digital square

FHIR® Webinar

Implementation Guides

July 28, 2021

Agenda & Structure

- House rules, intro, agenda
- 1 Recap:
  - FHIR basics and Profiling
  - Recap: Terminologies
- 2: ImplementationGuides
  - ImplementationGuide: what, when, how
  - FHIR Artifacts
  - Tools and process considerations
- 3. Practical example walkthrough

Remarks and disclaimers

- FHIR® is the registered trademark of Health Level Seven® (HL7®) International.
- The use of the FHIR® trademark does not constitute endorsement of this course/product/service by HL7 ®.
- This is not an official HL7 training. For such training opportunities, you are encouraged to
  - http://www.hl7.org/training
- This presentation is a collection of freely available materials.
  - All diagrams, except those that state otherwise, are original materials or taken from the FHIR website and support materials; all screenshots are from the FHIR website
- This presentation is shared under a Creative Commons Attribution 4.0 (CC BY 4.0) license - (ok to share and adapt if credits are given)

Goals

1. Recall the basic of FHIR® with regards to Profiling
    1. FHIR (and FHIR profiling) is about technical, computable artifacts
    2. Resources, Data Structures and Data Elements
    3. Terminologies – when to use, ValueSets, CodeSystems
2. Understand what is a FHIR® ImplementationGuide
    1. Concepts
    2. Tools
    3. Practices
3. Develop a simple ImplementationGuide

Setup your toolbox

- We’ll use open-source tools
  - Github account is required except for local experimentation
    - (Local implementation is harder to do, so GitHub account is highly recommended)
  - Github client is recommended – e.g. github desktop
  - Local build (not required if you just want to learn and experiment a little):
    - JAVA (JDK)
    - Jekyll (https://jekyllrb.com/docs/installation)
    - Sushi (https://fshschool.org/docs/sushi/installation) : npm install -g fsh-sushi
      - you need to install node.js if you don’t have it (https://nodejs.org/)
      - May need to update your settings in Windows:

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

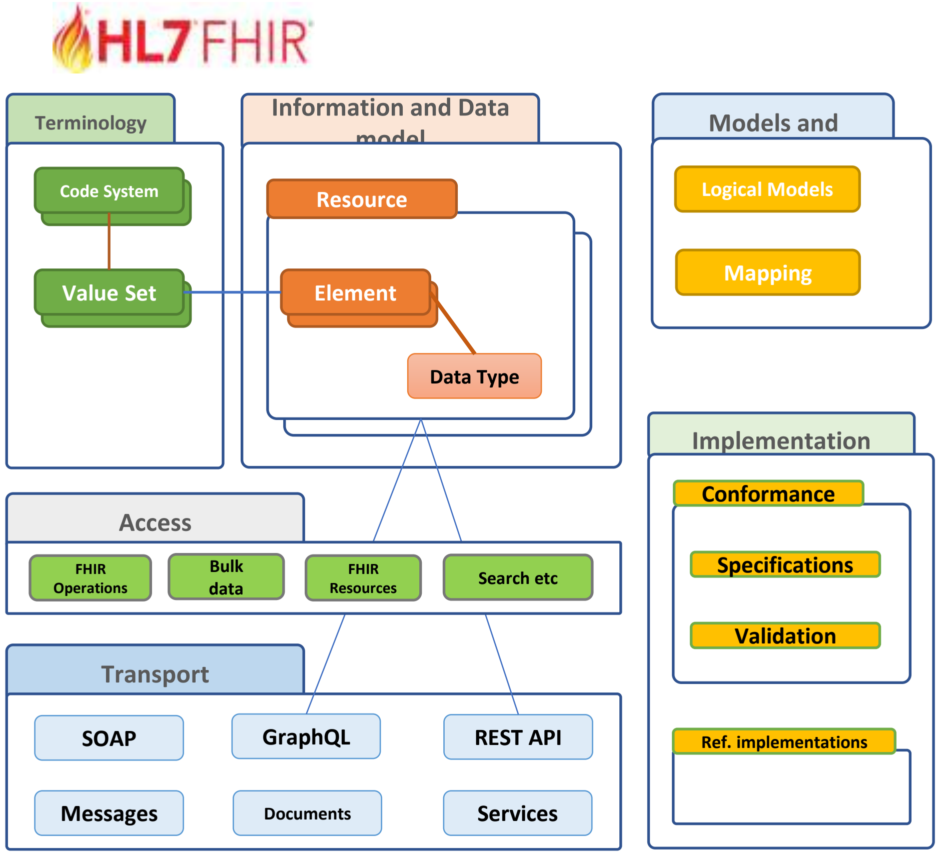
https:/go.microsoft.com/fwlink/?LinkID=135170

Part 1
Quick recap – FHIR, profiling, terminologies

FHIR Foundations and Profiling

FHIR Profiling

- FHIR® resources and profiles
  - Core resources are represent the common agreed data sets for exchange
  - *Can be Constrained and Extended*
- FHIR Terminologies
  - Define our own valuesets
  - If needed, CodeSystems, etc.



Profiling is done technically – and FHIR has a language for that

Profiling = defining FHIR content

- A “profile” is the name given to a constrained resource in FHIR®.
  - Profiled Resources are derived from other FHIR resources (or from profiles)
  - Example:
    - MedicationPrescriptionLine (profile of MedicationRequest).
    - MyPatient (Profile of USCore Patient)

- In FHIR, everything is defined with resources
  - StructureDefinition resource defines resources
  - ElementDefinition defines the individual data elements
  - ValueSets and CodeSystems define terminologies

http://build.fhir.org/profiling.html

Profiling data structures

- Select the right profile to constrain - from core specification or from existing profiles

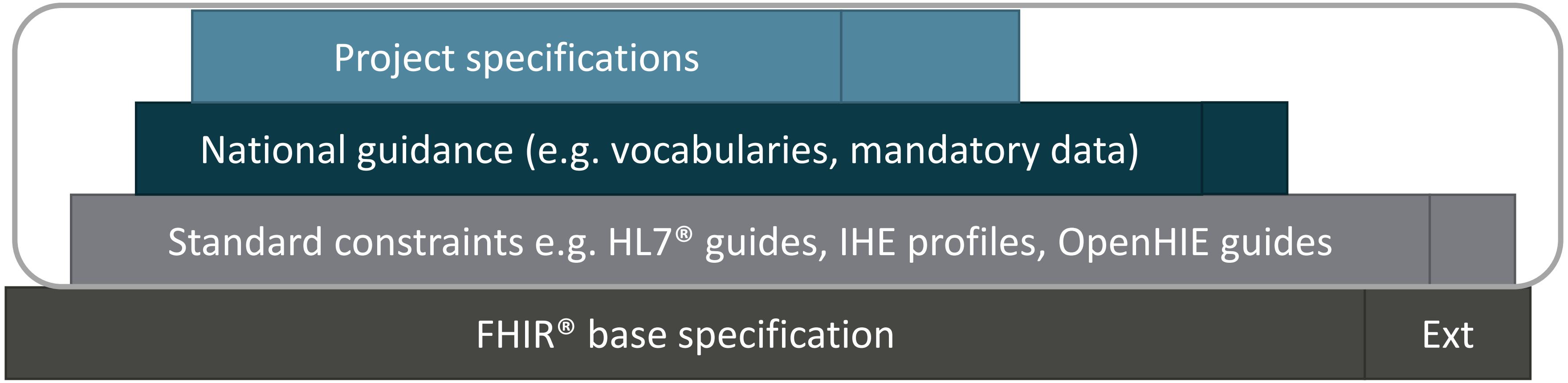
- Take one resource as base, (re)define the data elements by adding constraints
  - Change the name
  - Change the cardinality (0..0 removes the element, 1..1 or 1..* makes it mandatory)

- Extensions:
  - Take the Extension resource, add (Differential) constraints to its elements, and define context

- Update terminologies

Profiling in layers

- Profiling FHIR means constraining a FHIR specification
  - Fixing or binding some aspects of the specification
  - Defining which expansions to use
- This allows a layered specification – use it.



Project specifications

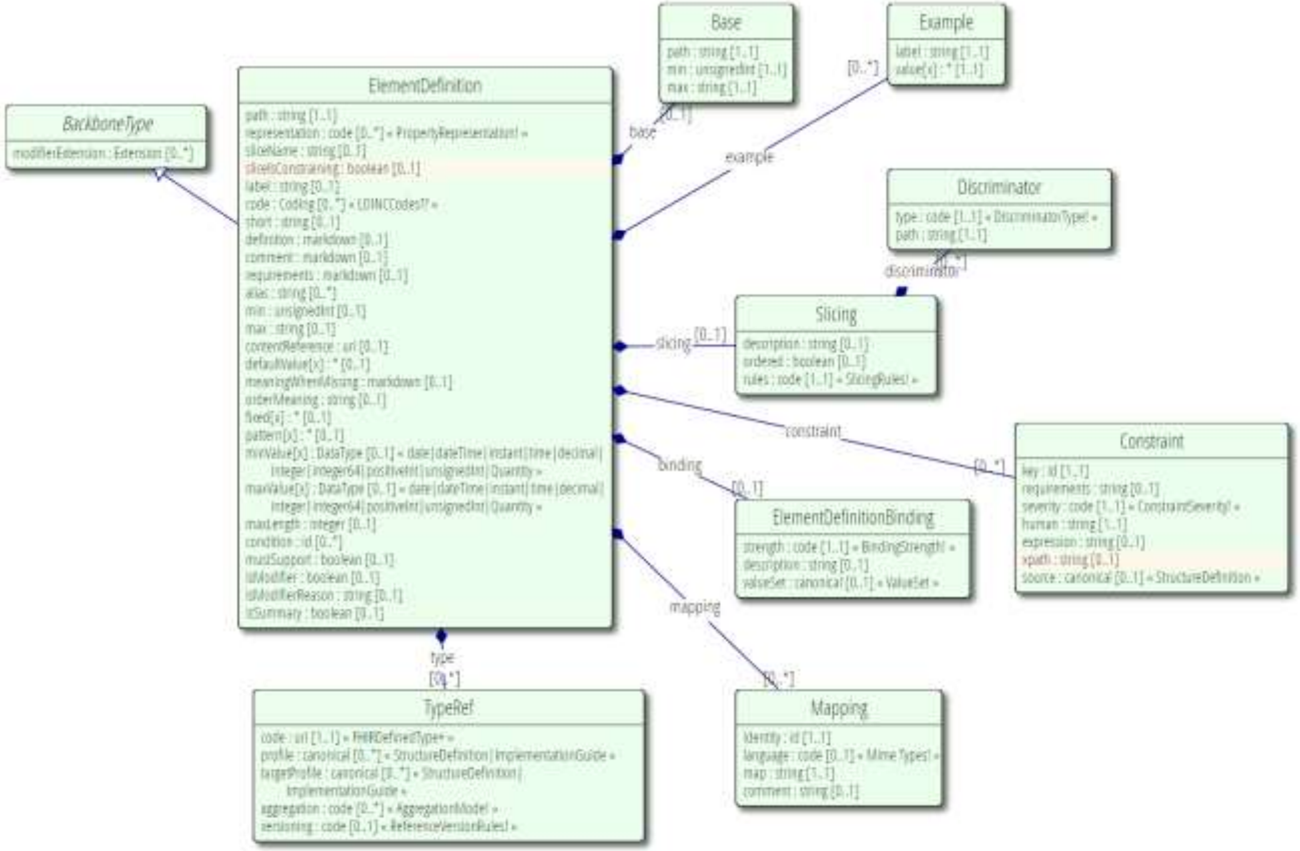
National guidance (e.g. vocabularies, mandatory data)

