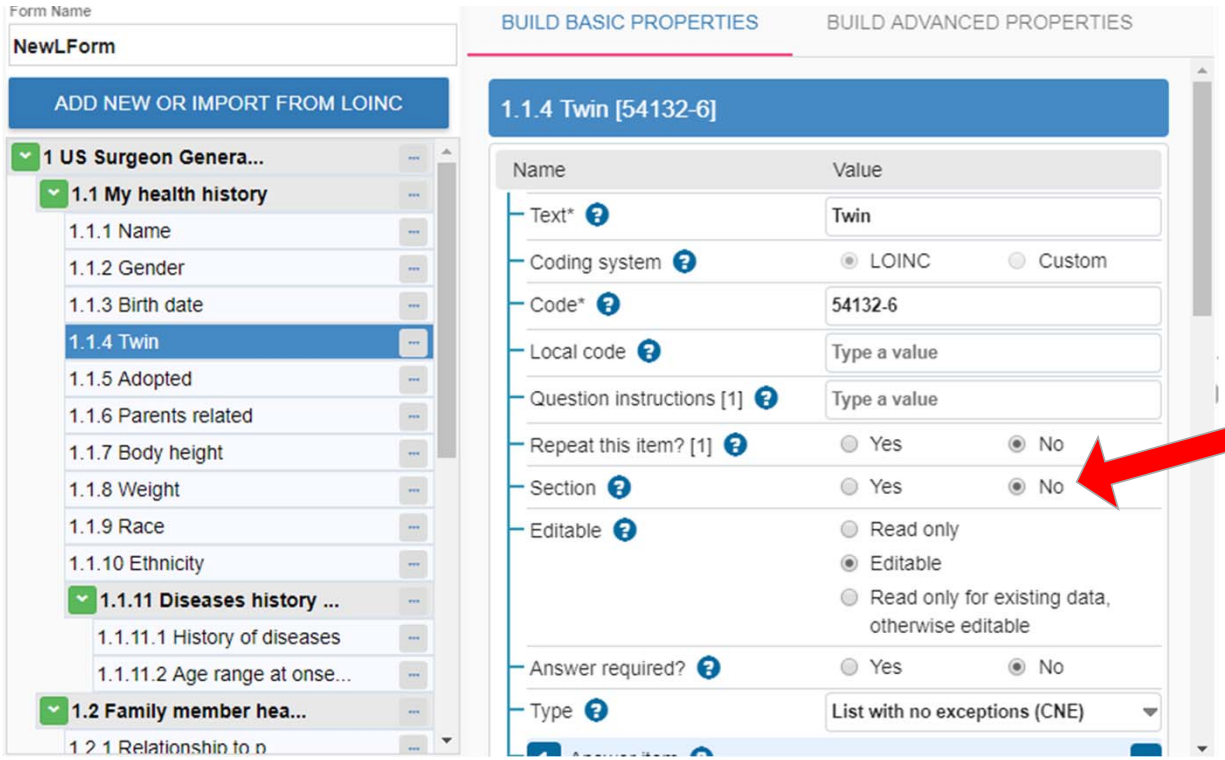
"Conditional show logic" (i.e. skip logic) allows set conditions under which the section or question will be shown.

- Multiple conditions can be used
- Select another question whose value triggers the showing of this item
- Specify the value or range that is required to show this item.



LHC-Forms Builder: Demo – Item properties

For items that are questions rather than sections, more options open up.



"Section" is set to "No"

