

```
"resourceType" : "Patient"
  "text" : {
    "status" : "generated"
    "_status" : {
      "id" : "12344"
    },
    "div" : "<div"
  },
  "identifier" : [
    {
      "use" : "tv"
    }
  ]
}
```

# FHIR FUNDAMENTALS COURSE

## FAST HEALTHCARE INTEROPERABILITY RESOURCES



**FHIR Course, Unit 4**  
**FHIR Conformance and Profiles**  
**Reading Material**

## Course Overview

### ***Module I: Introduction***

***Introduction to FHIR***

***Resources***

### ***Module II: Work with FHIR***

***RESTful FHIR***

***Searching with FHIR***

### ***Module III: FHIR advanced***

***Transactions***

***Paradigms***

***Messaging***

***Documents and CDA R2***

***Operations***

### ***Module IV: FHIR Conformance***

***Conformance Resources***

***Extensions***

***Profiles***

***Implementation***

---

## Table of Contents

Table of Contents.....	3
Unit Content and Learning Objectives.....	5
Why do we need profiles? .....	6
What can be constrained? .....	8
How do we document the constraints?.....	9
FHIR Conformance Resources.....	10
Operations .....	11
CapabilityStatement .....	11
OperationDefinition .....	11
SearchParameter.....	14
Content.....	15
StructureDefinition .....	15
DataElement .....	18
FHIR Extensions.....	18
Complex Extensions .....	20
Modifier Extensions .....	20
Binding .....	22
Terminology .....	23
CodeSystem .....	24
ValueSet .....	25
Valueset Operations: \$expand and \$validate .....	26
ConceptMap .....	27
NamingSystem .....	27
Publishing .....	28
ImplementationGuide.....	28
Processing the ImplementationGuide .....	29
Resource Validation against a Profile.....	31
Conformance Declaration in Resource Instances.....	31
Resource Validation .....	31
Tooling to Create Profiles.....	32
Forge .....	33
Trifolia .....	34
Trifolia on FHIR .....	34
Simplifier.....	<b>Error! Bookmark not defined.</b>
clinFHIR .....	35
Snapper.....	36
Some Important Profiles/Implementation Guides .....	37
Argonaut Project.....	37
IHE-on-FHIR.....	38
Smart-On-FHIR .....	40

FHIR Official Implementation Guides.....	41
Public Regional Implementation Guides .....	41
More Information about FHIR Profiling .....	<b>Error! Bookmark not defined.</b>
This week's assignment .....	42
Unit Summary and Conclusion .....	43
Additional Reading Material.....	44
Information about FHIR .....	44

## Unit Content and Learning Objectives

This Unit discusses some advanced topics regarding FHIR profiling: how to define extensions and restrictions on FHIR resources, and how to document these restrictions using FHIR resources.

This unit complements what you've learned in last week's unit: you can combine any resource, and extend FHIR freely, but you also need to **document your choices**: let the world (or at least, your partners in exchange, current and future), **what each application is expected to send and receive**, and **how to validate the resources** before going to (or even while in) production.

The good news is that the documentation format is in the form of FHIR resources.

**Note:** This unit is loosely based on Graham Grieve's 'Profiling Concepts' class, taught during FHIR DevDays, Michael Rutten's 'FHIR for Specifiers' tutorial, and the unit authors' interpretations and experience.

Conformance resources were augmented and revamped for STU3, and they constitute a good part of what became normative in R4 (FHIR Maturity Level 5).

Our goal for this unit **IS NOT** for you to become an expert in FHIR Conformance and Profiling but at least to get you acquainted with the involved resources and tooling, and their expected use, and also provide some examples of their use.

Also we want to bring into your attention some important sources of implementation guides.

## Why do we need profiles?

FHIR is an international specification, used in multiple and diverse contexts. As you may have already noticed, each server can implement a subset of the specification in terms of capabilities, content, format and transport mechanisms.

In fact, there are a lot more disagreements than agreements. Laws, regulations, cultures, requirements, tools and population differences may change the goals for the systems and for the information exchange.

The same concept can be implemented in different systems using different components, attributes, etc.

Example, as seen in Figure 1

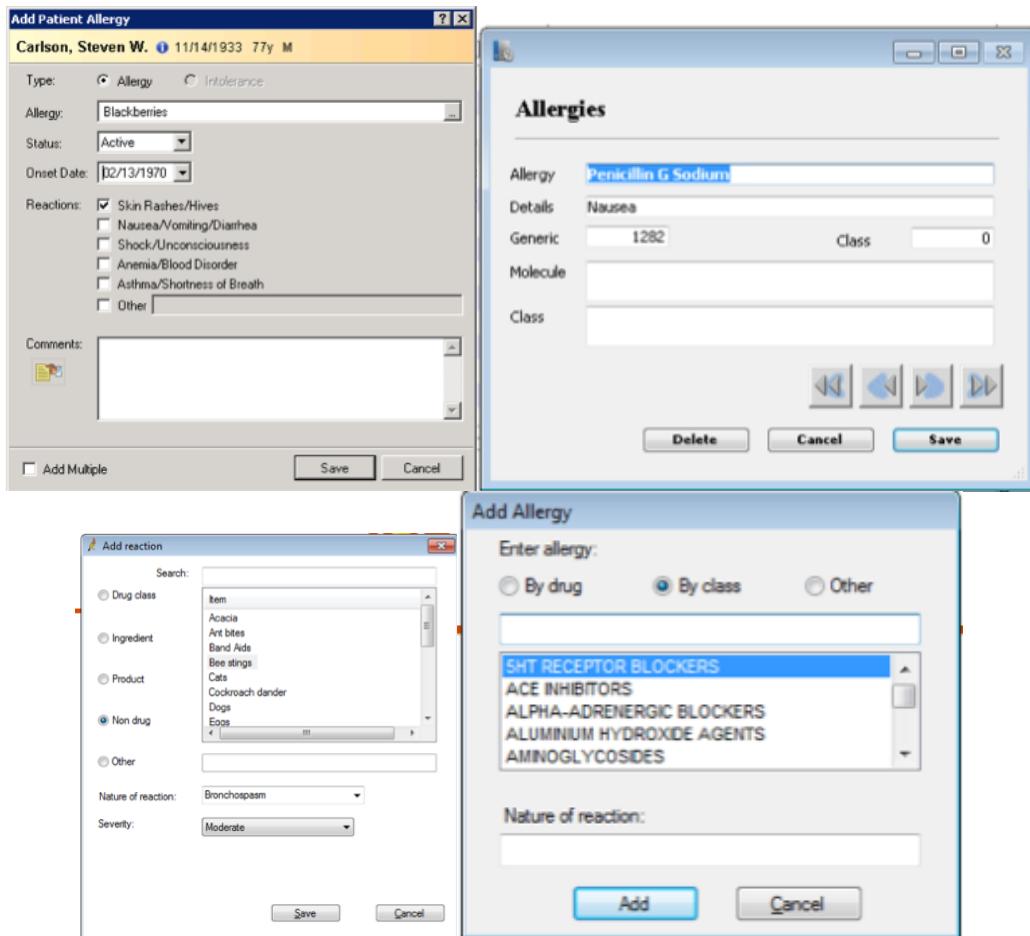
System 1: Allergy: Description (coded) + Nature of Reaction

System 2: Allergy: Description (coded) + Text + Nature of Reaction+ Severity

System 3: Allergy: Description (coded) + Text + Generic + Class + Molecule

System 4: Allergy: Description (coded) + Comments + Status + Onset Date+ Reactions (coded)

**FIGURE 1 – DIFFERENT SYSTEM APPROACHES TO THE SAME CONCEPT**



So FHIR needs to be implemented regardless of these differences, but also expressing in some way the choices and the specific implementation in each site or region.

For a long time, profiles in HL7 were developed in a different syntax than the standard itself (Schematron, huge Word files, special XML files and Schemas, etc.). And expressing profiles was always an afterthought.

FHIR brings a new view to conformance and validation: profiles, conformance statements, and implementation guides can be created using FHIR resources, and a FHIR server can validate instances of resources against these profiles as one of the defined operations.

So, FHIR introduces built-in profiling: a profile is defined and verified using the same standard and by the same servers.

We can see FHIR as a ‘platform spec’: we will always need additional agreements. The only difference with previous HL7 specs is that these agreements are computable, even ‘discoverable’: you can ask a server what services it can provide, and act accordingly.

Why do we create profiles?