Standard constraints e.g. HL7® guides, IHE profiles, OpenHIE guides

FHIR® base specification

ElementDefinition

- Every element has its definition –data type, cardinality, binding…



BackboneType
modifierExtension : Extension [0..*]

ElementDefinition
path : string [1..1]
representation : code [0..*] « PropertyRepresentation! »
sliceName : string [0..1]
sliceIsConstraining : boolean [0..1]
label : string [0..1]
code : Coding [0..*] « LOINCCodes?? »
short : string [0..1]
definition : markdown [0..1]
comment : markdown [0..1]
requirements : markdown [0..1]
alias : string [0..*]
min : unsignedInt [0..1]
max : string [0..1]
contentReference : uri [0..1]
defaultValue[x] : * [0..1]
meaningWhenMissing : markdown [0..1]
orderMeaning : string [0..1]
fixed[x] : * [0..1]
pattern[x] : * [0..1]
minValue[x] : DataType [0..1] « date|dateTime|instant|time|decimal|integer|integer64|positiveInt|unsignedInt|Quantity »
maxValue[x] : DataType [0..1] « date|dateTime|instant|time|decimal|integer|integer64|positiveInt|unsignedInt|Quantity »
maxLength : integer [0..1]
condition : id [0..*]
mustSupport : boolean [0..1]
isModifier : boolean [0..1]
isModifierReason : string [0..1]
isSummary : boolean [0..1]

Base
path : string [1..1]
min : unsignedInt [1..1]
max : string [1..1]

Example
label : string [1..1]
value[x] : * [1..1]

Discriminator
type : code [1..1] « DiscriminatorType! »
path : string [1..1]

Slicing
description : string [0..1]
ordered : boolean [0..1]
rules : code [1..1] « SlicingRules! »

Constraint
key : id [1..1]
requirements : string [0..1]
severity : code [1..1] « ConstraintSeverity! »
human : string [1..1]
expression : string [0..1]
xpath : string [0..1]
source : canonical [0..1] « StructureDefinition »

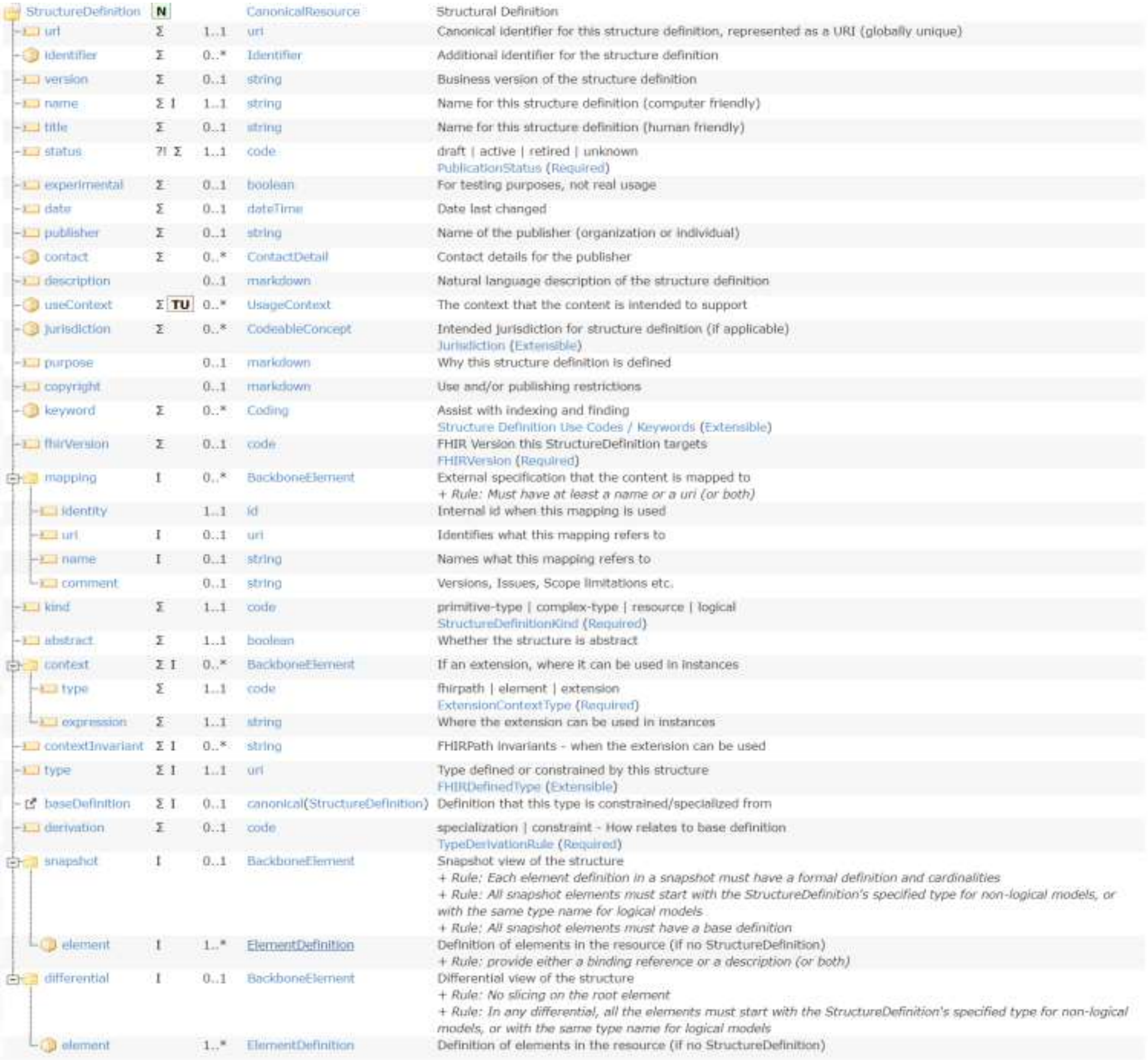
ElementDefinitionBinding
strength : code [1..1] « BindingStrength! »
description : string [0..1]
valueSet : canonical [0..1] « ValueSet »

TypeRef
code : uri [1..1] « FHIRDefinedType+ »
profile : canonical [0..*] « StructureDefinition|ImplementationGuide »
targetProfile : canonical [0..*] « StructureDefinition|ImplementationGuide »
aggregation : code [0..*] « AggregationMode! »
versioning : code [0..1] « ReferenceVersionRules! »

Mapping
identity : id [1..1]
language : code [0..1] « Mime Types! »
map : string [1..1]
comment : string [0..1]

StructureDefinition

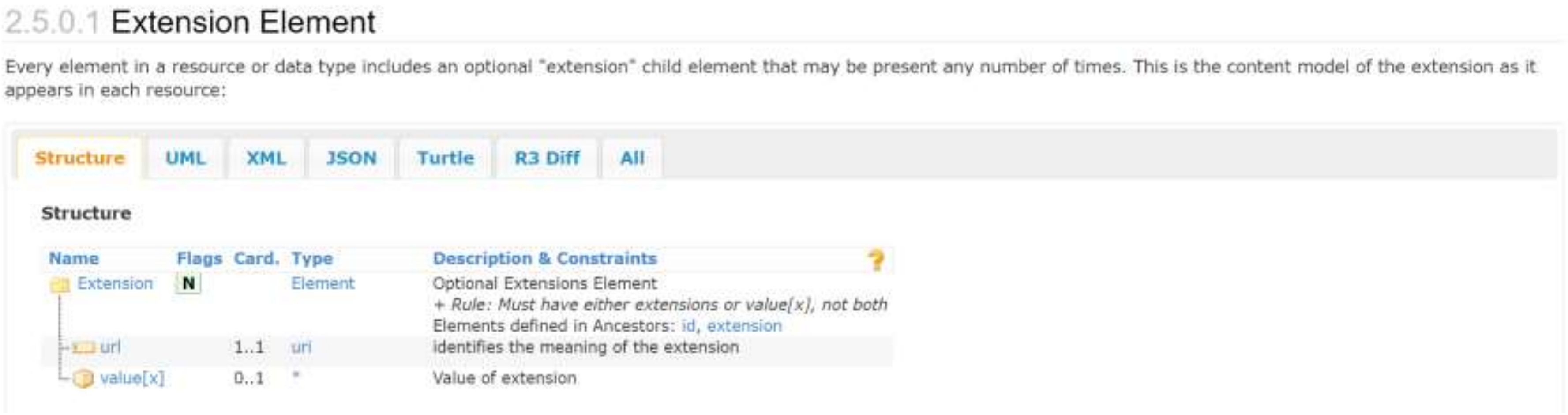
- Defines a data structure – a set of elements
  - Snapshot – full structure
  - Differential – difference to base
- Can be used to define Logical Models – an abstract representation of a data structure



StructureDefinition	N		CanonicalResource	Structural Definition
url	Σ	1..1	uri	Canonical identifier for this structure definition, represented as a URI (globally unique)
identifier	Σ	0..*	Identifier	Additional identifier for the structure definition
version	Σ	0..1	string	Business version of the structure definition
name	Σ I	1..1	string	Name for this structure definition (computer friendly)
title	Σ	0..1	string	Name for this structure definition (human friendly)
status	?! Σ	1..1	code	draft | active | retired | unknown PublicationStatus (Required)
experimental	Σ	0..1	boolean	For testing purposes, not real usage
date	Σ	0..1	dateTime	Date last changed
publisher	Σ	0..1	string	Name of the publisher (organization or individual)
contact	Σ	0..*	ContactDetail	Contact details for the publisher
description		0..1	markdown	Natural language description of the structure definition
useContext	Σ TU	0..*	UsageContext	The context that the content is intended to support
jurisdiction	Σ	0..*	CodeableConcept	Intended jurisdiction for structure definition (if applicable) Jurisdiction (Extensible)
purpose		0..1	markdown	Why this structure definition is defined
copyright		0..1	markdown	Use and/or publishing restrictions
keyword	Σ	0..*	Coding	Assist with indexing and finding Structure Definition Use Codes / Keywords (Extensible)
fhirVersion	Σ	0..1	code	FHIR Version this StructureDefinition targets FHIRVersion (Required)
mapping	I	0..*	BackboneElement	External specification that the content is mapped to *+ Rule: Must have at least a name or a uri (or both)*
identity		1..1	id	Internal id when this mapping is used
uri	I	0..1	uri	Identifies what this mapping refers to
name	I	0..1	string	Names what this mapping refers to
comment		0..1	string	Versions, Issues, Scope limitations etc.
kind	Σ	1..1	code	primitive-type | complex-type | resource | logical StructureDefinitionKind (Required)
abstract	Σ	1..1	boolean	Whether the structure is abstract
context	Σ I	0..*	BackboneElement	If an extension, where it can be used in instances
type	Σ	1..1	code	fhirpath | element | extension ExtensionContextType (Required)
expression	Σ	1..1	string	Where the extension can be used in instances
contextInvariant	Σ I	0..*	string	FHIRPath invariants - when the extension can be used
type	Σ I	1..1	uri	Type defined or constrained by this structure FHIRDefinedType (Extensible)
baseDefinition	Σ I	0..1	canonical(StructureDefinition)	Definition that this type is constrained/specialized from
derivation	Σ	0..1	code	specialization | constraint - How relates to base definition TypeDerivationRule (Required)
snapshot	I	0..1	BackboneElement	Snapshot view of the structure *+ Rule: Each element definition in a snapshot must have a formal definition and cardinalities* *+ Rule: All snapshot elements must start with the StructureDefinition's specified type for non-logical models, or with the same type name for logical models* *+ Rule: All snapshot elements must have a base definition*
element	I	1..*	ElementDefinition	Definition of elements in the resource (if no StructureDefinition) *+ Rule: provide either a binding reference or a description (or both)*
differential	I	0..1	BackboneElement	Differential view of the structure *+ Rule: No slicing on the root element* *+ Rule: In any differential, all the elements must start with the StructureDefinition's specified type for non-logical models, or with the same type name for logical models*
element		1..*	ElementDefinition	Definition of elements in the resource (if no StructureDefinition)

http://build.fhir.org/structuredefinition

Extensions



2.5.0.1 Extension Element

Every element in a resource or data type includes an optional "extension" child element that may be present any number of times. This is the content model of the extension as it appears in each resource:

