

\$validate-code behaviour

A tracker item has been created to clarify the behaviour of \$validate-code

https://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=17218

This came from a discussion of the ambiguity regarding the validation that should be done for the display term. For SNOMED CT for example, should this return true (i.e. valid) if the term provided to \$validate-code matches

- the term specified in the ValueSet, if provided?
- the preferred term? If so, in which language reference set?
- any of the acceptable synonyms? Again, if so in which language reference set?
- any of the active synonyms?
- any active description? (i.e. any active synonym or FSN)
- any description, current or historic?

This comes partly down to the use case for performing this validation. Validating a code against a ValueSet makes sense to determine if someone has sent a valid code for the binding.

Validating the display is perhaps a bit more subtle. The most obvious use case is to validate if it is likely there is a mistake or not. For example it may be useful to know if a coded element's display term matches the code in the terminology as confirmation.

- code = 387517004, term = Paracetamol - matches preferred term
- code = 387517004, term = Acetaminophen - not the preferred term, but still a valid term
- code = 387517004, term = Amoxicillin - a mistake has probably been made!

In the above case it is useful to know from \$validate-code whether the term is a valid description for the code - at least all active descriptions. Receiving the FSN as the display is perhaps odd, and probably wrong (wouldn't imagine many clinical systems where it is valid for it to be the display term), but not unsafe (it is *supposed* to be unambiguous) and still useful for validation/confirmation.

In this scenario, requiring the specified term to be the preferred term is problematic - which language reference set is the first problem, but also if the version of the code system being used isn't specified it may not be possible to determine the right value even if the language reference set is known. Similarly if the version of the terminology isn't known, then matching inactive descriptions may be useful when dealing with old versions...but then descriptions are inactivated for a reason.

Perhaps a ternary valued response of "valid", "matches inactive description", and "invalid" would solve this issue...but adds complexity for something that is perhaps an edge case.

Questions

- what are the use cases for \$validate-code involving validating the term?
- based on that, what should be considered "valid" for an input code and term generally for a FHIR CodeSystem and ValueSet combination?
- then specifically, for SNOMED CT what should the interpretation be and is any supporting clarification necessary in the Using SNOMED CT with FHIR page?

Group Discussion

26 June 2018 - Basically we really need to know what SNOMED Edition is being used, then we can come up with sensible behaviour.

In general, valid codes should return "valid = true" and term issues should be identified as a warning in the message. Any matching term should result in positive validation.

RH: If we had an option to specify a "strict" flag, then we could validate the term in a more exacting way.

DM: Developers want finer grained responses to enumerate the validation result of the term, distinct from the code. Ontoserver currently validates the term if it's either the PT or FSN. If not specified, the latest "default" edition for the server is used ie AU langrefset. Note: Potential to make two calls - one without the term to check the code and then another with the term to check that also.

PJ: The value-set will define the language used, but this would not apply when validating against the code-system.

If the URI that is passed is an edition, then hopefully we could look up a default (set?) of language reference sets.

Case sensitivity: To be discussed.

10 July 2018

Determined that it would be best to have multiple response parameters to capture the various possible scenarios so the client can choose the correct behaviour. Specifically discussed was adding a well known set of extension parameters which can be considered as optional response parameters in the specification in future.

The table below is intended to be built out to cover the various scenarios. Note that the philosophy with the boolean result parameter from validate-code is that it should return true if the code is correct, and only false if the text provided does not match at all, this is considered most fault tolerant for clients that don't use the additional parameters to discern the result more closely.