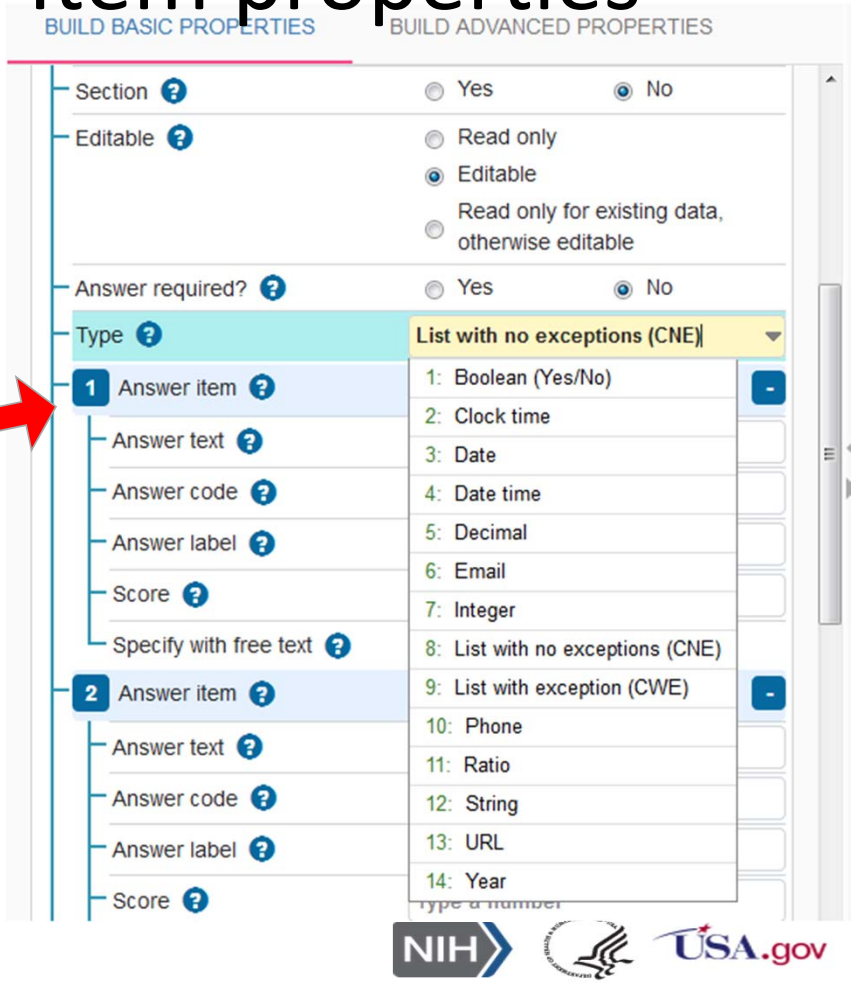
- Implies question

Presenter: Ajay

LHC-Forms Builder: Demo – Item properties

Options for questions allow control of:

- Whether an question's value is read-only
- Whether the answer is required
- Can select the data type for a question, which restricts the allowed input.
- Lists Can either allow free text or not (CWE vs CNE)
- List answers:
  - Can have "labels" (e.g. "a", "b") which appear with the answer text
  - Can have an "other" option which allows entry of a separate free text answer
  - Can have "scores" which are summed

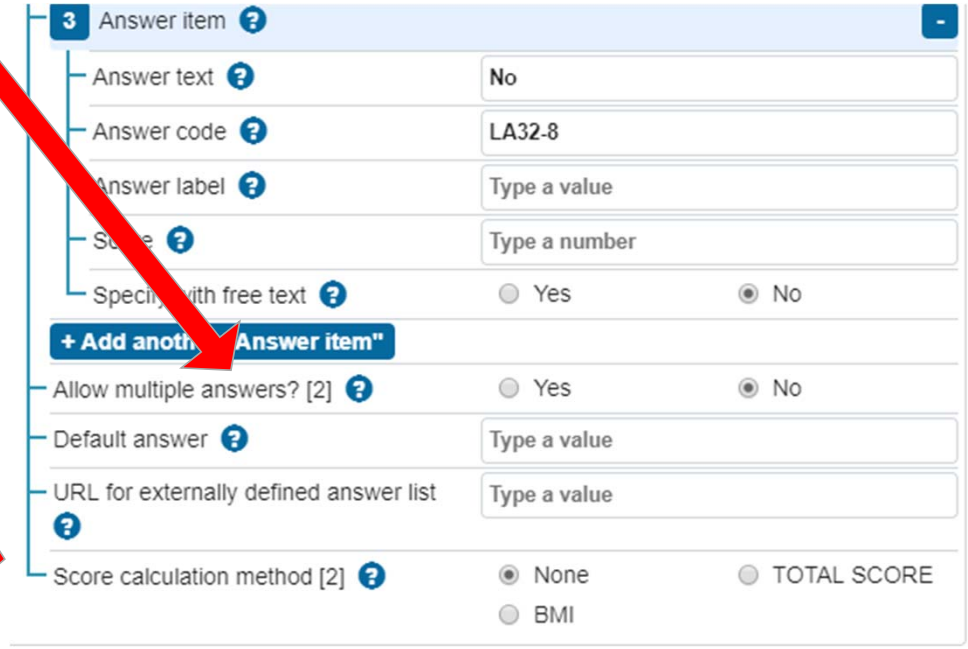


LHC-Forms Builder: Demo – Item properties

More on lists:

- Can allow multiple answers
- Can have a default answer
- Can be accessed via a URL which supports autocompletion (in autocomplete-lhc format).

The “Score calculation method” shown here is actually for numeric fields, but sums the scores associated with entered answers to list questions.