Structure

Name	Flags	Card.	Type	Description & Constraints
Extension	N		Element	Optional Extensions Element *+ Rule: Must have either extensions or value[x], not both* Elements defined in Ancestors: id, extension
url		1..1	uri	identifies the meaning of the extension
value[x]		0..1	*	Value of extension

We can extend most anything in FHIR – resources, elements, datatypes

There are many standard extensions already out there:

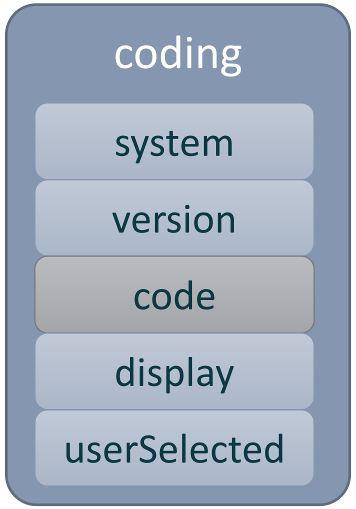
- HL7: https://build.fhir.org/extensibility-registry.html
- (we can define our own extensions)

Terminologies

Terminology - Coded Data elements

code

- Meaning and system are implied by the element
- Used in “core” aspects of the specification –
    - Bundle.type
    - Patient.gender



coding

system

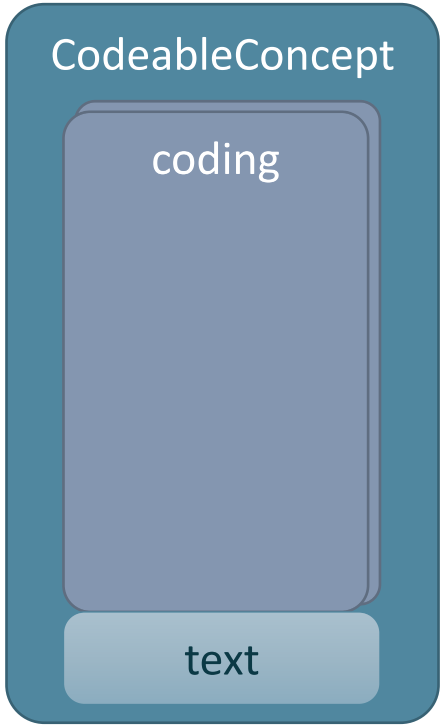
version

code

display

userSelected

- Version, display, userSelected are useful when implementing your own codes
- (not very commonly used)



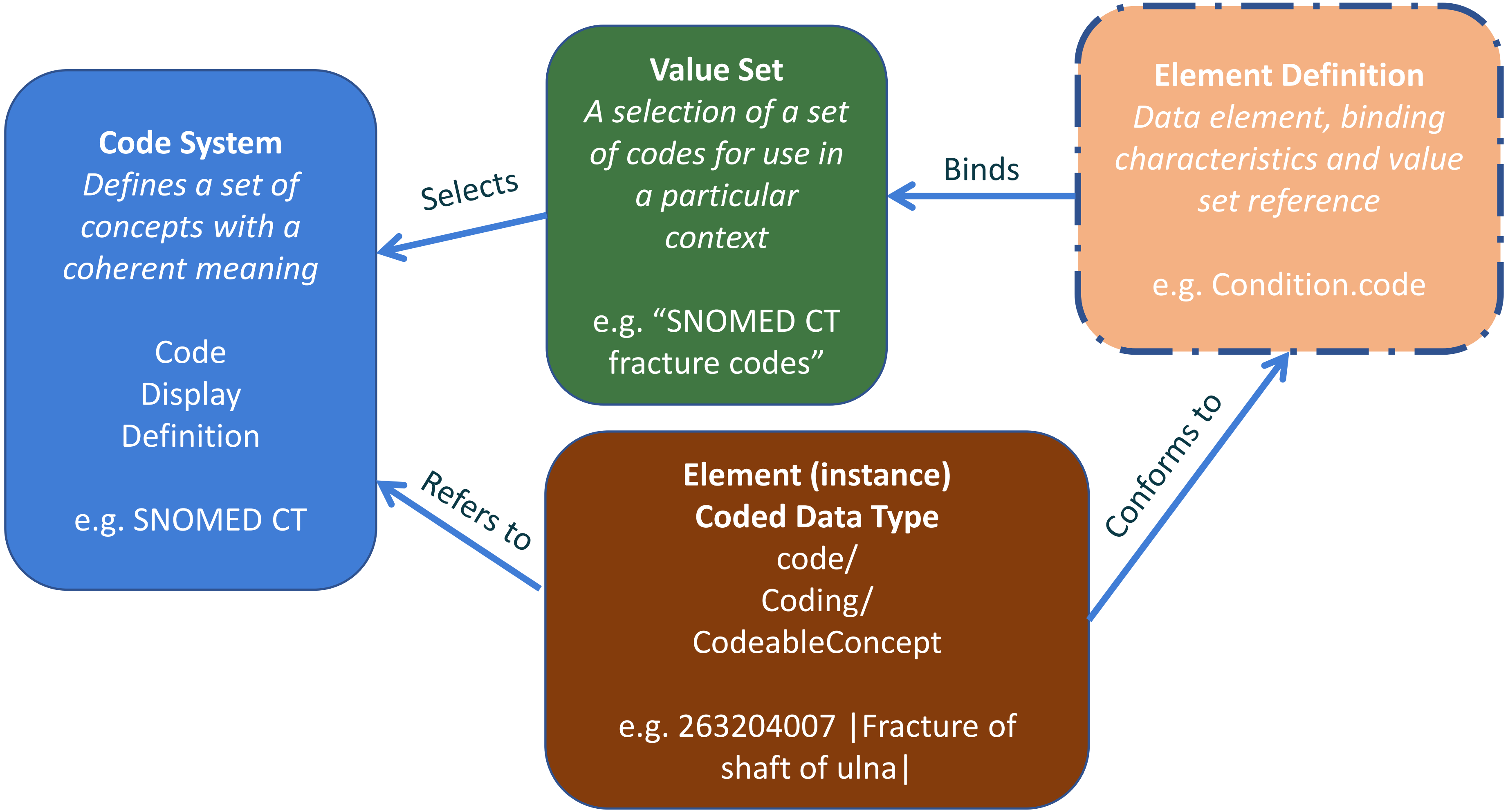
CodeableConcept

coding

text

- Same concept can be represented by different codes
- Text to represent the concept (or when a code is not available)

Terminology - Coded Data



Code System
Defines a set of concepts with a coherent meaning

Code
Display
Definition

e.g. SNOMED CT

Value Set
A selection of a set of codes for use in a particular context

e.g. “SNOMED CT fracture codes”

Element Definition
Data element, binding characteristics and value set reference

e.g. Condition.code

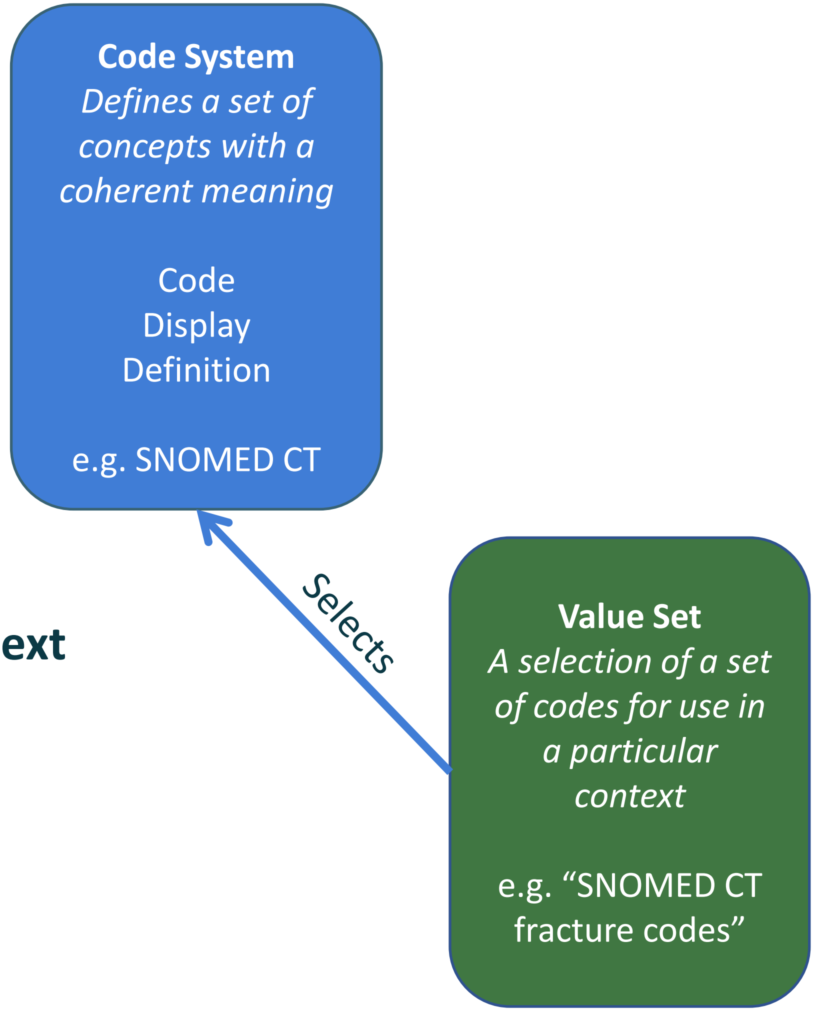
Element (instance)
Coded Data Type
code/
Coding/
CodeableConcept

e.g. 263204007 |Fracture of shaft of ulna|

CodeSystem and Value Set

- **Code systems** define **symbols** with **specific meanings**
  - E.g. LOINC, SNOMED, ICD-10, IETF language codes, local lab result codes, etc.