- 1) Communicate our colleagues what was decided or expected.
- 2) Allow automatic comparison and verification
- 3) Support code generation / run-time discovery
- 4) Create a publish/subscribe ecosystem
- 5) Allow conformance validation

**A FHIR server becomes also a FHIR extension/profile and implementation guide registry and resource validation tool.**

More info: you can follow a free profiling tutorial created by Mitre here:  
<https://fshschool.org/courses/fsh-seminar/01-reading-an-ig.html>

## What can be constrained?

Mostly all components for a FHIR server can be constrained:

The basic questions are “What is used?”, “What is NOT used”, “What is extended?”

And the basic premise is **“We cannot overrule the specification”**:

- 6) if something is mandatory in FHIR, we cannot deem it optional or not used.
- 7) If a code is bound to a required vocabulary in FHIR, we cannot change the vocabulary

A short list of items we can constrain:

**Elements in resources:**

Which **optional elements are mandatory for us?**

Which **elements need to be repeated** in our implementation? How can we **differentiate the repetitions?**

**Codes in Coded Elements:**

Which **terminologies** are used? **Applied to which coded elements?**

**Behavior:**

Which **specific RESTful API interactions** do our servers support for each resource they manage?

**Do we allow deletion** on resources? Do we support **versioning?**

Special operations: definition of special NON-Crud operation and its parameters

**Searches:**

Which search parameters do our servers support?

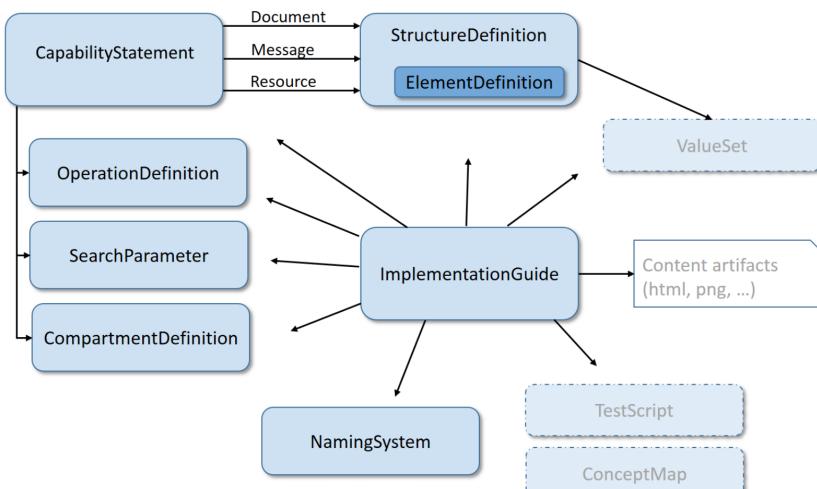
**Extensions:**

Which specific extensions do we support or require? Applied to which resources /elements / data types?

## How do we document the constraints?

These constraints and their derived usage statements need to be

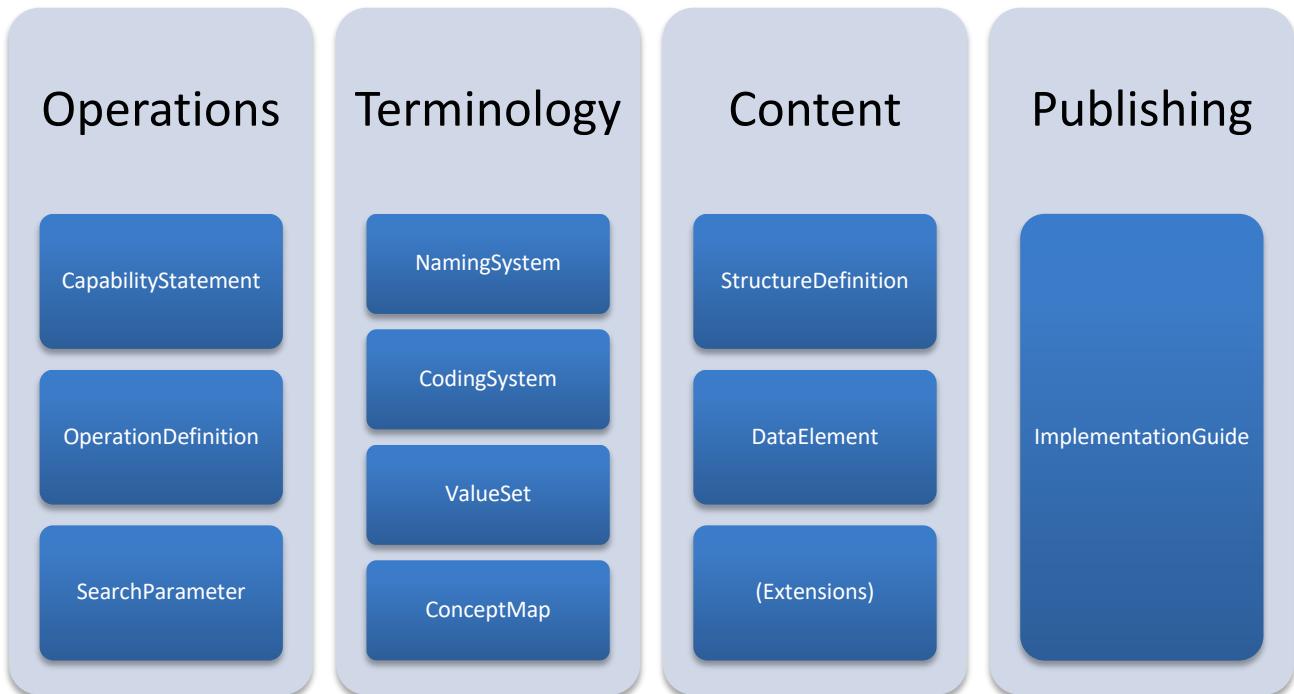
- 1) Authored in a structured manner
- 2) Published in a repository
- 3) Used as a basis for validation, code, and UI generation



## FHIR Conformance Resources

HL7 choose FHIR resources to be used for the expression of these constraints.

Let's explore the use of each resource and their combinations, dividing them into four areas:



As you may already have thought, the **ImplementationGuide** resource groups all the other resources in a package for publishing and make them available for the implementers.

Note: There are more conformance resources (MessageDefinition, GraphDefinition, Compart-  
mentDefinition, etc.) but we will focus in this subset because it includes those more widely used.

## Operations

This group of resources is about the server BEHAVIOUR, what the server can and cannot do, and how is this implemented.

### CapabilityStatement

This is the most important resource in this space: it allows to see what specific API calls the server will accept, and over which resource types

Maybe it's the first thing to do when implementing or confronting (as a FHIR client) a server.

You can see a partial rendering of a CapabilityStatement in FIGURE 2.

You can also seek this course's server (or any other) capability statement by pasting this into your Internet browser or REST Client (it may take a while, be patient):

<http://fhirserver.hl7fundamentals.org/fhir/metadata>

**FIGURE 2 – PARTIAL RENDERING OF A FHIR SERVER CONFORMANCE STATEMENT**

Mode	SERVER									
Description	All the functionality defined in FHIR									
Transaction	y									
System History	y									
System Search	y									
Resource Type	Profile	Read	V-Read	Search	Update	Creates	Deletes	History		
Account	<a href="http://hl7.org/fhir/StructureDefinition/Account">http://hl7.org/fhir/StructureDefinition/Account</a>	y	y	y	y	y	y	y	y	y
AllergyIntolerance	<a href="http://hl7.org/fhir/StructureDefinition/AllergyIntolerance">http://hl7.org/fhir/StructureDefinition/AllergyIntolerance</a>	y	y	y	y	y	y	y	y	y
Appointment	<a href="http://hl7.org/fhir/StructureDefinition/Appointment">http://hl7.org/fhir/StructureDefinition/Appointment</a>	y	y	y	y	y	y	y	y	y
AppointmentResponse	<a href="http://hl7.org/fhir/StructureDefinition/AppointmentResponse">http://hl7.org/fhir/StructureDefinition/AppointmentResponse</a>	y	y	y	y	y	y	y	y	y
AuditEvent	<a href="http://hl7.org/fhir/StructureDefinition/AuditEvent">http://hl7.org/fhir/StructureDefinition/AuditEvent</a>	y	y	y	y	y	y	y	y	y
Basic	<a href="http://hl7.org/fhir/StructureDefinition/Basic">http://hl7.org/fhir/StructureDefinition/Basic</a>	y	y	y	y	y	y	y	y	y
Binary	<a href="http://hl7.org/fhir/StructureDefinition/Binary">http://hl7.org/fhir/StructureDefinition/Binary</a>	y	y	y	y	y	y	y	y	y

Information included in the capability statement:

**Server wide:** Software name and version, special operations supported, and their parameters, FHIR Version, Supported Syntaxes: JSON, XML, other.