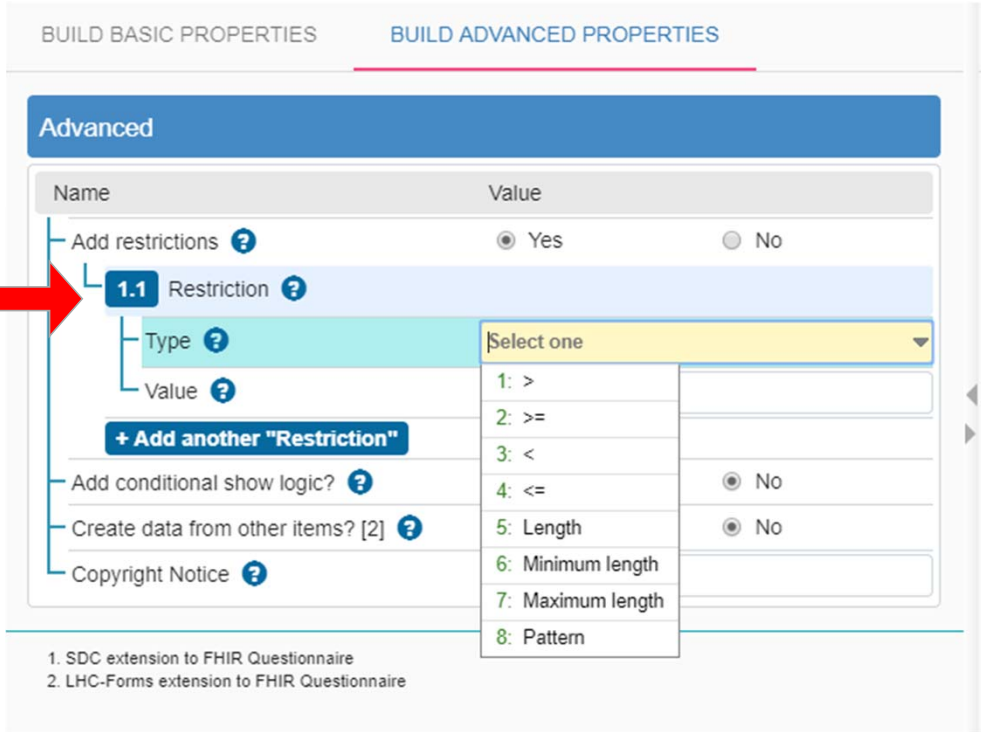


LHC-Forms Builder: Demo – Item Properties

Under “Advanced”, there are two new properties which apply to questions.

“Restrictions” allows you to restrict permitted input for the question.

- Minimum/maximum
- Number of characters
- Pattern matching
- Multiple restrictions can be added

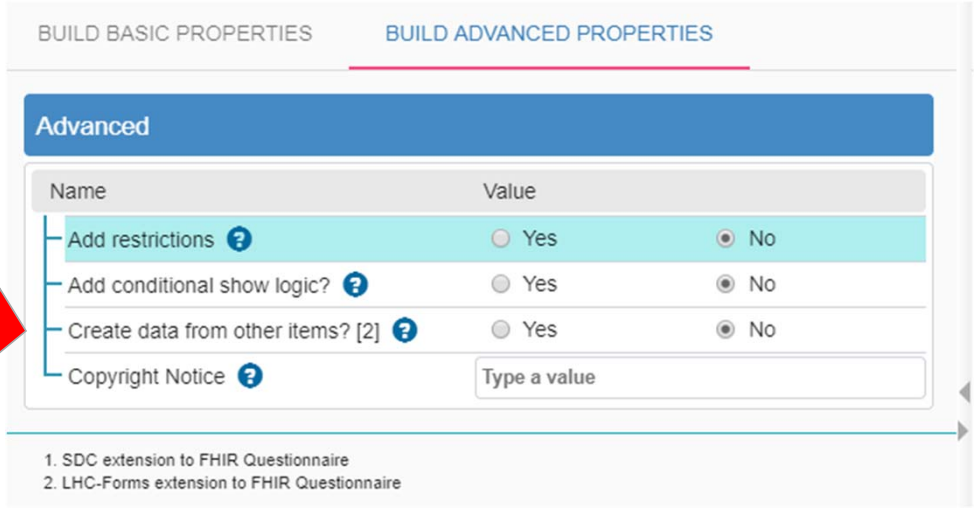


Presenter: Ajay

LHC-Forms Builder: Demo – Item Properties

Under “Advanced”, there are two new properties which apply to questions.

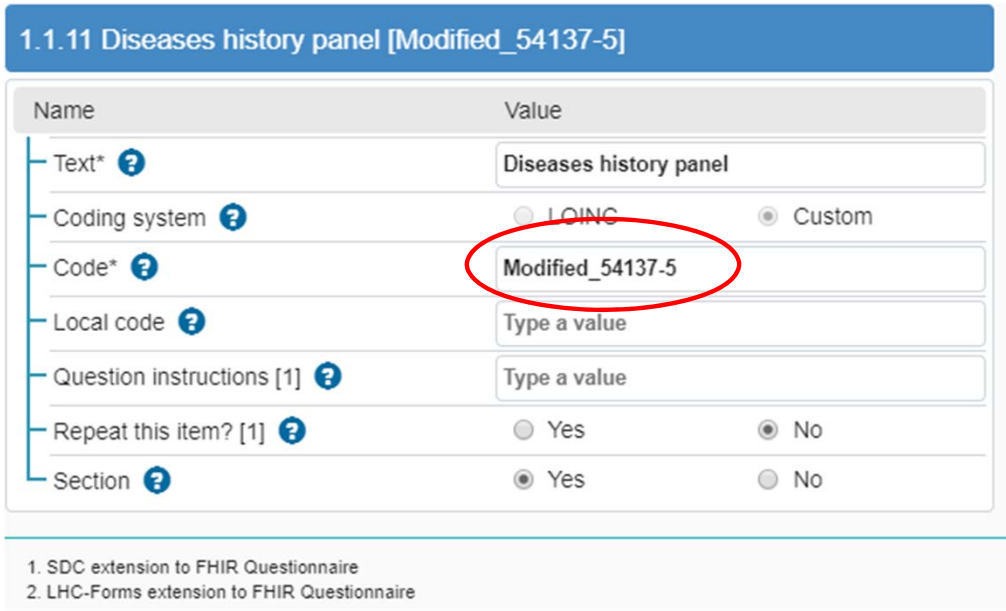
“Create data from other items” is the data control feature Ye discussed which allows one field to control another field’s content. The user interface here is not quite right yet, but we hope to fix that soon.