- **Value sets** define **collections of codes** for use in a **particular context**
  - Can come from a single code system or multiple code systems
    - E.g. “European country codes”
    - “The LOINC codes that I use”
    - All LOINC order codes
    - A particular SNOMED CT hierarchy
    - Substance codes plus “No known allergy”



Code System
Defines a set of concepts with a coherent meaning

Code
Display
Definition

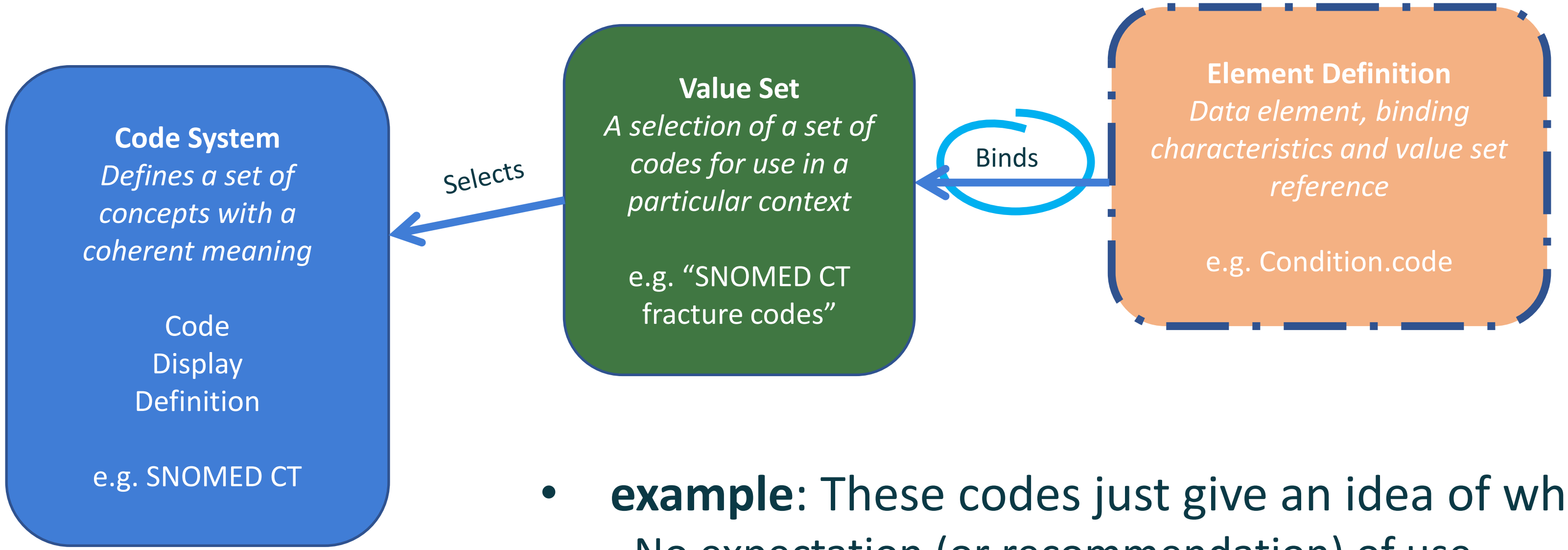
e.g. SNOMED CT

Selects

Value Set
A selection of a set of codes for use in a particular context

e.g. “SNOMED CT fracture codes”

Terminology Binding


Code System
Defines a set of concepts with a coherent meaning
Code
Display
Definition
e.g. SNOMED CT
Selects
Value Set
A selection of a set of codes for use in a particular context
e.g. “SNOMED CT fracture codes”
Binds
Element Definition
Data element, binding characteristics and value set reference
e.g. Condition.code

- **example**: These codes just give an idea of what you might use
  No expectation (or recommendation) of use
- **preferred**: You SHOULD use the specified codes
  But if you have a good reason, you can use something else instead – it is not required to use the specified codes in order to be conformant
- **extensible**: You must use the specified codes if they apply
  Free to use other codes or text if the value set doesn’t cover the concept
- **required**: You must use the specified codes
  Or omit the element if no code applies for the concept

Considerations when using profiling

Validate often

- Use examples
- Use Logical Models, discuss with stakeholders
- Set up a way of working, recurring calls, for review

Choose your strength

- Choose cardinalities: Making something mandatory can seem to make data will be “cleaner” but there is risk of information loss because non-compliant messages are rejected
- Same for terminology bindings

Part 2
ImplementationGuides

ImplementationGuide: What, when, how

ImplementationGuides

- The FHIR community uses ImplementationGuides to transport functional (and technical) requirements into FHIR technical specifications.

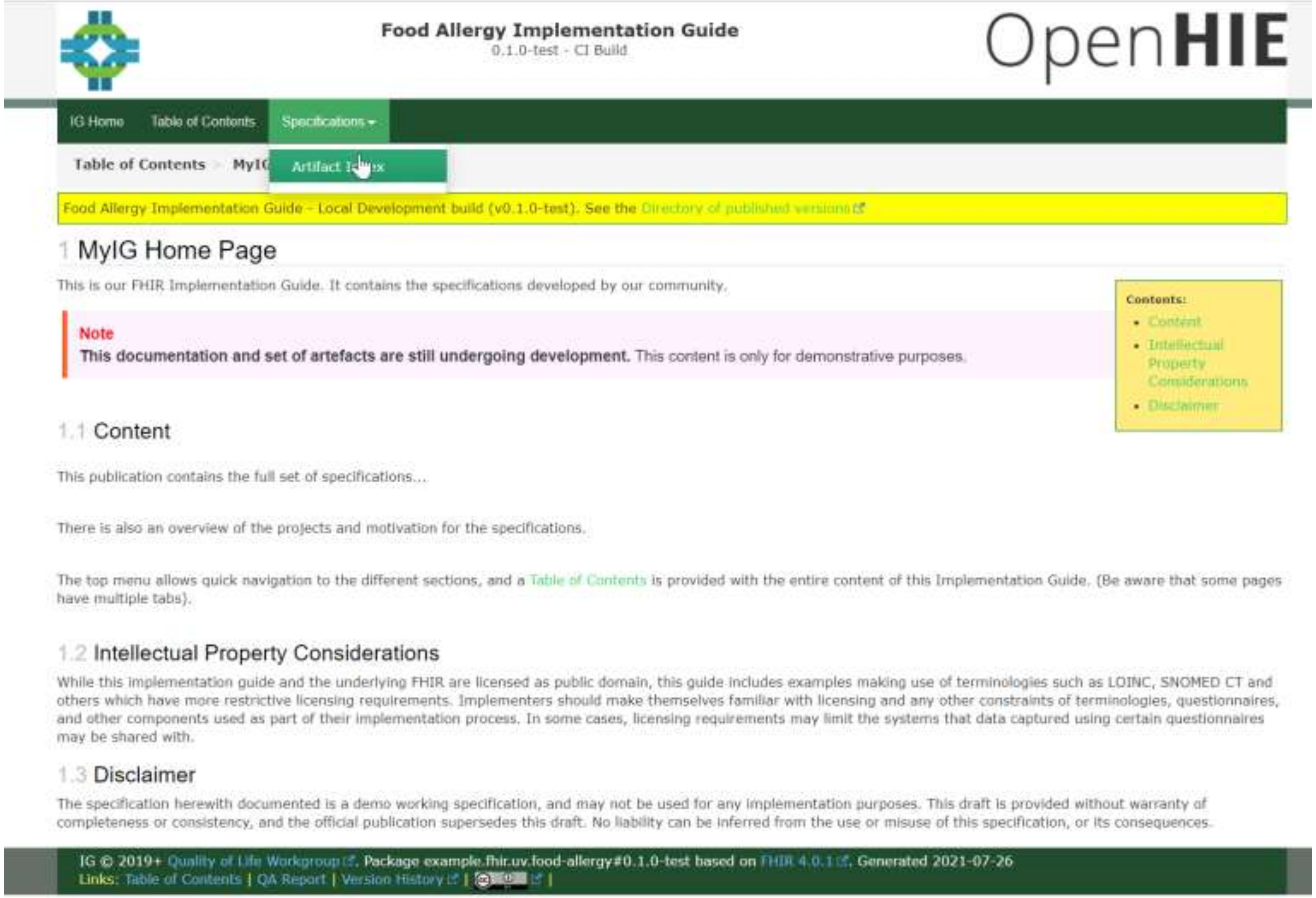
- ImplementationGuides and associated tooling produce a web publication that meets the common needs of implementers
  - Data specifications
  - Narratives
  - Examples
  - Etc.

Purpose of ImplementationGuides

- Requirements and expectations should define the technical specifications
  - Not the other way around
  - Of course, the existing technical base provides a good starting point – don’t reinvent if not needed; use best practices

- Purpose is important:
  - Be clear about the purpose: Are you defining your system? Or expectations for many systems?
  - When a constraint is applied, it cannot be removed in upper layers.
    - Be flexible with what you accept, strict with what you send.
    - Avoid systems to become non-compliant because of “ideal” constraints.

Example



Food Allergy Implementation Guide
0.1.0-test - CI Build

Food Allergy Implementation Guide - Local Development build (v0.1.0-test). See the Directory of published versions

1 MyIG Home Page

This is our FHIR Implementation Guide. It contains the specifications developed by our community.

Note
This documentation and set of artefacts are still undergoing development. This content is only for demonstrative purposes.

Contents:
- Content
- Intellectual Property Considerations
- Disclaimer

1.1 Content

This publication contains the full set of specifications...

There is also an overview of the projects and motivation for the specifications.

The top menu allows quick navigation to the different sections, and a Table of Contents is provided with the entire content of this Implementation Guide. (Be aware that some pages have multiple tabs).

1.2 Intellectual Property Considerations

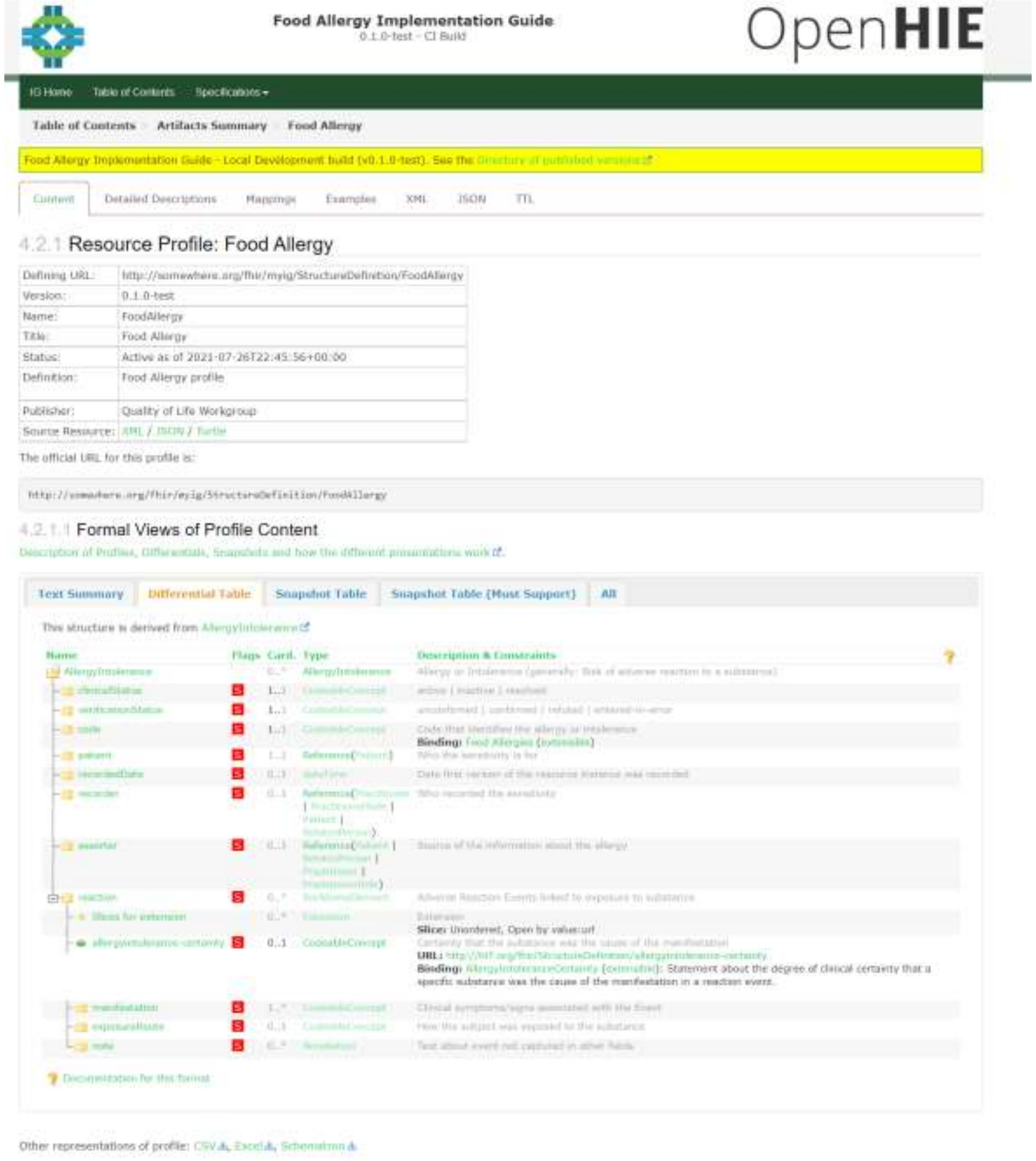
While this implementation guide and the underlying FHIR are licensed as public domain, this guide includes examples making use of terminologies such as LOINC, SNOMED CT and others which have more restrictive licensing requirements. Implementers should make themselves familiar with licensing and any other constraints of terminologies, questionnaires, and other components used as part of their implementation process. In some cases, licensing requirements may limit the systems that data captured using certain questionnaires may be shared with.