**Per resource class:** CRUD Interactions supported, Conditional Create/Update/Delete for each resource type, Search inclusions, Search parameters

### OperationDefinition

The RESTful API defines a set of common interactions (read, update, search, etc.) performed on a repository of typed resources. This approach solves many cases, but sometimes it is better to use an RPC-like paradigm, complementing to the regular CRUD operations.

In this paradigm there are **NAMED** operations, performed with input parameters (resources, data). Named operations are invoked in FHIR servers using the dollar sign (**fhirserver.com/base/\$operation\_name**)

Operations may be performed on a **specific resource instance**, a **resource type**, or a **whole system**.

The OperationDefinition resource allows to provide a readable and computable definition of the custom (non-CRUDS) server defined operation.

Each Operation is defined by:

**Context:** system, resource type, or resource instance

**Name:** A name for the Operation

**Idempotent:** The operation does not change underlying resources, so it can be used through a GET

**Parameter List**, with this information for each parameter:

**Name:** the name of the parameter.

**Use:** In | Out | Both

**Type:** a data type or a Resource type

**Profile:** a StructureDefinition that applies additional restrictions about the resource

**Documentation** - a description of the parameter's use

You can see two examples of “Operation Definition” below

- 1) Query for all the ‘risk’ patients filtered optionally (by ward). This operation on the Patient resource has a ‘ward’ in parameter with the filters and returns a Bundle with Patient resources. The algorithm applied to define which patients are at risk is known by the server.

```
{
  "resourceType" : "OperationDefinition",
  "id" : "example-query-high-risk",
  "text" : {
    "status" : "generated",
    "div" : "<div xmlns=\"http://www.w3.org/1999/xhtml\">redacted</div>"
  },
  "url" : "http://hl7.org/fhir/OperationDefinition/example-query-high-risk",
  "version" : "0.0.1",
  "name" : "ExampleOfHighRiskPatientQuery",
  "title" : "Example of High-Risk Patient Query",
  "status" : "draft",
  "kind" : "query",
  "description" : "Example Named Query of Patients based on risk assessments",
  "code" : "example-query-high-risk",
  "resource" : ["Patient"],
  "system" : false,
  "type" : true,
  "instance" : false,
  "parameter" : [
    {
      "name" : "ward",
      "use" : "in",
      "min" : 0,
      "max" : "*",
      "documentation" : "Ward filters to apply to patient locations",
      "type" : "string",
      "searchType" : "reference"
    }
  ]
}
```

```

    "name" : "result",
    "use" : "out",
    "min" : 1,
    "max" : "1",
    "documentation" : "Searchset bundle",
    "type" : "Bundle"
  }
},
},

```

- 2) This is a very small JSON representation of the \$ihe\_pix operation defined by the IHE PIX Mobile Implementation Guide

```

"resource":{
  "resourceType":"OperationDefinition",
  "name": "$ihe-pix",
  "status": "draft",
  "type": [
    "Patient"
  ],
  "parameter": [
    {
      "name": "sourceIdentifier",
      "use": "in"
    },
    {
      "name": "targetSystem",
      "use": "in"
    },
    {
      "name": "_format",
      "use": "in"
    }
  ]
},

```

GET [base]/Patient/\$ihe-pix?sourceIdentifier=[token]]{&targetSystem=[uri]}{&\_format=[mime-type]}

Parameter 1: sourceIdentifier (1..1) Token

The Patient identifier search parameter that will be used by the Patient Identifier Cross-reference Manager to find cross matching identifiers associated with the Patient resource.

Parameter 2: targetSystem 0..1 uri

The target Patient Identifier Assigning Authority from which the returned identifiers should be selected

Parameter 3: \_format 0..1 mime-type The requested format of the response

Example Call:

[base]/Patient/\$ihe-pix?sourceIdentifier=urn:oid:1.3.6.1.4.1.21367.2010.1.2.300|NA5404

More details here: <https://profiles.ihe.net/ITI/PIXm/ITI-83.html>

You can see a complete description of a FHIR operation here

<https://www.hl7.org/fhir/operation-patient-match.html>

## SearchParameter

All search parameters supported by standard FHIR resources are defined using the SearchParameter resource.

It includes all the information needed about the way this specific resource class can be searched inside of a FHIR server:

- code (how the parameter is named in actual searches)
- type (number, string, token, etc.)
- target (which resource is the parameter applied to)
- xpath or fhir path (where in the resource is this content related to this parameter located)

Note: Wondering what a ‘**fhir path**’ is? FHIR needed an equivalent to XPath for JSON, or better yet, something that could logically refer to the content of a resource regardless of the syntax. Hence, ‘FHIRPath’. If you want to learn more about FHIRPath, please read this: <http://hl7.org/fhirpath/>. This resource is also used if you want to support searches not defined by the core spec itself. To do this, you need to create a new SearchParameter for your resource, and include it both in the server conformance statement and the implementation guide for your project.

Here you can find an example of a special SearchParameter defined by HL7 (in this case is ‘searching for an extension)

<http://hl7.org/fhir/searchparameter-example-extension.html>

## Content

### StructureDefinition

#### Overview

StructureDefinition allows creation of basic types and resources, i.e.: all FHIR resources are expressed as StructureDefinition resources.

Basically it is a **header** with information about the structure itself: FHIR version, publisher, base resource, mappings, and context, plus a **list of data element** definitions.

#### Use

StructureDefinition is used to define profiles/constraints on resources:

- 1) Limit cardinality
- 2) Change value set mappings
- 3) Add additional constraints
- 4) Add extensions
- 5) Define external or logical mapping

**Remember the basic principle:** While doing all this, you cannot contradict the standard's basic definitions: mandatory elements and coding, etc. A profiled element is still an element of the same type.

### Structure Definition Profiles

The list of a specific data element definitions, is in fact two possible lists:

**Differential:** a list of elements that changed from the original definition. This reflects the human vision of the profiled resource. It's sparse (only contains the minimum information needed)

**Snapshot:** it's the result to apply the 'differential' to the base StructureDefinition in which this particular definition is based. Lists everything that is allowed in the resource. Used mainly by the tooling

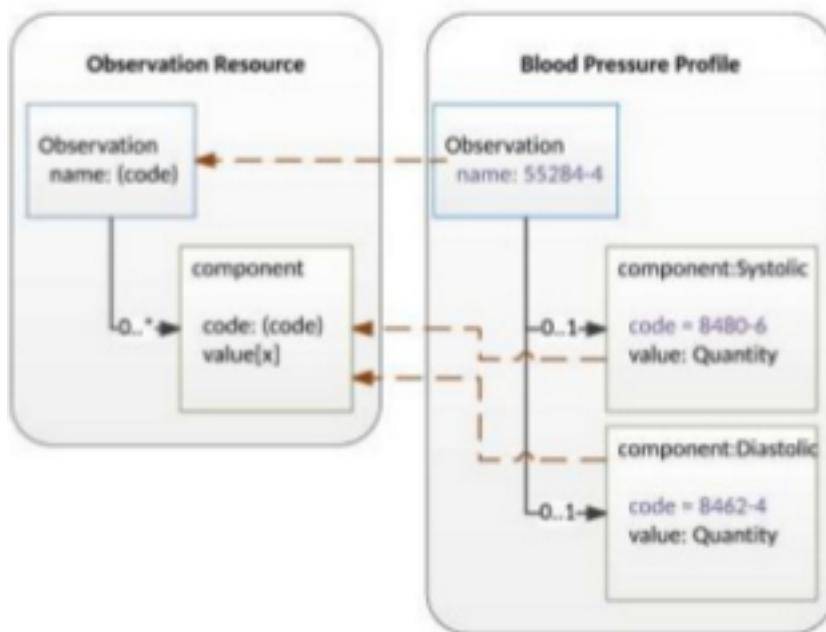
## Slicing and Discriminators

Any resource allowing more than one repetition can be profiled using slicing.

Why do we need to profile the repetitions? Some use cases follow:

- 1) Exactly two patient identifiers are needed: national patient id, and local id
- 2) Exactly two observations with specific codes for each one. Example: Blood Pressure Profile: Systolic BP + Diastolic BP, as shown in figure 3. We don't need 'any' Observation resource, we need two specific observations, identified by their code/value

**FIGURE 3 – EXAMPLE OF SLICING – TWO SPECIFIC OBSERVATIONS NEEDED AS COMPONENTS.**



## Example of slicing

In a small non-real country, Berzerkistan, the MOH (Ministry of Health) needs to receive for their Healthcare Unified Repository a bundle with patient, organization, practitioner, encounter and condition resources whenever a specific disease ('reportable disease') is diagnosed.

Patients in Berzerkistan are identified by the BNI (Berzerkistan National Identifier), but since there were some problems (repetitions, wrong assignments, etc.), the MOH has also created a National Healthcare Identifier (NHIB). For the patient resource, both identifiers are mandatory.