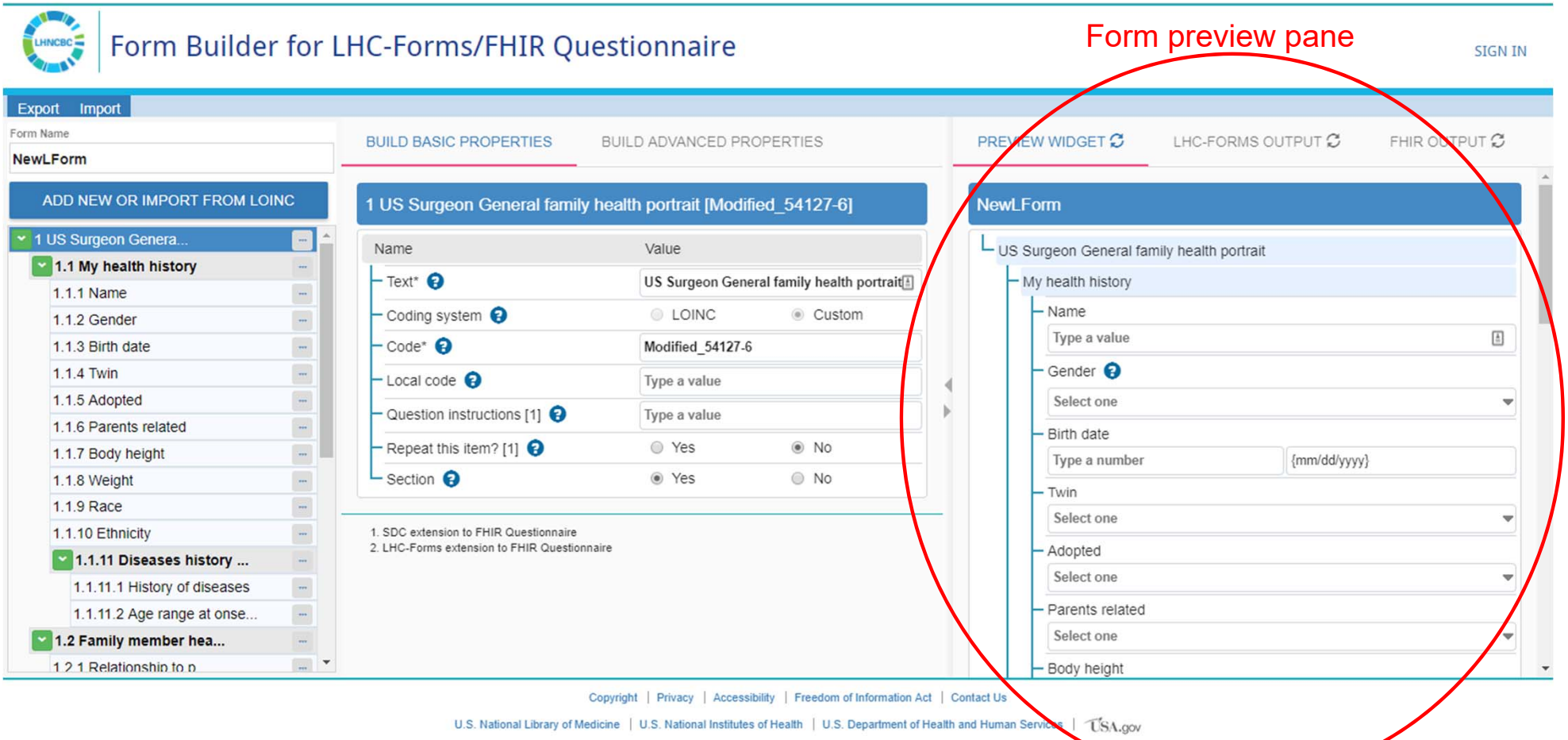
Presenter: Ajay

LHC-Forms Builder: Demo – Item properties

- All items (sections and panels) are required to have a code.
- Imported LOINC questions or panels start out having LOINC codes.
- When a LOINC item is changed, its code and its parent sections' codes are changed to make it clear that the item is no longer the standard coded item from LOINC.