1.3 Disclaimer

The specification herewith documented is a demo working specification, and may not be used for any implementation purposes. This draft is provided without warranty of completeness or consistency, and the official publication supersedes this draft. No liability can be inferred from the use or misuse of this specification, or its consequences.

IG © 2019+ Quality of Life Workgroup. Package example.fhir.uv.food-allergy#0.1.0-test based on FHIR 4.0.1. Generated 2021-07-26
Links: Table of Contents | QA Report | Version History | |

https://costateixeira.github.io/FoodAllergy-Webinar/



Food Allergy Implementation Guide
0.1.0-test - CI Build

OpenHIE

IG Home | Table of Contents | Specifications ▾

Table of Contents › Artifacts Summary › Food Allergy

Food Allergy Implementation Guide - Local Development build (v0.1.0-test). See the Directory of published versions

Content | Detailed Descriptions | Mappings | Examples | XML | JSON | TTL

4.2.1 Resource Profile: Food Allergy

Defining URL:	http://somewhere.org/fhir/myig/StructureDefinition/FoodAllergy
Version:	0.1.0-test
Name:	FoodAllergy
Title:	Food Allergy
Status:	Active as of 2021-07-26T22:45:56+00:00
Definition:	Food Allergy profile
Publisher:	Quality of Life Workgroup
Source Resource:	XML / JSON / Turtle

The official URL for this profile is:

```
http://somewhere.org/fhir/myig/StructureDefinition/FoodAllergy
```

4.2.1.1 Formal Views of Profile Content

Description of Profiles, Differentials, Snapshots and how the different presentations work.

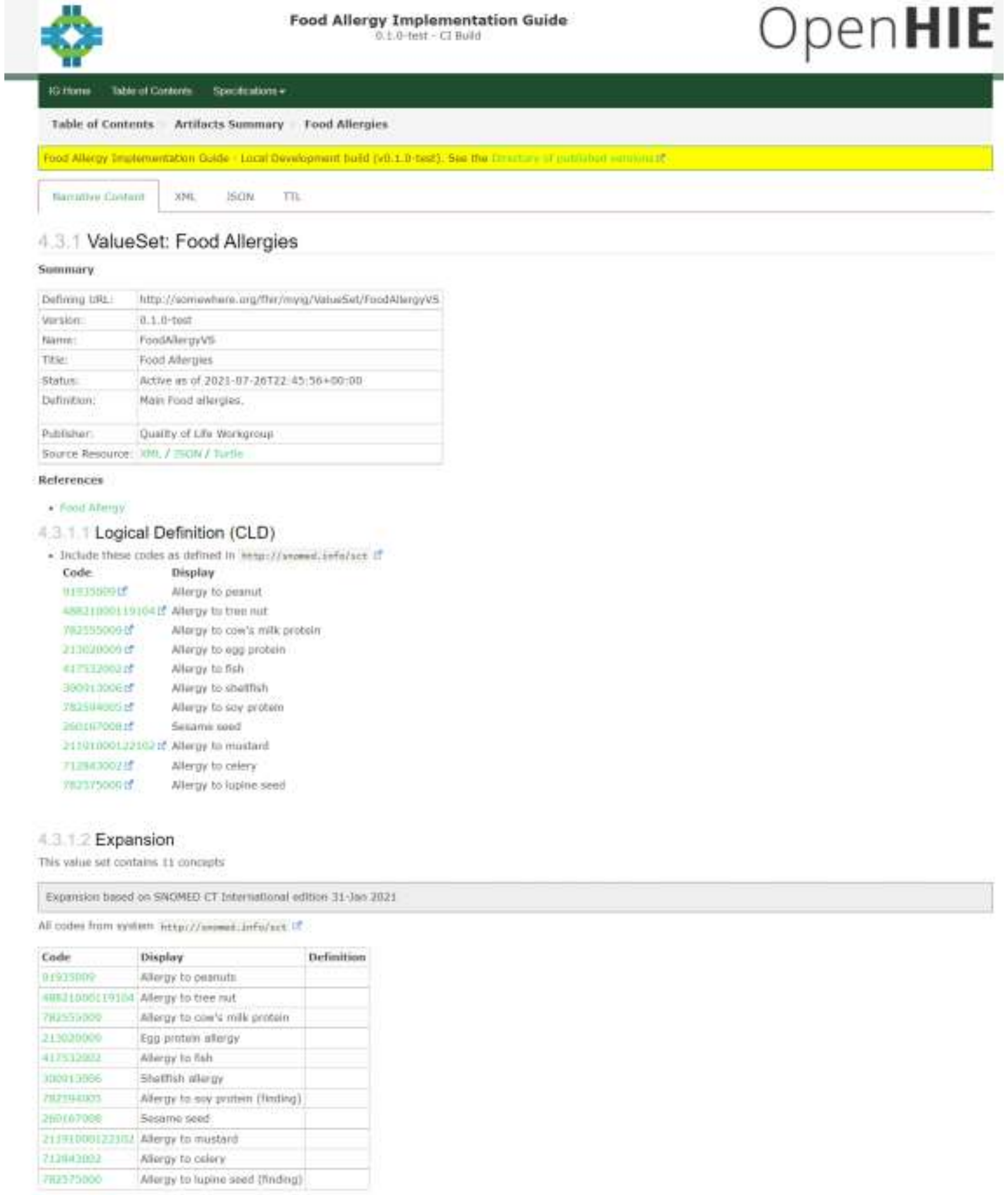
Text Summary | Differential Table | Snapshot Table | Snapshot Table (Must Support) | All

This structure is derived from AllergyIntolerance

Name	Flags	Card.	Type	Description & Constraints
AllergyIntolerance		0..*	AllergyIntolerance	Allergy or Intolerance (generally: Risk of adverse reaction to a substance)
clinicalStatus	S	1..1	CodeableConcept	active | inactive | resolved
verificationStatus	S	1..1	CodeableConcept	unconfirmed | confirmed | refuted | entered-in-error
code	S	1..1	CodeableConcept	Code that identifies the allergy or intolerance **Binding:** Food Allergies (extensible)
patient	S	1..1	Reference(Patient)	Who the sensitivity is for
recordedDate	S	0..1	dateTime	Date first version of the resource instance was recorded
recorder	S	0..1	Reference(Practitioner | PractitionerRole | Patient | RelatedPerson)	Who recorded the sensitivity
asserter	S	0..1	Reference(Patient | RelatedPerson | Practitioner | PractitionerRole)	Source of the information about the allergy
reaction	S	0..*	BackboneElement	Adverse Reaction Events linked to exposure to substance
Slices for extension		0..*	Extension	Extension **Slice:** Unordered, Open by value:url
allergyintolerance-certainty	S	0..1	CodeableConcept	Certainty that the substance was the cause of the manifestation **URL:** http://hl7.org/fhir/StructureDefinition/allergyintolerance-certainty **Binding:** AllergyIntoleranceCertainty (extensible): Statement about the degree of clinical certainty that a specific substance was the cause of the manifestation in a reaction event.
manifestation	S	1..*	CodeableConcept	Clinical symptoms/signs associated with the Event
exposureRoute	S	0..1	CodeableConcept	How the subject was exposed to the substance
note	S	0..*	Annotation	Text about event not captured in other fields

Documentation for this format

Other representations of profile: CSV, Excel, Schematron



Food Allergy Implementation Guide
0.1.0-test - CI Build

OpenHIE

IG Home | Table of Contents | Specifications ▾

Table of Contents › Artifacts Summary › Food Allergies

Food Allergy Implementation Guide - Local Development build (v0.1.0-test). See the Directory of published versions

Narrative Content | XML | JSON | TTL

4.3.1 ValueSet: Food Allergies

Summary

Defining URL:	http://somewhere.org/fhir/myig/ValueSet/FoodAllergyVS
Version:	0.1.0-test
Name:	FoodAllergyVS
Title:	Food Allergies
Status:	Active as of 2021-07-26T22:45:56+00:00
Definition:	Main Food allergies.
Publisher:	Quality of Life Workgroup
Source Resource:	XML / JSON / Turtle

References

- Food Allergy

4.3.1.1 Logical Definition (CLD)

- Include these codes as defined in `http://snomed.info/sct`

Code	Display
91935009	Allergy to peanut
48821000119104	Allergy to tree nut
782555009	Allergy to cow's milk protein
213020009	Allergy to egg protein
417532002	Allergy to fish
300913006	Allergy to shellfish
782594005	Allergy to soy protein
260167008	Sesame seed
21191000122102	Allergy to mustard
712843002	Allergy to celery
782575000	Allergy to lupine seed