*Note: This country does not exist, it is a completely fictional country created only for this Unit's Reading Material and Assignment. It is also known as "Unit4FhirFunCourseLand". Thus the site for the country's health ministry site is also fictional.*

*Regardless of this, the complete Implementation Guide can be found here:*

<http://courses.hl7fundamentals.org/profile-berze/site/index.html>

This is how the slice looks (note how we 'sliced' the identifier element so now there are two identifier elements needed). In this case the slice is based in the identifier.system attribute.

The first one of them needs to be valued as <http://www.berzerkistan.gov/bni>

The second one of them needs to be valued as <http://www.berzerkistan.gov/nhib>

 <b>Slices for identifier</b>	2..2	<b>Identifier</b>	An identifier for this patient <b>Slice:</b> Unordered, Open by value:use
 <b>identifier:BniIdentifier</b>	1..1	<b>Identifier</b>	An identifier for this patient
 <b>use</b>	1..1	<b>code</b>	usual   official   temp   secondary   old (If known) <b>Required Pattern:</b> official
 <b>system</b>	1..1	<b>uri</b>	The namespace for the identifier value <b>Required Pattern:</b> <a href="http://www.berzerkistan.gov/bni">http://www.berzerkistan.gov/bni</a>
 <b>value</b>	1..1	<b>string</b>	The value that is unique
 <b>identifier:NHibIdentifier</b>	1..1	<b>Identifier</b>	An identifier for this patient
 <b>use</b>	1..1	<b>code</b>	usual   official   temp   secondary   old (If known) <b>Required Pattern:</b> usual
 <b>system</b>	1..1	<b>uri</b>	The namespace for the identifier value <b>Required Pattern:</b> <a href="http://www.berzerkistan.gov/nhib">http://www.berzerkistan.gov/nhib</a>
 <b>value</b>	1..1	<b>string</b>	The value that is unique

And this is how a valid instance looks like (the Patient.identifier part)

```
"identifier" : [
    {
        "use" : "official",
        "system" : "http://www.berzerkistan.gov/bni",
        "value" : "9000031"
    },
    {
        "use" : "usual",
        "system" : "http://www.berzerkistan.gov/nhib",
        "value" : "23327758999"
    }
]
```

## DataElement

Data elements describe the characteristics of a piece of data - identifier, name, definition, data type, length, permitted value set, usage notes, etc. They are used inside of StructureDefinition as its building blocks.

## FHIR Extensions

In our first unit, you've learnt how to include an extension in your FHIR resource instance. Remember: it is composed by an url and a value, the url pointing to the extension definition.

Here you will learn how to DEFINE extensions.

**An extension is also defined using a StructureDefinition resource.**

### Simple Extension

A simple extension is an extension as we have seen until now: single valued.

Let's see an example of an extension.

In the same imaginary small country, Berzerkistan, where the State imposes taxes on private provided healthcare (PPH), people with no coverage ('uncovered') pays 21% PPH tax on any privately provided healthcare service, people who choose its insurance company ('voluntary') pays 10.5% PPH tax, and people who uses the Unions' healthcare services ('mandated') doesn't pay PPH tax at all (this was decided after the Unions led a massive national strike that lasted 6 days in November 2010)

This is Berzerkistan PPH PatientTaxSituation table

PPH Code	PPH Description	PPH Tax
U	Uncovered	21%
V	Voluntary	10.5%
M	Mandated	0%

Each patient has an assigned tax category, called "PatientTaxSituation". It is a coded element. Since there is no such item in the Patient resource as defined by the HL7 FHIR standard, the Government Agency in charge decided to create a FHIR extension.

You can see this definition here <http://courses.hl7fundamentals.org/profile-berze/site/Structure-Definition-PPHTaxSituation.html>

Some highlights of the extension definition:

**Metadata about the extension:** name, status, description, requirements, fhirVersion

**constrainedType:** type of element, in this case, we are constraining Extension, i.e. defining a new extension.

**contextType:** where is this element used: in a resource, in a datatype?. In this case: resource.

**context:** which resource this extension applies to?. In our case: **patient**.

Finally, the definition includes the differential for the extension. Simple extensions have two elements: url and value[x]. Any url, and a choice of values (quantity, string, etc.)

This specific extension (thus the ‘differential’) has

1. A fixed url: <http://minhealth.gov.bk/core/StructureDefinition/PPHTaxSituation>

2. A specific value type: valueCode.

Maybe you think that something is missing in this explanation...?

**“Why does the Berzerkistan government charges tax on healthcare services?”**

That wasn’t the question, although is a good one. But we cannot answer that.

**The question is “Where is the PPH code table? How do we know that we are using the correct code for the patient tax situation in our instance?”**

There is also a part of the StructureDefinition where we can include this, and a special resource to refer to this table, called a **ValueSet**.

The part of the extension (but this apply to any coded element constrained by the FHIR spec or a specific profile) defining where to find the value set to validate the codes is called **Binding**.

The ValueSet example (the table of valid codes represented as a FHIR resource) extracting the values from the original code system is here:

<http://courses.hl7fundamentals.org/profile-berze/site/ValueSet-PPHTaxSituationVS.html>

You can find more information in the next section (Terminology) about ValueSets and CodeSystems and how to build them.

## Complex Extensions

Complex extensions are defined by an url and a list of elements.

This is an example from US Core (no value, and several parts – they are called ‘local extensions’)

Name	Flags	Card.	Type	Description & Constraints
Extension		0..1	Extension	US Core ethnicity Extension
extension:ombCategory	S	0..1	Extension	Hispanic or Latino Not Hispanic or Latino
url		1..1	uri	"ombCategory"
value[x]		1..1	Coding	Value of extension
extension:detailed		0..*	Extension	Extended ethnicity codes
url		1..1	uri	"detailed"
value[x]		1..1	Coding	Value of extension
extension:text	S	1..1	Extension	Binding: Detailed ethnicity (required)
url		1..1	uri	ethnicity Text
value[x]		1..1	string	Value of extension
url		1..1	uri	"text"
value[x]		0..0		"http://hl7.org/fhir/us/core/StructureDefinition/us-core-ethnicity"

In fact, complex extensions can be the base of completely new resources, if no resource in FHIR suits your needs, this can be achieved using something called the ‘Basic’ resource. You can extend a Basic resource with all the extensions you need (See <https://www.hl7.org/fhir/basic.html> and create your own FHIR resources.

An example of this is here: <http://hl7.org/fhir/R4B/basic-example.xml.html>

Although we cannot do more than tell you that doing so will do little good for overall global interoperability. Please do not. Refrain from doing it. Please!

## Modifier Extensions

Modifier extensions are extensions that exchange partners in a specific implementation need to understand and use, because the extension CHANGES the meaning of the resource.

Examples of this are: recording an instruction NOT to take a medication, using the condition resource to transmit the fact that the specific condition belongs to another person, not the patient itself, asserting that a performer WAS NOT involved in a procedure.

So, if your extension modifies the content of the resource and the resource CANNOT BE PROCESSED SAFELY by applications IGNORING IT, then your extensions deserves to be labelled Modifier Extension.

Declaration as ‘Modifier Extension’ is simple: just include isModifier=True in the structure definition.

Implementation of modifier extensions is more difficult, because you will need to add that information to your system and throughout the pipeline from the system to the exchange partners and back.

Modifier extensions are included using the modifierExtension element instead of the Extension element.



## Binding

**Binding** is the relationship between a coded element and the value set for its validation.

It includes ‘**strength**’ (how strict is code validation?) and ‘**valueSetReference**’ (where is the list of codes can we use?).