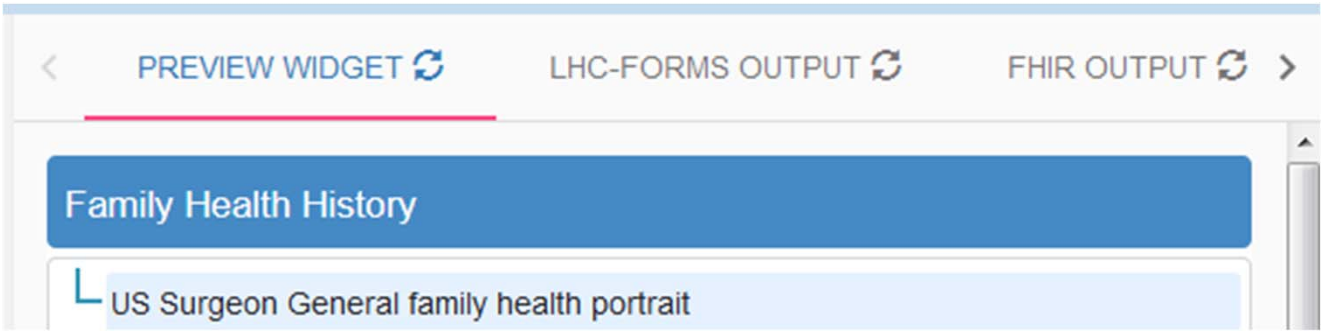
Presenter: Ajay



Presenter: Ajay

LHC-Forms Builder: Demo – Preview Pane

The preview pane has three tabs at top that control which type of preview is shown.

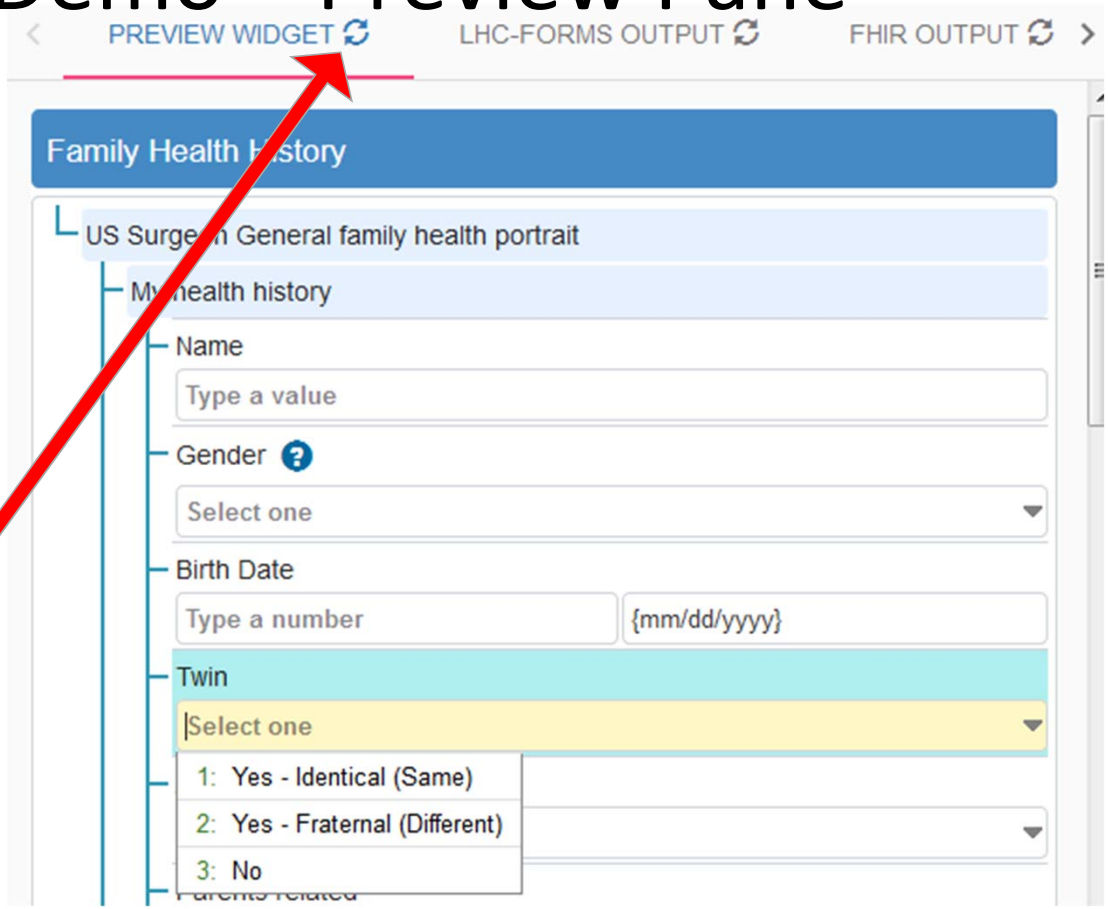


Presenter: Ajay

LHC-Forms Builder: Demo – Preview Pane

The "Preview Widget" tab shows a working preview of the form using the LHC-Forms rendering widget.

It does not update in response to every key stroke, so sometimes you might need to use the refresh button to see the latest changes you have made.