4.3.1.2 Expansion

This value set contains 11 concepts

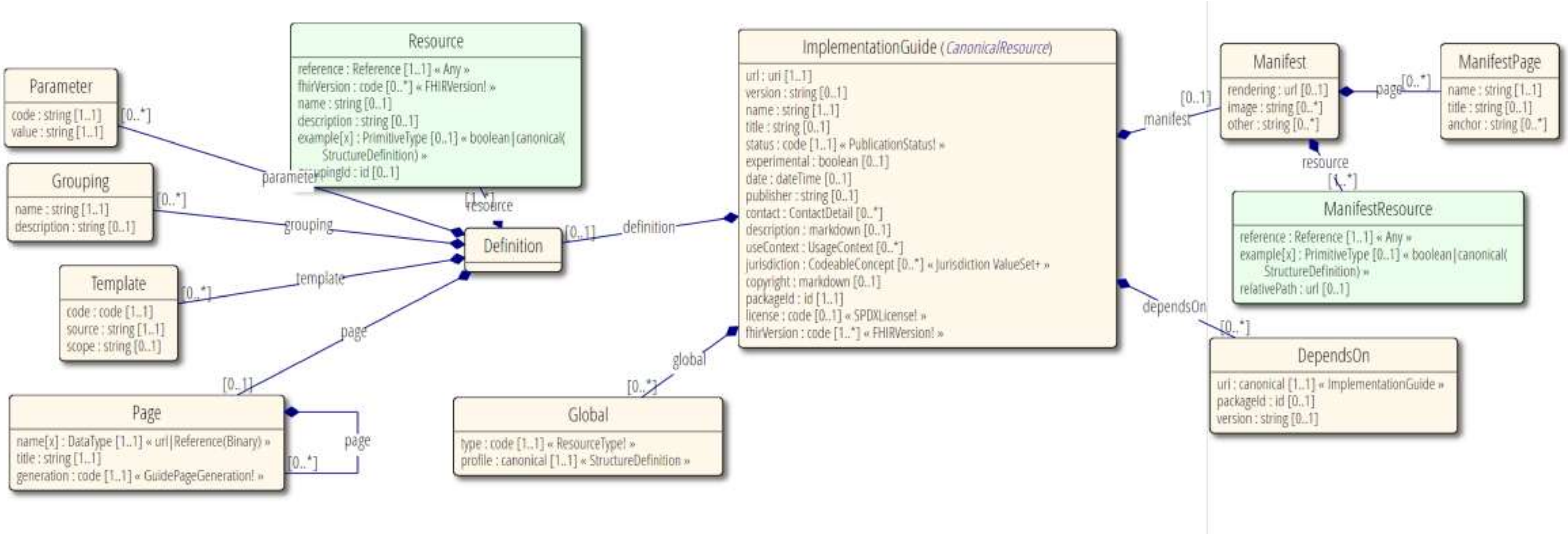
Expansion based on SNOMED CT International edition 31-Jan 2021

All codes from system `http://snomed.info/sct`

Code	Display	Definition
91935009	Allergy to peanuts	
48821000119104	Allergy to tree nut	
782555009	Allergy to cow's milk protein	
213020009	Egg protein allergy	
417532002	Allergy to fish	
300913006	Shellfish allergy	
782594005	Allergy to soy protein (finding)	
260167008	Sesame seed	
21191000122102	Allergy to mustard	
712843002	Allergy to celery	
782575000	Allergy to lupine seed (finding)	

FHIR artifacts

ImplementationGuide – a FHIR resource



ImplementationGuide (*CanonicalResource*)
url : uri [1..1]
version : string [0..1]
name : string [1..1]
title : string [0..1]
status : code [1..1] « PublicationStatus! »
experimental : boolean [0..1]
date : dateTime [0..1]
publisher : string [0..1]
contact : ContactDetail [0..*]
description : markdown [0..1]
useContext : UsageContext [0..*]
jurisdiction : CodeableConcept [0..*] « Jurisdiction ValueSet+ »
copyright : markdown [0..1]
packageId : id [1..1]
license : code [0..1] « SPDXLicense! »
fhirVersion : code [1..*] « FHIRVersion! »

Resource
reference : Reference [1..1] « Any »
fhirVersion : code [0..*] « FHIRVersion! »
name : string [0..1]
description : string [0..1]
example[x] : PrimitiveType [0..1] « boolean|canonical(StructureDefinition) »
groupingId : id [0..1]

Parameter
code : string [1..1]
value : string [1..1]

Grouping
name : string [1..1]
description : string [0..1]

Template
code : code [1..1]
source : string [1..1]
scope : string [0..1]

Definition

Page
name[x] : DataType [1..1] « url|Reference(Binary) »
title : string [1..1]
generation : code [1..1] « GuidePageGeneration! »

Global
type : code [1..1] « ResourceType! »
profile : canonical [1..1] « StructureDefinition »

Manifest
rendering : url [0..1]
image : string [0..*]
other : string [0..*]

ManifestPage
name : string [1..1]
title : string [0..1]
anchor : string [0..*]

ManifestResource
reference : Reference [1..1] « Any »
example[x] : PrimitiveType [0..1] « boolean|canonical(StructureDefinition) »
relativePath : url [0..1]

DependsOn
uri : canonical [1..1] « ImplementationGuide »
packageId : id [0..1]
version : string [0..1]

ImplementationGuide source artifacts

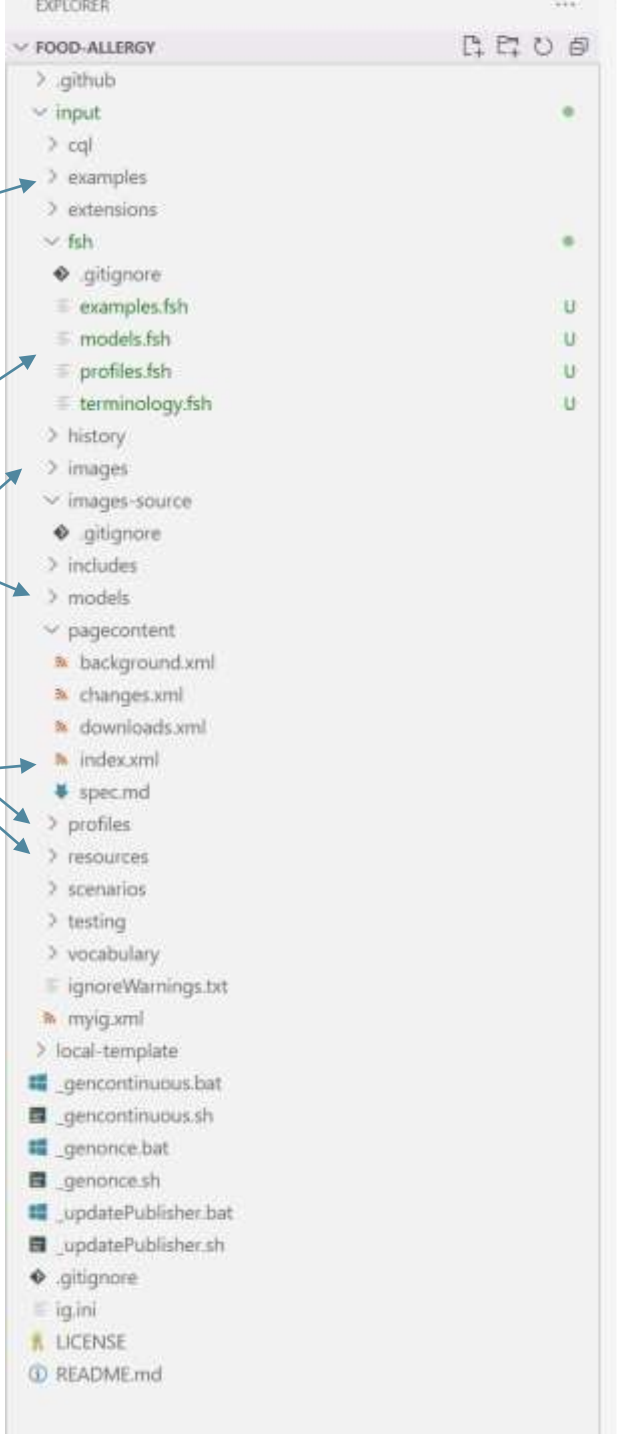
- Creating an ImplementationGuide means to create a set of files that provide the content

- This is why we use Version Control and repositories e.g. github

- ImplementationGuides follow a standard structure

ImplementationGuide source artifacts

- FHIR resources – in json, xml or ttl format
  - Profiles, logical models, valuesets, examples, etc.
- Shorthand resources (in files in dedicated folder)
- Narrative pages in markdown or xhtml format
- Images that are needed for the narrative



```
EXPLORER                                   ...
∨ FOOD-ALLERGY
  > .github
  ∨ input
    > cql
    > examples
    > extensions
    ∨ fsh
      .gitignore
      examples.fsh                          U
      models.fsh                            U
      profiles.fsh                          U
      terminology.fsh                       U
    > history
    > images
    ∨ images-source
      .gitignore
    > includes
    > models
    ∨ pagecontent
      background.xml
      changes.xml
      downloads.xml
      index.xml
      spec.md
    > profiles
    > resources
    > scenarios
    > testing
    > vocabulary
    ignoreWarnings.txt
    myig.xml
  > local-template
  _gencontinuous.bat
  _gencontinuous.sh
  _genonce.bat
  _genonce.sh
  _updatePublisher.bat
  _updatePublisher.sh
  .gitignore
  ig.ini
  LICENSE
  README.md
```

ImplementationGuide target artifacts

- Web page publication (as deployable HTML content)
  - Consistent navigation – so that everyone can find their way around others' IGs
  - Table of contents, menus, artifacts
  - Other technical artifacts e.g. packages for reusing, etc.

  - IG URL

Tools and process

Specification process

- Create the Implementation Guide
  - Get requirements - Data definitions, elements, terminologies
  - Create boilerplate / initial content
  - Check the FHIR community for existing guidance or interested people
  - Implement your content – profiles, extensions, terminologies, narrative, examples
    - Always checking if there is already something similar or reusable
  - Build your IG & repeat
- Publish and deploy

Tools needed

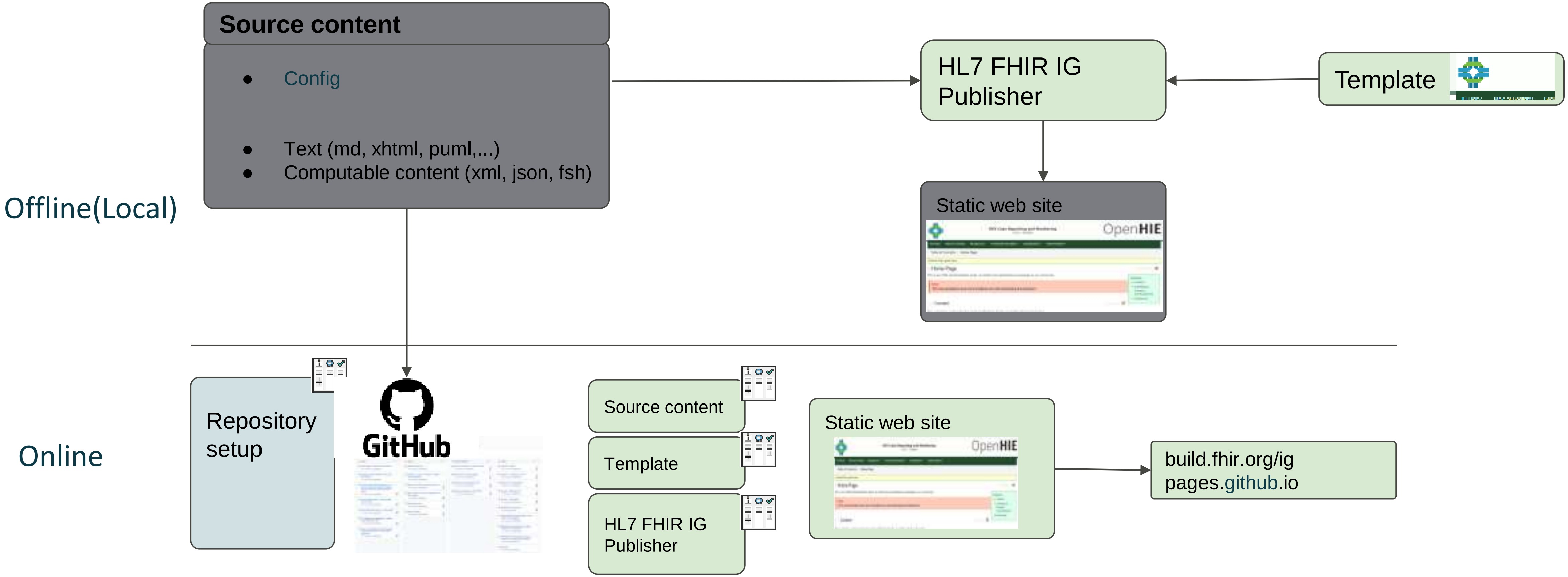
- Creating content:
    - (by hand, using any text editor)
    - Forge – free for non-commercial use - https://fire.ly/products/forge
    - FHIR Shorthand – a FHIR specification to type (little) text for profiling https://fshschool.org – allows you to experiment and share

- Publishing
    - Simplifier.net – a tool to produce and discover the ImplementationGuides from the community
    - FHIR Implementation Guide Publisher – an open source standard tool

- Editors and IDEs (for editing text): Visual Studio, Notepad++, …

- Repository / Version control – GitHub, GitLab, BitBucket..