scenario			result	message	display	code validity code	display validity code
code	term	ValueSet term					
valid code	no term	no	true		code system preferred display	valid code in ValueSet	-
valid code	no term	yes	true		ValueSet display for code	valid code in ValueSet	-
code not in ValueSet	no term	n/a	false	explanation the code is invalid	code system preferred display	valid code not in ValueSet	-
code not in ValueSet	preferred display	n/a	false	explanation the code is invalid	code system preferred display	valid code not in ValueSet	correct display term
code not in ValueSet	acceptable term	n/a	false	explanation the code is invalid and warning that the display was not the preferred display	code system preferred display	valid code in ValueSet	acceptable display term
code not in ValueSet	active but unacceptable term for locale	n/a	false	explanation the code is invalid and warning that the display matches an active term for the code, but not an acceptable one	code system preferred display	valid code in ValueSet	active display term
code not in ValueSet	inactive /historical term for code	n/a	false	explanation the code is invalid and warning the provided term is historical for the code and not active	code system preferred display	valid code in ValueSet	historical display term
code not in ValueSet	unknown term for code system	n/a	false	explanation the code is invalid and explanation that the term was unknown	code system preferred display	valid code in ValueSet	unknown term
valid code	preferred display	no	true		code system preferred display	valid code in ValueSet	correct display term
valid code	preferred display	yes	true	explanation the display was valid for the code but didn't match the ValueSet	ValueSet display for code	valid code in ValueSet	correct display term
valid code	acceptable term	no	true	warning that the display was not the preferred display	code system preferred display	valid code in ValueSet	acceptable display term
valid code	acceptable term	yes	true	explanation the display was not preferred for the code and didn't match the ValueSet	ValueSet display for code	valid code in ValueSet	correct display term
valid code	active but unacceptable term for locale	no	true	warning that the display matches an active term for the code, but not an acceptable one	code system preferred display	valid code in ValueSet	active display term
valid code	active but unacceptable term for locale	yes	true	warning the display was not acceptable for the code and didn't match the ValueSet	ValueSet display for code	valid code in ValueSet	correct display term
valid code	inactive /historical term for code	no	true	warning the provided term is historical for the code and not active	code system preferred display	valid code in ValueSet	historical display term
valid code	inactive /historical term for code	yes	true	warning the provided term is historical for the code and didn't match the ValueSet	ValueSet display for code	valid code in ValueSet	correct display term
valid code	unknown term for code system	no	false	explanation that the term was unknown	code system preferred display	valid code in ValueSet	unknown term
valid code	unknown term for code system	yes	true		ValueSet display for code	valid code in ValueSet	correct display term
known code for code system family, but not in code system version	no term	n/a	false	explanation the code was not valid for the code system version	code system preferred display	known code not in code system version	-
known code for code system family, but not in code system version	preferred display	n/a	false	explanation the code was not valid for the code system version	code system preferred display	known code not in code system version	correct display term

known code for code system family, but not in code system version	acceptable term	n/a	false	explanation the code was not valid for the code system version	code system preferred display	known code not in code system version	acceptable display term
known code for code system family, but not in code system version	active but unacceptable term for locale	n/a	false	explanation the code was not valid for the code system version	code system preferred display	known code not in code system version	active display term
known code for code system family, but not in code system version	inactive /historical term for code	n/a	false	explanation the code was not valid for the code system version	code system preferred display	known code not in code system version	historical display term
known code for code system family, but not in code system version	unknown term for code system	n/a	false	explanation the code was not valid for the code system version	code system preferred display	known code not in code system version	unknown term
syntactically valid code (i.e. concept SCTID) but not known to any version on the server	unknown or no term	n/a	false	explanation the code was not valid for any known version for this code system	-	unknown but syntactically valid code	
valid identifier for component other than a concept (e.g. description or relationship)	unknown or no term	n/a	false	explanation the code is not for the right type of component	-	incorrect identifier type	
syntactically invalid code	unknown or no term	n/a	false	explanation the code is syntactically invalid, doesn't meet the code system's scheme	-	code not valid for code system scheme	

code validity and display validity are intended to be extensible bindings to a Code datatype.

This table needs further refinement and consideration of particularly whether all scenarios are worth the effort. Description handling is somewhat SNOMED CT specific, but generalisable to other code systems as most would fall under a subset of the scenarios (i.e. may not draw all the distinctions SNOMED CT does wrt term preferences).

Case significance status in SNOMED CT should also be taken into account, so "Heart structure" and "heart structure" would be matched equally for 80 891009 and get the same display validity code as they are equivalent once case significance is taken into account.