Presenter: Ajay

LHC-Forms Builder: Demo – Preview Pane

The "LHC-Forms Output" tab shows preview of the JSON LHC-Forms definition.

Presenter: Ajay

LHC-Forms Builder: Demo – Preview Pane

The "FHIR Output" tab shows preview of the JSON FHIR Questionnaire definition.

Presenter: Ajay

LHC-Forms Builder: Demo – Saving & Loading

At the top of the form are buttons that open menus for savings (Export) and loading (Import).

- Each menu gives two options:
  - local file
  - FHIR server
- Save to a local file often
  - Currently reloading the page does not warn you that you'll lose any changes!

Presenter: Ajay

LHC-Forms Builder: Exercise

- Use the form builder to:
  1. Build a form
  2. Save the definition to a FHIR server as a Questionnaire
- Use the LHC-Forms demo app to:
  1. Load the saved form
  2. Fill it out
  3. Save the response to a FHIR server as a QuestionnaireResponse

Presenter: Ajay

Unified Code for Units of Measure (UCUM)

Presenter: Paul

UCUM-LHC: Unit Validation and Conversion

- For “Unified Code for Units of Measure” (UCUM)
- Website: https://ucum.nlm.nih.gov/ucum-lhc/
- Library can be downloaded from GitHub or installed with “bower”
- Library supports:
  - Validation of unit expressions
  - Conversion of values between different unit expressions
- Unit codes in UCUM are not always what one would expect, but there are synonyms
- Some special syntax: “.” = multiplication, * = exponentiation

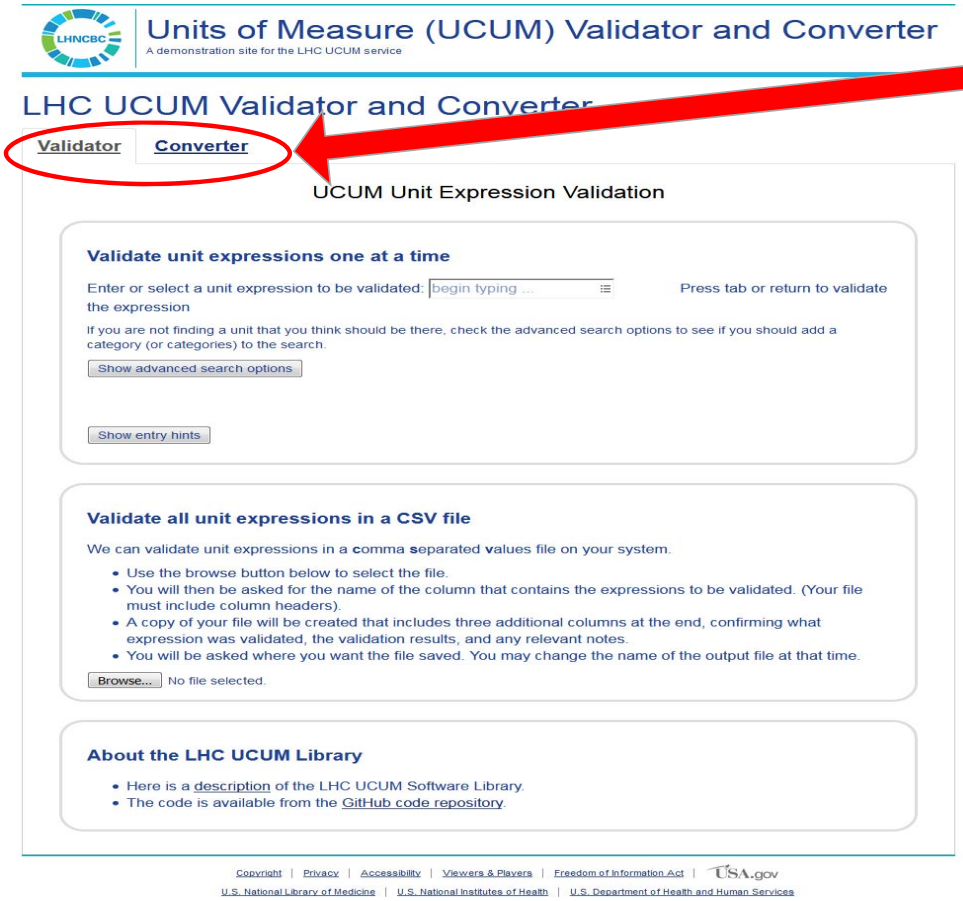
Presenter: Paul

UCUM-LHC: Demo

- Demo of library at https://lhncbc.github.io/ucum-lhc/demo.html
- Validation of individual expressions
- Batch validation of expressions in a spreadsheet file
- Conversion from one expression to another

Presenter: Paul

UCUM-LHC: Demo



- Demo has separate tabs for validating and converting
- Top part of page has section for validating individual UCUM unit expressions.
- Lower part of page has an option for validating a spreadsheet containing UCUM expressions in a column.
  - Results are appended in a new column and the revised spreadsheet is returned.