Build process overview



Offline(Local)

Source content

- Config
- Text (md, xhtml, puml,...)
- Computable content (xml, json, fsh)

HL7 FHIR IG Publisher

Template

Static web site

Online

Repository setup

GitHub

Source content

Template

HL7 FHIR IG Publisher

Static web site

build.fhir.org/ig
pages.github.io

Publication process

- When we iterate through an ImplementationGuide, we want to share with others and keep track

- There are several ways to publish the “Development” version of an IG
  - If the IG is hosted on GitHub: (github.com/<org>/<repo>)
    - HL7 CI Builder publishes it on build.fhir.org/ig/<org>/<repo>
    - The online template provides a workflow which publishes it on <org>.github.io/<repo>

Part 3
Let’s do it

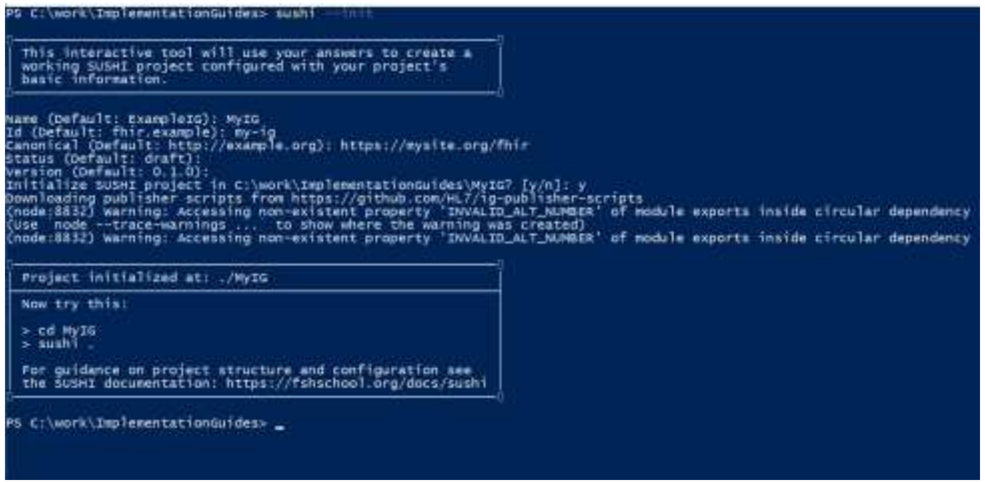
Tools used in this exercise

- Repository: GitHub + online build
- Git client: Github desktop
- IDE: VisualStudio
- Profiling: Sushi
- Template – OpenHIE IG template
- Publishing: ImplementationGuide Publisher – online and offline

Agenda / Steps

- **Create an IG and setup IG repository - local and online**
  - Sushi --init → upload
  - **Online template → download**
- **Configure & customize IG** (if you didn’t use sushi in previous step)
- **Add narrative**
- **Add Logical Model**
- **Add Profiles**
- **Add ValueSet**
- **Add Example**
- **Check output quality**

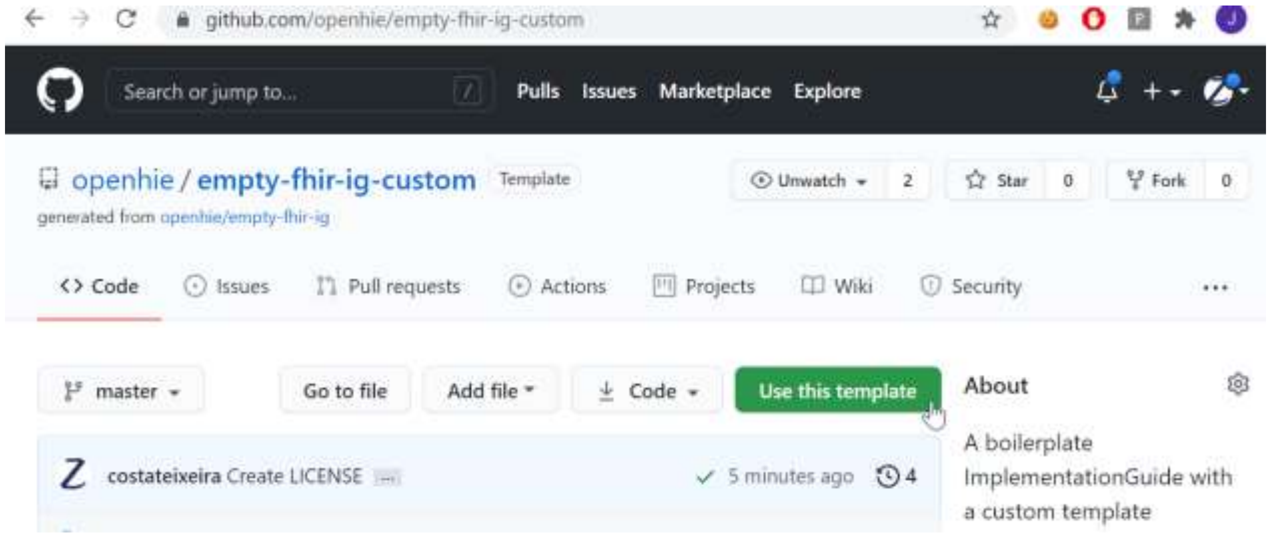
Creating your IG



```
PS C:\work\ImplementationGuides> sushi --init

  This interactive tool will use your answers to create a
  working SUSHI project configured with your project's
  basic information.

Name (Default: ExampleIG): MyIG
Id (Default: fhir.example): my-ig
Canonical (Default: http://example.org): https://mysite.org/fhir
Status (Default: draft):
Version (Default: 0.1.0):
Initialize SUSHI project in C:\work\ImplementationGuides\MyIG? [y/n]: y
Downloading publisher scripts from https://github.com/HL7/ig-publisher-scripts
(node:8832) Warning: Accessing non-existent property 'INVALID_ALT_NUMBER' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:8832) Warning: Accessing non-existent property 'INVALID_ALT_NUMBER' of module exports inside circular dependency

  Project initialized at: ./MyIG

  Now try this:

  > cd MyIG
  > sushi .

  For guidance on project structure and configuration see
  the SUSHI documentation: https://fshschool.org/docs/sushi

PS C:\work\ImplementationGuides> _
```

- Locally: **sushi --init**

  or
- **Remotely:** https://github.com/openhie/empty-fhir-ig-custom



github.com/openhie/empty-fhir-ig-custom

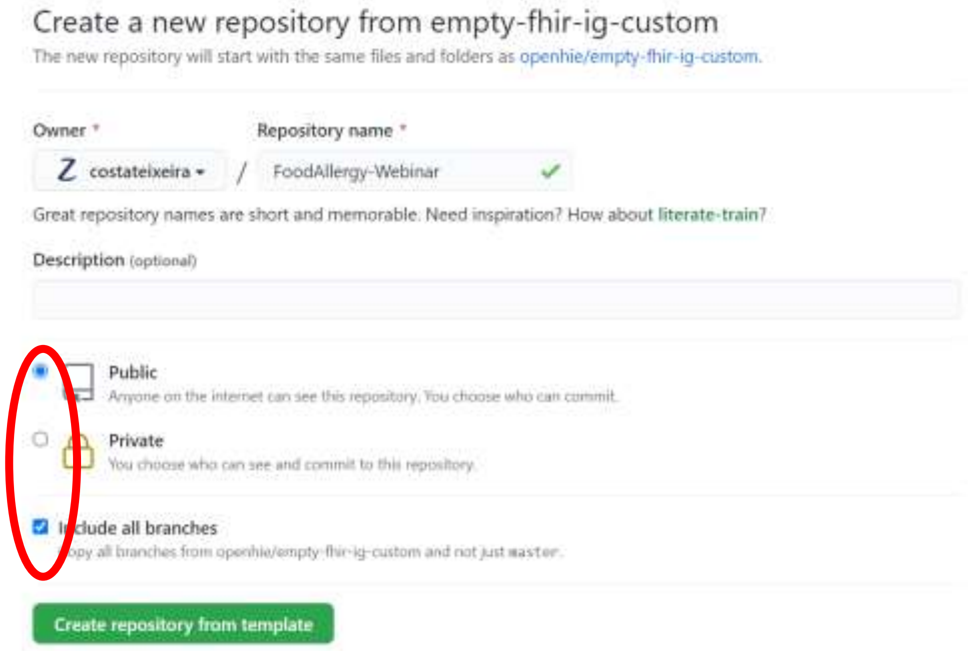
openhie / empty-fhir-ig-custom Template

generated from openhie/empty-fhir-ig

costateixeira Create LICENSE … ✓ 5 minutes ago 4

About

A boilerplate ImplementationGuide with a custom template



Create a new repository from empty-fhir-ig-custom

The new repository will start with the same files and folders as openhie/empty-fhir-ig-custom.

Owner * costateixeira / Repository name * FoodAllergy-Webinar ✓

Great repository names are short and memorable. Need inspiration? How about literate-train?

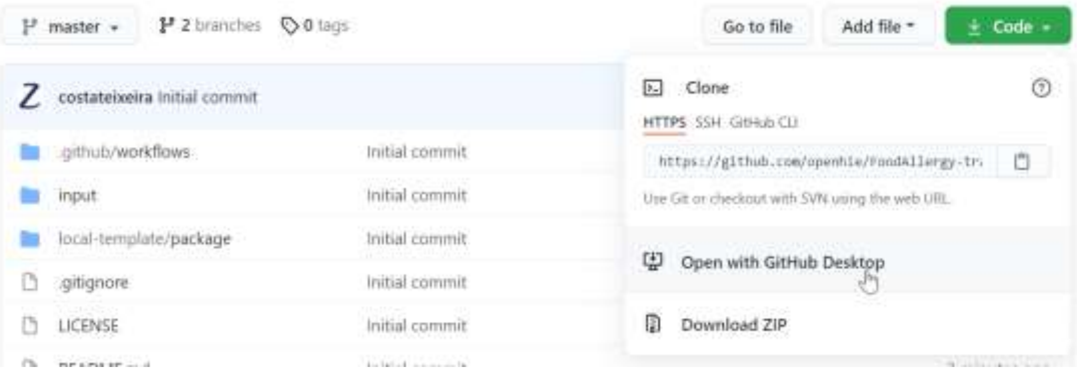
Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

☑ Include all branches
Copy all branches from openhie/empty-fhir-ig-custom and not just master.

Sync offline - online



costateixeira Initial commit

.github/workflows	Initial commit
input	Initial commit
local-template/package	Initial commit
.gitignore	Initial commit
LICENSE	Initial commit

```
https://github.com/openhie/FoodAllergy-tr
```

Use Git or checkout with SVN using the web URL.

1. Change your IG filename

- Just rename the xml
- Make sure you update the ig.ini file that points to it

- (not needed if you use sushi –init)

1. Adapt your IG id, name, etc.

- In the ig.xml, change
  - id
  - url
  - name
  - title
  - publisher
  - contact
  - description
  - packageId

(Not needed if you use sushi --init)

Build !