Options for **binding strength** are

“**required**”: no other codes can be used in a valid instance

“**example**”: these codes are just an example

“**preferred**”: we prefer this set of codes, but others can be used

“**extensible**”: you can add your own codes to this list

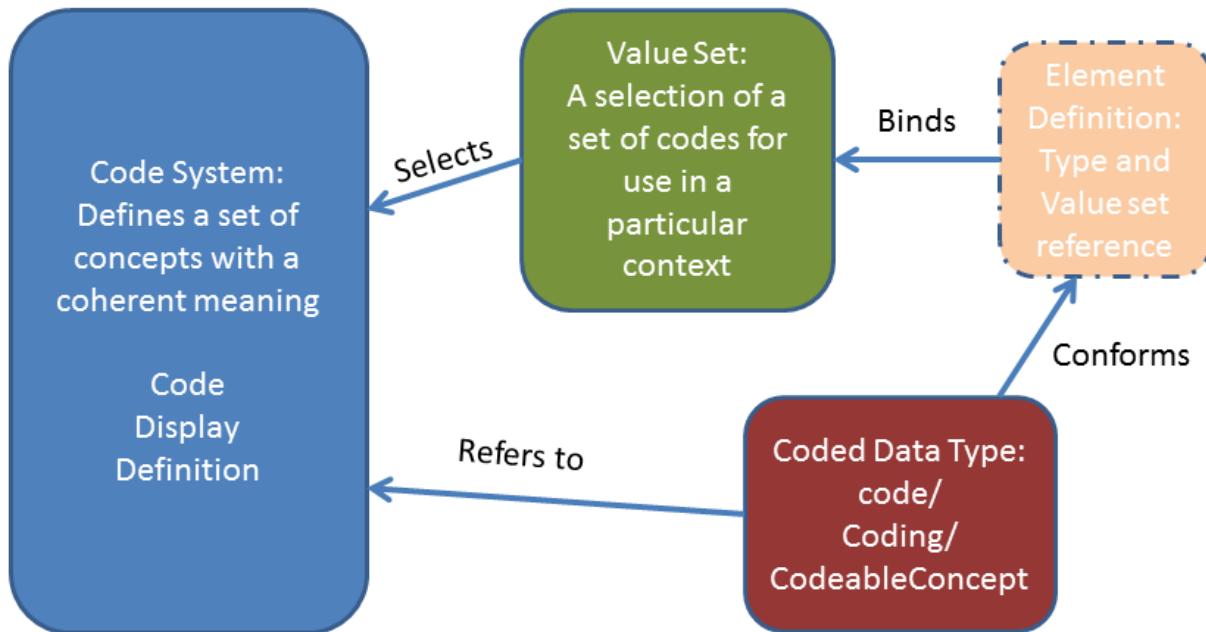
This is an example from our IG: codes are to be drawn from a specific valueset, and you cannot use any other codes.

PPHTaxSituation	1..1	code	Extension URL: <a href="http://minhealth.gov.bk/core/StructureDefinition/PPHTaxSituation">http://minhealth.gov.bk/core/StructureDefinition/PPHTaxSituation</a> Binding: Set of Valid Codes for PPH Tax Situation (required)
-----------------	------	------	---

## Terminology

These are terminology/vocabulary related resources. The most important for Profile implementation is ValueSet: you can store your constrained ValueSet in a FHIR Server, and refer the vocabulary constrains in your profile to the server.

Here are all the terminology resources, and its relationships.



The difference this has with previous implementation of vocabulary with HL7 Standards is that references to vocabulary can ‘resolve’, so you can search the vocabulary in the server that contains it. It’s not just a reference to a comma-separated file you must download and/or import into your system.

CodeSystems, ValueSets and ConceptMaps in FHIR have support for special operations, intended to create Terminology Services

Note: Do you want to understand what a terminology service does? Please read  
<https://www.hl7.org/fhir/terminology-service.html>

## CodeSystem

Code Systems define a set of codes with meaning, which codes exists and how are they understood. Code Systems are also known as Terminologies, Classifications, Enumerations or Ontologies.

The code system resource has an identifying URL (it never changes no matter where the resource is stored physically), version and another metadata.

This resource is not intended for Code System maintenance, but to publish the metadata and (optionally) the content (set of codes) to support expansion and validation. Large code systems (SNOMED CT, LOINC ICDx) have their own publishing and distribution methods.

Externally published code systems have a special URL defined for its use in FHIR artifacts. You can find most of them here <https://www.hl7.org/fhir/terminologies-systems.html>

Publishing as CodeSystem resources is intended for smaller ad-hoc code systems.

This is an example of a Code System:

In a small non-real country, Berzerkistan, the MOH (Ministry of Health) dictated that all organizations and practitioners charge their patients a percentage of the fees for the services based on a classification created by the MOH, called ‘Patient Tax Situation’.

You can find a simple CodeSystem definition for the Ministry of Health of Berzerkistan, which applies a tax for healthcare here:

<http://courses.hl7fundamentals.org/profile-berze/site/CodeSystem-PPHTaxSituationCS.html>

This is the simplest terminology operation for a CodeSystem: **\$lookup** (it is not just a code search; it may imply retrieving and returning the concept information from the underlying code system information)

[basefhir]\$/lookup?system=[system]&code=[code]

You can try it, searching for code “M” in this CodeSystem (Patient Tax Situation for BK)

[http://fhirserver.hl7fundamentals.org/fhir/CodeSystem/\\$lookup?system=http://min-health.gov.bk/core/CodeSystem/PPHTaxSituationCS&code=M](http://fhirserver.hl7fundamentals.org/fhir/CodeSystem/$lookup?system=http://min-health.gov.bk/core/CodeSystem/PPHTaxSituationCS&code=M)

## ValueSet

Value Sets are the way to express exactly which values are accepted for system/value of a coded element. In your implementation guide or profile, you can bind each coded element to a specific value set. You can build a value set by:

- enumerating the specific concepts from one or more code systems

Example:

<https://www.hl7.org/FHIR/valueset-doc-section-codes.html>

### 4.3.1.232.1 Content Logical Definition

This value set includes codes from the following code systems:

- Include these codes as defined in <http://loinc.org>
- | Code    | Display   |
|---------|---|
| 10154-3 | Chief complaint Narrative - Reported                      |
| 10157-6 | History of family member diseases Narrative               |
| 10160-0 | History of medication use Narrative                       |
| 10164-2 | History of present illness Narrative                      |
| 10183-2 | Hospital discharge medications Narrative                  |
| 10184-0 | Hospital discharge physical findings Narrative            |
| 10187-3 | Review of systems Narrative - Reported                    |
| 10210-3 | Physical findings of General status Narrative             |
| 10216-0 | Surgical operation note fluids Narrative                  |
| 10218-6 | Surgical operation note postoperative diagnosis Narrative |
| 10218-6 | Surgical operation note postoperative diagnosis Narrative |
| 10223-6 | Surgical operation note surgical procedure Narrative      |

- defining some criteria by which the concepts are extracted from the original code system. Example: <https://www.hl7.org/FHIR/valueset-activity-reason.html>. This valueset extracts “all codes from SNOMED CT where concept is-a 404684003 (Clinical finding)”.

### 4.3.1.137.1 Content Logical Definition

This value set includes codes from the following code systems:

- Include codes from <http://snomed.info/sct> where concept is-a 404684003 (Clinical finding)

### 4.3.1.137.2 Expansion

This expansion generated 19 Apr 2017

This value set has >1000 codes in it. In order to keep the publication size manageable, only a selection (1000 codes) of the whole set of codes is shown

All codes from system <http://snomed.info/sct>

Code	Display	Definition
109006	Anxiety disorder of childhood OR adolescence	
122003	Choroidal hemorrhage	
127009	Spontaneous abortion with laceration of cervix	
129007	Homolothemia	
134006	Decreased hair growth	
140004	Chronic pharyngitis	
144008	Normal peripheral vision	
147001	Superficial foreign body of scrotum without major open wound but with infection	
150003	Abnormal bladder continence	

### Valueset Operations: \$expand and \$validate

We are not discussing in-depth all the vocabulary issues in this introductory course...but we will introduce three terminology operations. The first one was '\$lookup' for CodeSystem. Now we will review two operations for ValueSet: \$expand, and \$validate.

Of course, these are more complex than they seem here, but we will use them just to show you the Operations concept, and the fact that the profile, the constraints, and the vocabulary for your implementation guide...everything can be leveraged from your FHIR native server operations, if implemented, of course.

Just for the sake of curiosity, now that you know the PPH table, perform these operations in our server, no need to use a FHIR or REST Client, just from your internet browser:

[http://fhirserver.hl7fundamentals.org/fhir/ValueSet/\\$expand?url=http://minhealth.gov.bk/core/ValueSet/PPHTaxSituationVS&filter=mandated](http://fhirserver.hl7fundamentals.org/fhir/ValueSet/$expand?url=http://minhealth.gov.bk/core/ValueSet/PPHTaxSituationVS&filter=mandated)

This will only include the codes for MANDATED PPH Tax Situation

And finally these two:

[http://fhirserver.hl7fundamentals.org/fhir/ValueSet/\\$validate-code?url=http://min-health.gov.bk/core/ValueSet/PPHTaxSituationVS&code=M](http://fhirserver.hl7fundamentals.org/fhir/ValueSet/$validate-code?url=http://min-health.gov.bk/core/ValueSet/PPHTaxSituationVS&code=M)

You should get a “**Found Code**” message.

[http://fhirserver.hl7fundamentals.org/fhir/ValueSet/\\$validate-code?url=http://min-health.gov.bk/core/ValueSet/PPHTaxSituationVS&code=X](http://fhirserver.hl7fundamentals.org/fhir/ValueSet/$validate-code?url=http://min-health.gov.bk/core/ValueSet/PPHTaxSituationVS&code=X)

You should get a “**Code Not Found**” message.