Presenter: Paul

UCUM-LHC: Demo

Validate unit expressions one at a time

Enter or select a unit expression to be validated: kg.m

Press tab or return to validate the expression

If you are not finding a unit that you think should be the to

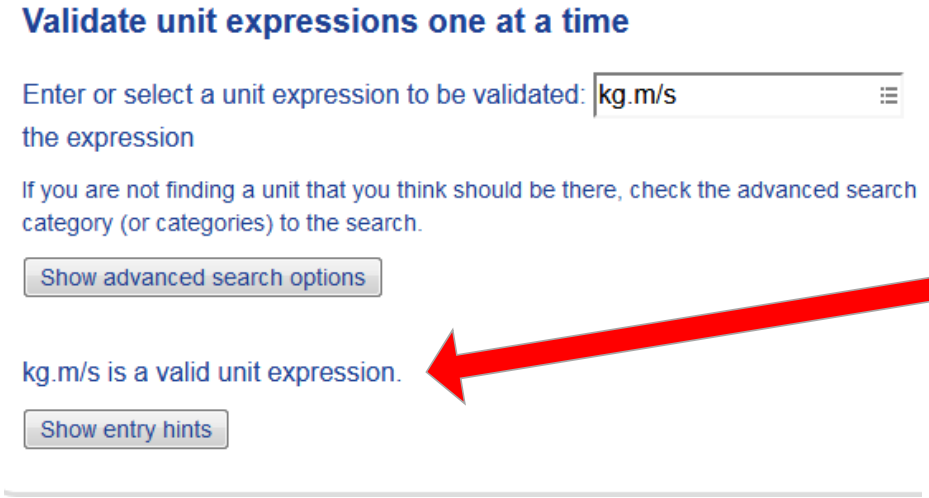
Show advanced search options

m is a valid unit expression.

cs_code	name	guidance
m	meter	unit of length = 1.09361 yards
m2	square meter	unit often used to represent body surface area
m[H2O]	meter of water column	
m[Hg]	meter of mercury column	
mA	milliampere	unit of electric current

- Unit expression components (e.g. “kg” and “m”) autocomplete as you type, to aid in learning UCUM.

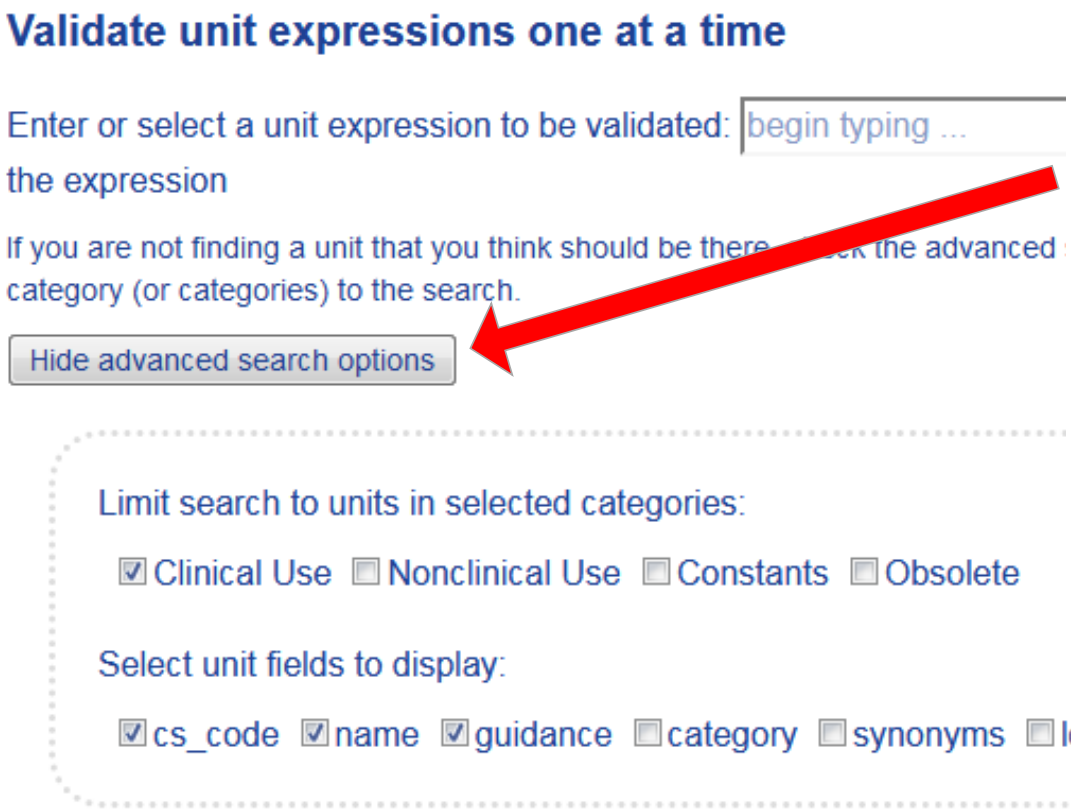
UCUM-LHC: Demo



After expression is entered, hitting tab or return will cause the expression to be checked, and the result will appear below.