Build

- Locally: Run _genonce.bat / _genonce.sh
    - *The first time you need to download the publisher*
        - *just run _updatePublisher.bat / _updatePublisher.sh*

- Online: Setup online continuous build
    - If you use the template provided – just check that your repository uses github pages
        - **<org>.github.io/<repo>**
    - If you commit to the online repository, check it out: **build.fhir.org/ig/<org>/<repo>**
        - (instructions on https://github.com/FHIR/auto-ig-builder)

2. Add narrative pages

- **Narrative pages can be added by creating markdown or xhtml files – and adding them to the ImplementationGuide resource**
  - Create the .md or .xhtml file in the folder input/pagecontent
  - Reference the page in the ig xml
  - (Optionally, add a menu link to that page)

- Add a page to describe “Food Allergy Reporting and Sharing”
- Not forgetting to add it to the ImplementationGuide resource XML

3. Add a Logical Data Model (functional)

- *Logical models are StructureDefinitions, based on a special resource (Base)*

- Add a .fsh file to your repository
- You can try and share specific shorthand content using https://fshschool.org/FSHOnline/

- Publish for validation

Example content

- FoodAllergy
  - Patient (Mandatory)
  - Clinical status (Mandatory, coded)
  - Verification status (Mandatory, coded)
  - Allergen (Mandatory, coded) **– ASK FOR LIST OF CODES**
  - Date recorded (if known)
  - Recorder (if known)
  - Asserter (if known
  - History of known reactions
    - Manifestation (required)
    - Certitude (optional)
    - Exposure route (if known)
    - Note (if exists)

https://fshschool.org/FSHOnline/#/share/3eVQNWY

4. Add a profile

- *StructureDefinitions are sets of DataElements and their characteristics.*
- *StructureDefinitions can contain a Differential from a base resource or profile*

- Select your base profile - http://hl7.org/fhir/allergyintolerance.html
- Create a StructureDefinition that changes some of the elements
  - Cardinality
  - Short description
  - Definition

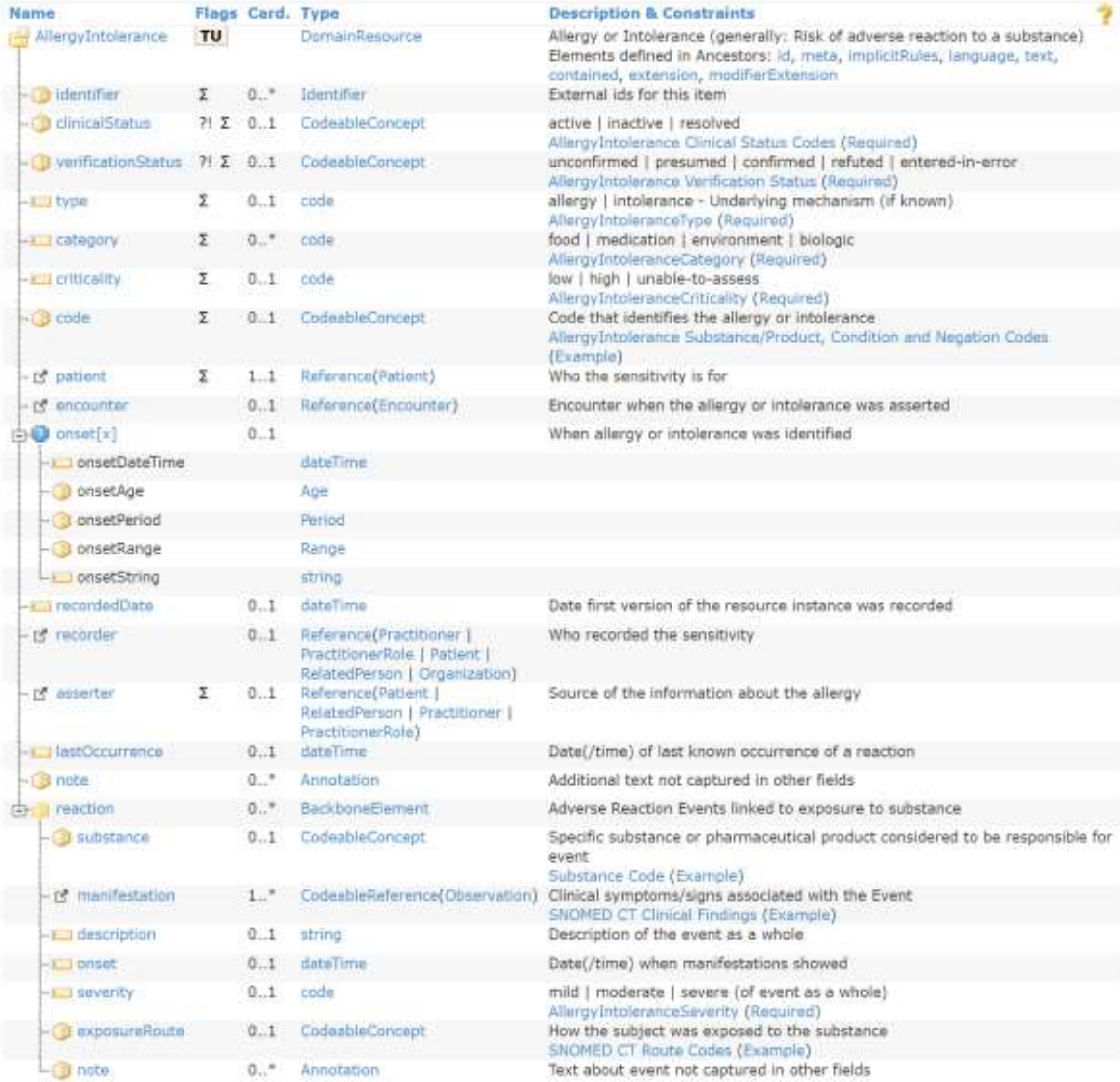
  - More tricks on https://build.fhir.org/ig/HL7/fhir-shorthand/reference.html
    - You can test on FSH

4. Add a profile

- One approach to follow is to ensure that all elements mentioned on the Logical Model are reflected in the profile
  - If element is in there, mark it as “Must Support”
  - If element is not in there, add extension

- If element needs changes, add constraints
  - If constraints are not possible, we can’t change it – just add another element

4. Add a profile – compare with base resource



Name	Flags	Card.	Type	Description & Constraints
AllergyIntolerance	TU		DomainResource	Allergy or Intolerance (generally: Risk of adverse reaction to a substance) Elements defined in Ancestors: id, meta, implicitRules, language, text, contained, extension, modifierExtension
identifier	Σ	0..*	Identifier	External ids for this item
clinicalStatus	?! Σ	0..1	CodeableConcept	active | inactive | resolved AllergyIntolerance Clinical Status Codes (Required)
verificationStatus	?! Σ	0..1	CodeableConcept	unconfirmed | presumed | confirmed | refuted | entered-in-error AllergyIntolerance Verification Status (Required)
type	Σ	0..1	code	allergy | intolerance - Underlying mechanism (if known) AllergyIntoleranceType (Required)
category	Σ	0..*	code	food | medication | environment | biologic AllergyIntoleranceCategory (Required)
criticality	Σ	0..1	code	low | high | unable-to-assess AllergyIntoleranceCriticality (Required)
code	Σ	0..1	CodeableConcept	Code that identifies the allergy or intolerance AllergyIntolerance Substance/Product, Condition and Negation Codes (Example)
patient	Σ	1..1	Reference(Patient)	Who the sensitivity is for
encounter		0..1	Reference(Encounter)	Encounter when the allergy or intolerance was asserted
onset[x]		0..1		When allergy or intolerance was identified
onsetDateTime			dateTime	
onsetAge			Age	
onsetPeriod			Period	
onsetRange			Range	
onsetString			string	
recordedDate		0..1	dateTime	Date first version of the resource instance was recorded
recorder		0..1	Reference(Practitioner | PractitionerRole | Patient | RelatedPerson | Organization)	Who recorded the sensitivity
asserter	Σ	0..1	Reference(Patient | RelatedPerson | Practitioner | PractitionerRole)	Source of the information about the allergy
lastOccurrence		0..1	dateTime	Date(/time) of last known occurrence of a reaction
note		0..*	Annotation	Additional text not captured in other fields
reaction		0..*	BackboneElement	Adverse Reaction Events linked to exposure to substance
substance		0..1	CodeableConcept	Specific substance or pharmaceutical product considered to be responsible for event Substance Code (Example)
manifestation		1..*	CodeableReference(Observation)	Clinical symptoms/signs associated with the Event SNOMED CT Clinical Findings (Example)
description		0..1	string	Description of the event as a whole
onset		0..1	dateTime	Date(/time) when manifestations showed
severity		0..1	code	mild | moderate | severe (of event as a whole) AllergyIntoleranceSeverity (Required)
exposureRoute		0..1	CodeableConcept	How the subject was exposed to the substance SNOMED CT Route Codes (Example)
note		0..*	Annotation	Text about event not captured in other fields

- patient - MS
- clinicalStatus - Mandatory
- verificationStatus - Mandatory
- code – Mandatory
- recordedDate MS
- recorder MS
- asserter MS
- reaction MS
  - manifestation MS
  - Certitude – Need extension
  - exposureRoute - MS
  - note (if exists) - MS

https://fshschool.org/FSHOnline/#/share/3zGvHU6

5. Add a ValueSet and binding

- ValueSets are for the coded elements
- Choose your strength: this binding should not be required, but extensible or preferred
- We'll use 12 allergens: Peanuts, tree nuts, milk, eggs, fish, shellfish, soy, sesame seeds, mustard, celery, lupin
- Create and build

https://fshschool.org/FSHOnline/#/share/2UQuaMG

6. Add an example

- Examples can also be defined in sushi / shorthand

- Create an instance for a suspected allergy to peanut

https://fshschool.org/FSHOnline/#/share/3eX1sAo

7. Check the QA report

- The QA report shows errors. If you want to deploy this ImplementationGuide, these errors must be fixed.

9. Use it

- In a validator
- Deploy a server

Additional notes

More ImplementationGuide content

- These techniques can be used to document more specifications
  - Defining content exchange aggregates with Bundles, Composition, MessageHeader..
  - Defining Operations and Search Parameters
  - Defining other requirements in narrative format

Guidance

- Check out the IG registry: http://fhir.org/guides/registry/
- FHIR Sample IG: https://github.com/FHIR/sample-ig
  - Example content, standard techniques
- FHIR Guidance IG: http://build.fhir.org/ig/FHIR/ig-guidance
  - Changing colors, adding features
- Always check chat.fhir.org

Final Questions and Answers

- Has this answered **your** questions?
- How do **you** expect to use ImplementationGuides?

Get in touch, be active

- Check with others (at chat.fhir.org or community.fhir.org)
- Create (or ask someone to create) a change request
- Join a FHIR® event like DevDays (devdays.com), discuss
- Join a FHIR® connectathon, test and provide feedback

Digital Square is supported by:

USAID
FROM THE AMERICAN PEOPLE

BILL & MELINDA
GATES foundation

PATH

Digital Square is a PATH-led initiative funded and designed by the United States Agency for International Development, the Bill & Melinda Gates Foundation, and a consortium of other donors.

This presentation was made possible by the generous support of the American people through the United States Agency for International Development. The contents are the responsibility of PATH and do not necessarily reflect the views of USAID or the United States Government.