## ConceptMap

ConceptMap allows to precisely define the relationship between concepts (and its related codes) in one Code System to concepts in another. This mapping is not ‘absolute’, maybe it depends on the use case (it’s a mapping for a certain goal).

This resource includes the metadata for the map (source and destination, version, author, etc.) and a list of codes in the source code system and the relationship for each code to one or more codes in the target code system.

You can see one example here, where the specimen type for HL7 V2 Table 0487 (Specimen Type) is mapped to SNOMED CT.

<https://www.hl7.org/fhir/conceptmap-example-specimen-type.html>

This resource has a related special operation called **\$translate**, which allows the server to give you the equivalence in one code system to the codes you have in a source code system.

This is an example of this operation:

```
GET [base]/ConceptMap/$translate?system=http://hl7.org/fhir/composition-status  
&code=preliminary&valueSet= http://hl7.org/fhir/ValueSet/composition-status  
&target=http://hl7.org/fhir/ValueSet/v3-ActStatus
```

## NamingSystem

This resource represents a namespace issuing unique identifiers for concepts (classification, code systems, value sets, etc.) or entities (people, devices, organizations, orders, etc.)

You will use one of the `uniqueId.value` elements of the NamingSystem for use as ‘system’ in your coded or identifier elements

NamingSystem includes the name, kind (code system or identifier), responsible, publisher, use context, and all equivalent unique identifiers for the system.

You can explore a NamingSystem example here: <https://www.hl7.org/fhir/namingsystem-example-id.xml.html#DomainResource.text.div-end>

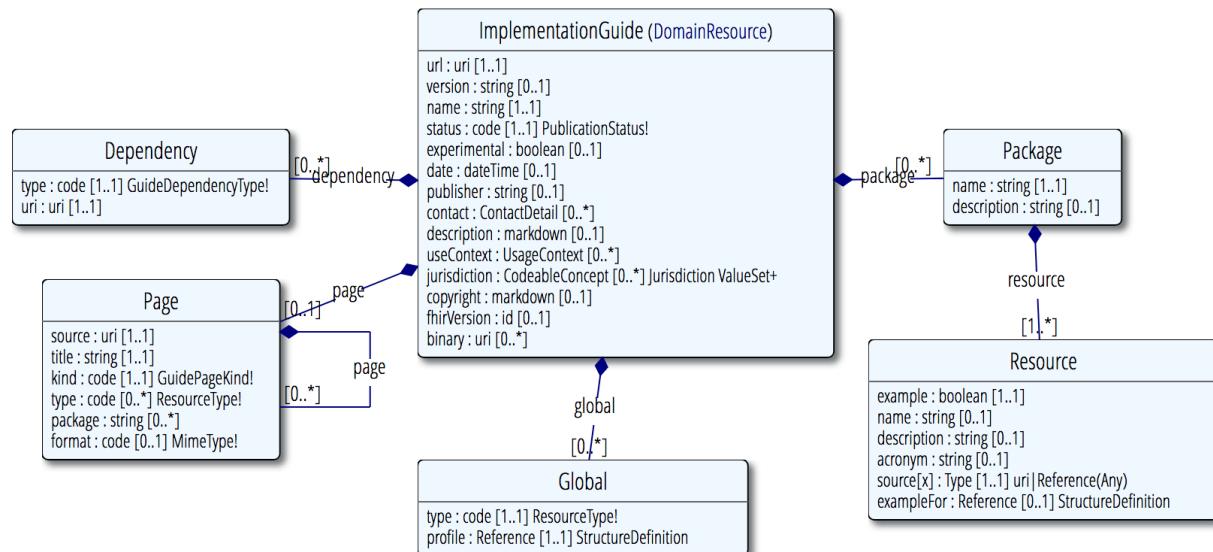
# Publishing

## ImplementationGuide

An implementation Guide is a package that bundles a set of StructureDefinition, NamingSystem, OperationDefinition, ValueSet and all other needed conformance resources, intended for publication. It contains the name, status, version of FHIR it is based on, and the HTML or markdown pages with the narrative for the implementation guide (scenarios, use cases, interaction details, etc.). So, this is what a typical implementation guide will include:

- 1) Title
- 2) Table of contents
- 3) Document information
- 4) Introduction to the guide
- 5) Principles & background
- 6) Functional requirements-level use cases
- 7) Design considerations
- 8) Package Contents
- 9) Privacy and Security Guidance
- 10) List of all artifacts used

Items 1 to 9 are just **Pages**: html or markdown that you need to create, using whatever editor you want to use, and the ancillary formatting artifacts (css, js, img files). Item 10 is a list of **all constrained resources grouped into packages**. You can also include **dependencies** on other implementation guides. You can build an implementation guide based on resources created by other implementation guide (remember that implementation guides are FHIR resources too, so this dependency is **computable**).



## Processing the ImplementationGuide

To create and publish an implementation guide one you have your ImplementationGuide and other resources, the process may still be cumbersome.

Some options:

- 1) Save all your resources and narrative to simplifier.net and ask Simplifier to build your implementation guide. The implementation guide will look like this one below:

The screenshot shows the header of the PCR FHIR Implementation Guide. It features the eHealth Ontario logo and the title "Provincial Client Registry (PCR) FHIR Implementation Guide - v1.0". Below the title is a navigation bar with links: Introduction, Business Context, Implementation Guidance, Profiles & Operations, Conformance Requirements, Terminology, and Downloads. The "Implementation Guidance" link is highlighted.

### Introduction

#### Background

The Provincial Client Registry (PCR) is a repository that stores demographics and identifiers pertaining to patients who have received health care in Ontario. It enables organizations to uniquely identify patients based on their local or global identifiers (e.g. MRN or HCN) and demographic information. The repository also enables the merging and unmerging of duplicate records from a variety of contributing sources, leading to improved quality of client information across different health care organizations.

The PCR supports eHealth Ontario programs such as identity, access and privacy, the Ontario Laboratory Information System (OLIS), Diagnostic Imaging Common Services, the cardiac care network, Connecting Ontario patient results online, and the Ministry of Health and Long-Term Care's Registered Person Database (Ontario Health Insurance Plan).

Ontario has published the following standards for querying or contributing to the PCR:

- Pan-Canadian HL7 V3 based PCR standard (used for both contribution and querying)
- HL7 V2 based standard (used exclusively for contribution)
- IHE PIX/PDQ based standard (used exclusively for querying)

This specification provides an additional FHIR-based interface for the PCR. It exposes 4 operations:

1. [IHE PIXm - Patient Identifier Cross-reference](#)
2. [Patient EMPI match](#)
3. [Patient Read](#)
4. [Patient Search](#)

Additional sections provide documentation on eHealth Ontario's use of the FHIR standard as well as on how [errors](#) are handled across the various types of interactions.

## Disadvantages of Simplifier.net: it's not free

Advantages: lots of tools, GUI interface

## 2) Run the FHIR IG Publishing Tool and use FSH/SUSHI

The advantage of this option is that the IG will look exactly like the IG's published by HL7. If you are interested in exploring this option, there is an entire section of HL7's confluence page with the details:

<https://confluence.hl7.org/display/FHIR/IG+Publisher+Documentation>

You also need 'input' for the Publisher, one of the ways it through a set of tools called FSH+SUSHI

FSH is a standard language created by HL7 to express constraint on resources.

SUSHI is a FSH compiler: it creates FHIR conformance resources (StructureDefinitions, ValueSets, CodeSystems, etc.) from FSH files, and prepares them to be processed by the FHIR IG Publisher

This is the Berzerkistan Patient resource profile written in FSH

```

Profile:      BerzerkistanPatient
Parent:       Patient
Id:          BerzerkistanPatient|
Title:        "Patient for Berzerkistan"
Description:   "Berzerkistan Patient Profile"
* extension contains PPHTaxSituation named PPHTaxSituation 1..1
* identifier 2..2
* identifier ^slicing.discriminator.type = #value
* identifier ^slicing.discriminator.path = "use"
* identifier ^slicing.rules = #open
// BniIdentifier
* identifier contains BniIdentifier 1..1
* identifier[BniIdentifier].use = #official
* identifier[BniIdentifier].system = "http://www.berzerkistan.gov/bni"
* identifier[BniIdentifier].value 1..1
* identifier[BniIdentifier].system 1..1
// NHibIdentifier
* identifier contains NHibIdentifier 1..1
* identifier[NhibIdentifier].use = #usual
* identifier[NhibIdentifier].system = "http://www.berzerkistan.gov/nhib"
* identifier[NhibIdentifier].value 1..1
* identifier[NhibIdentifier].system 1..1
* name 1..1
* name.family 1..1
* name.given 1..*
* gender 1..1
* birthDate 1..1
* telecom 1..1
* telecom.system 1..1
* telecom.value 1..1
* telecom.system = #phone
* telecom.value obeys phone-number-country-code
* text 1..1

```

Too much information? Want more? This is a complete course on FSH/SUSHI and the Publisher created by Mitre: <https://fshschool.org/courses/fsh-seminar>

## Resource Validation against a Profile

### Conformance Declaration in Resource Instances

To declare conformance to a specific profile in a FHIR instance / resource, you need to include a specific tag: **profile** in the metadata section of the resource instance, referencing the implementation guide or profile.