Presenter: Paul

UCUM-LHC: Demo

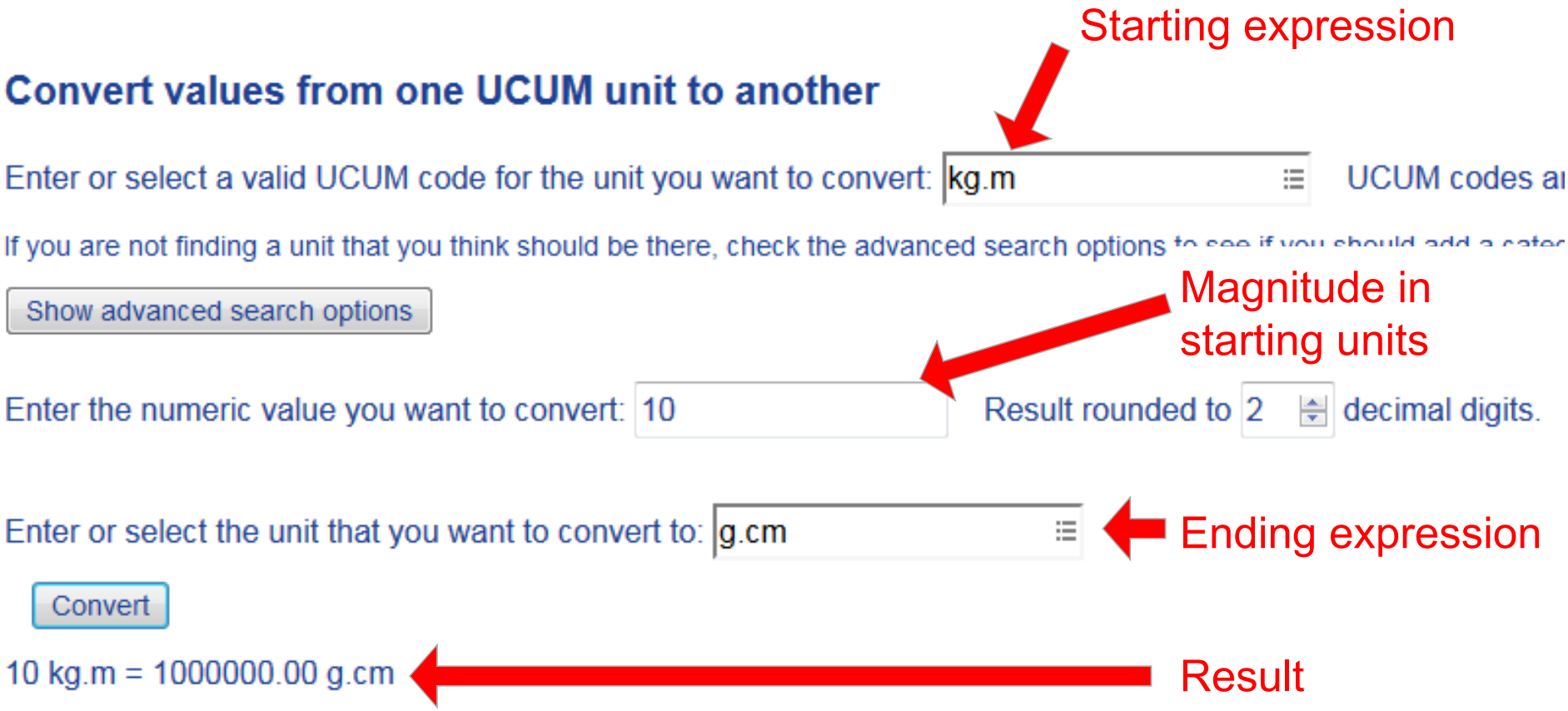


"Advanced tab provides options for:

- Narrowing type of units that appear in the autocompletion list
- Selecting fields that display in the autocompletion table

Presenter: Paul

UCUM-LHC: Converter Demo



Presenter: Paul

UCUM-LHC: Code

- Code can be installed as a bower package: bower install ucum-lhc
- Can also link to the library on our demo site, or download it:
  - https://lhncbc.github.io/ucum-lhc/dist/ucum-lhc.min.js
- Example usage:

```
<script src="https://lhncbc.github.io/ucum-lhc/dist/ucum-lhc.min.js"></script>
<script>
  var utils = ucumPkg.UcumLhcUtils.getInstance();
  var resultObj = utils.validateUnitString('m2/g4');
</script>
```

- Experiment here: https://jsfiddle.net/lforms/y6kde2xd/

NIH USA.gov

UCUM Web Service

- For validating and converting without using our JavaScript library
- Programs can send URL requests with units to validate or convert and get a XML, JSON, or plain text responses back.
- https://ucum.nlm.nih.gov/ucum-service.html
- These APIs were donated by Jozef Aerts of XML4Pharma
- Examples:
  - https://ucum.nlm.nih.gov/ucum-service/v1/ucumtransform/1/from/[in_i]/to/cm
  - https://ucum.nlm.nih.gov/ucum-service/v1/isValidUCUM/mm

Presenter: Paul