Cooking with CQL Q&As

Session 58 - Thursday, October 28, 2021

Using Fast Healthcare Interoperability Resources (FHIR)

# **Q:** In Fast Healthcare Interoperability Resources® 4.0.1, when writing a calculation that requires complex arithmetic, would it be best to use the fluent function to make the logic more readable to the measure implementer?

**A:** The intent of the fluent function is to help with complicated logic to provide a more human readable expression. You can use all arithmetic operators within this function to chain operations together to obtain closure and achieve a usable fluent syntax. However, it is important to look at each use case to determine whether it is best to use a fluent function or to use an equation with the required mathematics to achieve the most succinct and clear human readable expression. For additional details on the use of fluent functions, please refer to the Clinical Quality Language Developer’s Guide at <https://cql.hl7.org/03-developersguide.html#fluent-functions>.

An example of a fluent function

|  |  |
| --- | --- |
| define fluent function plus(x Integer, y Integer): | |
|  | Plus(x, y) | |
|  |  | |
|  | define fluent function minus(x Integer, y Integer): | |
|  | x - y | |
|  |  | |
|  | define Testplus: | |
|  | 2.plus(2).minus(12) | |

**Q:** In the Fast Healthcare Interoperability Resources®-based Global Common library, is the “Normalize Interval” an example of a fluent function in Clinical Quality Language using syntactic sugar\*?

\* Syntactic sugar is defined as a syntax in a programming language that also improves the readability of code.

|  |  |
| --- | --- |
|  | |
| define "Flexible Sigmoidoscopy Performed": | |
|  | [Procedure: Concepts"Flexible Sigmoidoscopy"] FlexibleSigmoidoscopy | |
|  | where FlexibleSigmoidoscopy.status = 'completed' | |
|  | and Global."Normalize Interval"(FlexibleSigmoidoscopy.performed) ends 5 years or less on or before end of "Measurement Period" | |
|  | |  |

**A:** Correct. It is exactly like an extension method in C sharp. The ELM translator expands the “toInterval” to a prefix implementation and used purely for readability.

**Q:** In Fast Healthcare Interoperability Resources® 4.0.1, the creation of US Core Elements library v3.1.1 and Quality Improvement (QI)-Core Elements library v4.1.0 were to define functions to expose extensions as fluent functions in Clinical Quality Language (CQL). How would the use of these fluent functions appear in the Measuring Authoring Tool (MAT) human readable format for a user?

**A:** The human readable, MAT, and CQL code would be the same and appear as

|  |  |
| --- | --- |
| define "Encounter With First ICU Stay With Principal Procedure of SCIP VTE Selected Surgery (2)": | |
|  | from | |
|  | "Encounter With ICU Location" QualifyingEncounterICU, | |
|  | "SCIP VTE Selected Surgery" SelectedProcedure | |
|  | let EncounterProcedure: singleton from ( | |
|  | (QualifyingEncounterICU.procedure()) P | |
|  | where P.rank = 1 | |
|  | and P.procedure.getId() = SelectedProcedure.id | |
|  | ) | |
|  | where date from end of SelectedProcedure.performed.toInterval() | |
|  | during VTE.CalendarDayOfOrDayAfter(VTE.StartOfFirstICU(QualifyingEncounterICU)) | |
|  | return QualifyingEncounterICU | |
|  | |  |

**Q:** Are fluent functions only feasible in Fast Healthcare Interoperability Resources (FHIR®) and not in a Quality Data Model (QDM) specification?

**A:** FHIR fluent functions are independent of the model. You can apply normalized intervals in the QDM global common functions library in QDM as fluent functions. The QDM global common functions library does not need extension functions because QDM doesn’t have extensions, but they can still be used to provide a post-fix notation

**Q:** In Fast Healthcare Interoperability Resources® 4.0.1, is there a difference in the use of fluent functions if you define the fluent function in a shared library instead of the code?

**A:** The expression does not change. You can define a fluent function in the code or in a shared library and it invokes the same way as if you defined it in the code.

**Q:** When using fluent functions in Fast Healthcare Interoperability Resources® 4.0.1, do you need to use the prefix if you define the intervals in different files?

**A:** No, you can define the intervals in different files if the libraries include the defined intervals.

**Q:** In Fast Healthcare Interoperability Resources® 4.0.1, when using fluent functions, where do the data requirements display for the end user?

**A:** The fluent functions will display in the Function section of the human readable output as a function dependency.

**Q:** When using fluent functions in Fast Healthcare Interoperability Resources (FHIR®) 4.0.1, how does FHIR resolve the conflict if two libraries have fluent functions with the same name?

**A:** When two libraries contain fluent functions with the same name, resolution first occurs in the local library then you apply the fluent functions in the order as declared in the include statements.

|  |  |
| --- | --- |
| include FHIRHelpers version '4.0.1' | |
|  |  | |
|  | include Concepts called Concepts | |
|  | include FHIRCommon called FC | |
|  |  | |
|  | include MATGlobalCommonFunctionsFHIR4 called Global | |
|  | include VTEFHIR4 called VTE | |

**Q:** Do fluent functions support overloading?

**A:** Yes, fluent functions support overloading. Overloading allows a single function to have multiple uses based on the input and output parameters included with the function. Overloading reduces the complexity of the code.