Example: this declaration inside of the json of your Patient resource is claiming conformance with the guide we've discussed throughout this unit.

```
"meta": {
  "profile": [
    "http://minhealth.gov.bk/core/StructureDefinition/BerzerkistanPatient"
  ]
}
```

### Resource Validation

In principle, resources can be validated against the original specification (the FHIR standard), against an implementation guide (conformance) or against a specific profile (conformance). Resources can be tested for conformance:

- 2) by using the **XML Schema or JSON Schema** for the base resource, basically what you've done in our first units.
- 3) by using the **FHIR validator** (it's a java JAR that you can download from here: <https://www.hl7.org/fhir/validator.zip>)
- 4) by asking a FHIR server using the special **\$validate** operation. It's the simplest way to execute this operation is to post the resource to a server:

\$validate operation example:

POST [base]/**Patient**/\$validate?profile=http://hl7.org/fhir/StructureDefinition/daf-patient

The <patient> resource to be checked needs to be in the POST body

After validating the resource, the server will return an OperationOutcome resource with the issues found.

Remember that not all servers support the \$validate operation, though some of the public test servers do, and that servers supporting the \$validate operation will only validate against profiles already registered with the server.

All these validation methods are incomplete. They can only validate the 'computable' aspects of conformance.

Some narrative rules can only be checked by inspection of the resources. The schema/schematron is the least capable, it doesn't check terminology.

**The final word is always derived from human inspection of the content of the resources, and the comparison with the relevant implementation guides requirements.**

## Tooling to Create Profiles

You can create Conformance resources just like you create any other FHIR resource, by editing XML or JSON... But it's a very cumbersome and complex process.

Several tools for creating FHIR conformance resources have been developed (and continue to evolve).

These are some of the tools. None of them are HL7's official tool for creating profiles. HL7 Workgroups used until not long ago, for profiles and balloting current artifacts a tool called Microsoft Excel (!)

**Disclaimer: This course does not endorse any commercial product, tooling, or company. We are just giving you directions on how to learn more about these subjects and including some screenshots for you to see what the tools are about.**

You are welcome to download the tools and try them (or just use them if they are web based)

## Forge

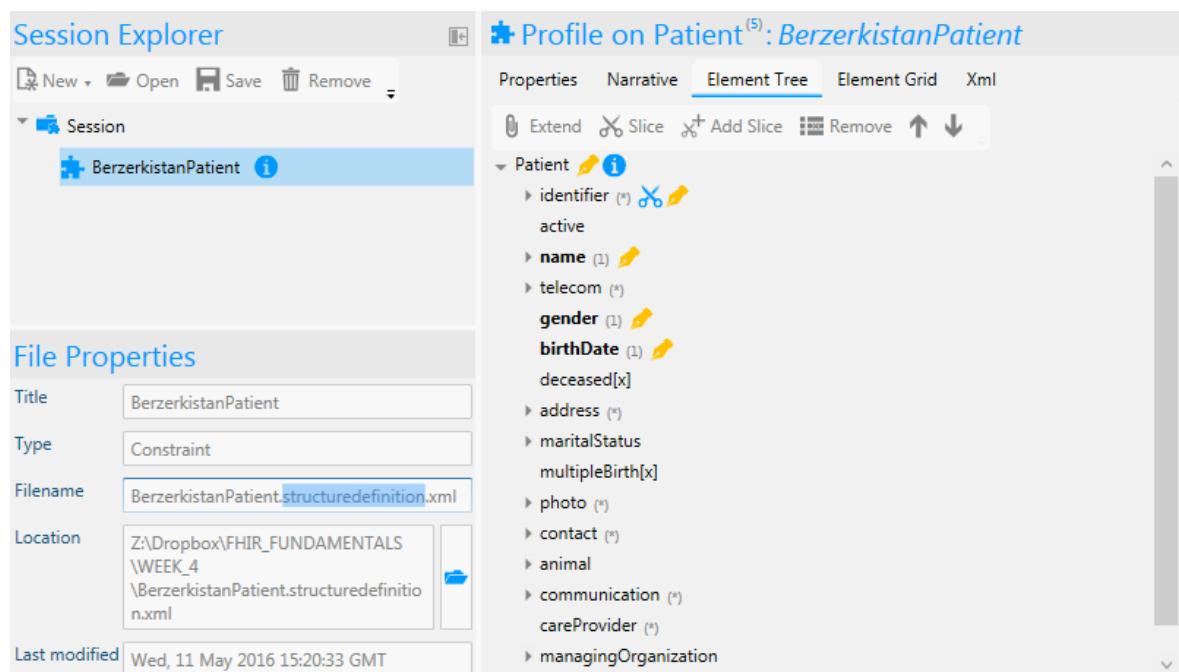
Web Site: <https://fire.ly/forge/>

Created and maintained by: Firely, The Netherlands

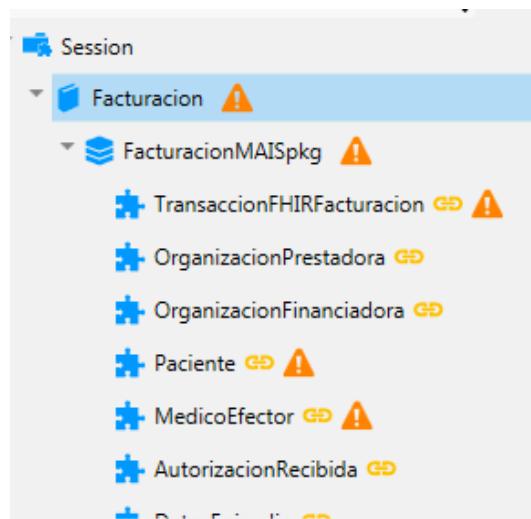
Conditions: Some free plans but essentially not free. Runs only in Windows

Description: conformance resource editor for profiles, constraints, extensions, slices and vocabulary bindings. It can be connected to simplifier or any other FHIR conformance resource server (remember that all the constraints can be saved as FHIR resources in any FHIR server).

Forge Explorer and Profile Editor



## Forge Implementation Guide Explorer



## Trifolia

Web Site: <https://trifolia.lantanagroup.com>

Created and maintained by: Lantana Group, US

Conditions: Requires registration, free for HL7 members, web based

Description: Originally created for the creation of HL7 CDA R2 implementation guides. FHIR functionality added in 2015. Allows editing of implementation guides, templates (profiles), and terminology (value sets and code systems)

The screenshot shows the Trifolia interface for managing FHIR profiles. At the top, it says "EHO FHIR eConsult". Below that are buttons for "FHIR DSTU2", "HL7", and "Draft". A navigation bar includes "Templates/Profiles" (with a note icon), "Notes", "Primitives", and "Audit Trail". A search bar is present. The main content area is titled "Bundle" and displays the URL "EHO eConsult Bundle" with the value "<https://ehealthontario.on.ca/eConsult/FHIR2/StructureDefinition/bundle>". There are also search and filter icons.

Trifolia's Profile Browser

This screenshot shows the "EHO eConsult Bundle" page. At the top, there are buttons for "Back to List", "Copy", "Bulk Copy", and "New Version". Below that is the URL "https://ehealthontario.on.ca/eConsult/FHIR2/StructureDefinition/bundle". A toolbar below the URL includes "EHO FHIR eConsult", "Draft", "Bundle", "Open", "HL7", and "borna jafarpour (HL7)". A navigation bar at the bottom includes "Constraints", "Relationships", "StructureDefinition JSON", and "StructureDefinition XML".

1. **SHALL** contain exactly one [1..1] type (**CONF:3248-191**).
2. **SHALL** contain exactly one [1..1] total (**CONF:3248-196**).
3. **MAY** contain zero or more [0..\*] entry (**CONF:3248-190**).
  1. The entry, if present, **SHALL** contain exactly one [1..1] fullUrl (**CONF:3248-193**).
  2. The entry, if present, **MAY** contain zero or one [0..1] resource (**CONF:3248-194**).

Trifolia's Constraint Viewer/Editor. A constraint over a bundle resource

## Trifolia on FHIR

<https://trifolia-fhir.lantanagroup.com/home>

FHIR specific profile editor developed by Lantana Group

## clinFHIR

Web Site: <http://clinfhir.com/>

Created and Maintained by: Dr. David Hay, HL7 New Zealand

Description: Resource editor for non-technical users, profile and extension editor and commenter.

You can select your Patient, Profile and Terminology server, and create your resources

clinFHIR has several other features you can test.

The screenshot shows the clinfhir Resource Editor interface. At the top, it displays the patient information: Patient: John Smith (461) Details and Profile: StructureDefinition/Observation. Below this are four buttons: Validate, Park, Save, and Cancel. The main area is divided into three sections: Resource Navigator, Element Details, and Resource: Json.

- Resource Navigator:** Shows the hierarchy of the Observation resource, with status and subject expanded.
- Element Details: Observation:** Displays child elements and their types. For example, identifier is a 0..\* Identifier, status is 1..1 code, category is 0..1 CodeableConcept, and subject is 0..1 Patient.
- Resource: Json:** Shows validation errors. A message says "Sorry, there were some issues!". A table lists the errors with columns for Severity, Code, and Details/Diagnostics.

Severity	Code	Details/Diagnostics
error		Error in component.where(code = \$context.code).empty() at 32: Found . expecting a valid token name (000000000BCE0E1){FHIRServer.exe} [0000000000FCF0E FHIRPath.TFHIRExpressionEngine.parseExpression + \$381 (0000000000040E9B){FHIRServer.exe} [0000000000441E9E System.SysUtils.RaiseExceptObject + \$2B (00000000000FD66){FHIRServer.exe} [0000000000410D6E]

clinfhir Resource Editor

The screenshot shows the clinfhir Extension Builder/Viewer interface. It has tabs for Query Parameters, Results, and results-table. The main area is titled "Find Extensions for Patient" with "Cancel" and "Select" buttons.

**Query Parameters:**

- Place of Birth for patient:** Description: The registered place of birth of the patient. A system may use the address.text if they don't store the birthPlace address in discrete elements.
- Target anatomic location or structure:** Description: Record details about the anatomical location of a specimen or body part. This resource may be used when a coded concept does not provide the necessary detail needed for the use case.
- Conformance expectation:** Description: Defines the level of expectation associated with a given system capability.

**Results:**

<b>Url</b>	http://hl7.org/fhir/StructureDefinition/birthPlace
<b>Name</b>	Place of Birth for patient
<b>Description:</b>	The registered place of birth of the patient. A system may use the address.text if they don't store the birthPlace address in discrete elements.
<b>Context</b>	Patient
<b>Publisher</b>	Health Level Seven International (FHIR Infrastructure)
<b>Data Type</b>	address

**results-table:**

```
{
  "resourceType": "StructureDefinition",
  "id": "birthPlace",
  "meta": {
    "versionId": "1",
    "lastUpdated": "2016-03-28T06:16:54Z"
  },
  "url": "http://hl7.org/fhir/StructureDefinition/birthPlace",
  "name": "Place of Birth for patient",
  "status": "draft",
  "publisher": "Health Level Seven International (FHIR Infrastructure)",
  "contact": [
    {
      "name": "FHIR Team"
    }
  ]
}
```

clinfhir Extension Builder/Viewer

## Snapper

Web Site: <https://ontoserver.csiro.au/snapper2/licence.html?source=author>

Free use, requires accepting the terms (for Author, Mapper and SNOMED CT).

Created and maintained by: The Australian e-Health Research Centre, CSIRO

Description: FHIR Terminology Editor and Browser (ValueSet, CodeSystem and ConceptMap).

	Code	Display	Definition
U	Uncovered	Patient pays 21%	
V	Voluntary	Patient pays 10.5%	
M	Mandated	Patient pays 0%	
Code that identifies the c...	Text to display to the user	Formal definition	

You can import, edit, and export or publish your code systems, value sets and concept maps using Snapper!

## Some Important Profiles/Implementation Guides

This section will list some profiles that are important, and you need to keep at the top of your mind if you think about implementing FHIR (and really get connected to anyone else out there). Not all of them are traditional implementation guides as described in this unit, but all of them are relevant.

**Again, we are not recommending any commercial product or company, just trying to let you know what is happening and where to find the stuff you need to participate.**

### Argonaut Project

Web Site: <https://confluence.hl7.org/display/AP>

**Description:** The Argonaut project is a private sector initiative to develop a FHIR based API to enable sharing of electronic healthcare records.

**Sponsors:** Accenture, Athenahealth, Beth Israel Deaconess Medical Center, Cerner, Epic, Intermountain Healthcare, Mayo Clinic, MEDITECH, McKesson, Partners Healthcare System, Boston Children's Hospital, Surescripts, The Advisory Board Company

Components: Secure Access through Smart App -> Data / Document Query

#### Argonaut Data Query Implementation Guide:

(<http://www.fhir.org/guides/argonaut/r2/>)

For secured access for patients/physician to health data. It has profiles to enable access to Patient records in any hospital – provided that the Hospital adopted the guide- , and then granular information about Allergies, Care Plan & Team, Conditions, Devices, Laboratory Tests and Reports, Immunizations, Medications, Procedures, Smoking Status, Vital Signs, and Documents.

#### Argonaut Provider Directory Implementation Guide

(<http://www.fhir.org/guides/argonaut/pd/>) for provider directory (includes searching practitioners and organizations)

There are more Argonaut implementation guides in the making: Scheduling, CDS-Hooks (enabling clinical decision support using FHIR resources), etc.

## Adoption:

Maybe this is THE most relevant guide in terms of adoption. It was adopted not only by the project sponsors, but by Apple, who adopted an earlier version of this guide for its Health module (embedded into IOS 11.3+). Apple claims to have thousands of hospital/groups in the US, UK and Canada connected.

An updated list can be obtained from here: <https://support.apple.com/en-us/HT208647>



A screensnap of Apple's personal health record feature with iOS 11.3.

See more: <http://www.healthcareitnews.com/news/how-apples-health-records-could-reshape-patient-engagement>

Also it was adopted (silently) by Google, initially through its subsidiary Apigee, in a product called apigee-health-apix (Managed API for Healthcare).

See more here : <https://cloud.google.com/architecture/using-apigee-with-the-cloud-healthcare-api>

## IHE-on-FHIR

Although IHE used to publish its implementation guides as giant PDF files, it is embracing the FHIR standard for several profiles because of industry alignment and maybe also because FHIR has finally ‘come of age’.

All the FHIR based IHE profiles are listed here: <https://wiki.ihe.net/index.php/Category:FHIR>, so we will not include the actual links to each one of them.

The **first set of profiles** were a step to **FHIR-enable the IHE ‘legacy’** (and more relevant) profiles:

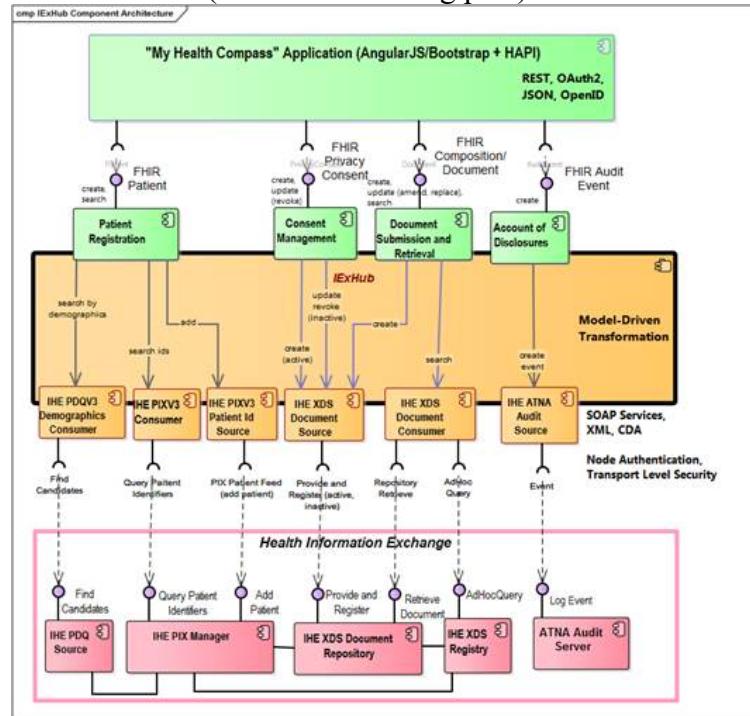
- ATNA / Restful Query for ATNA – For audit log
- Mobile Access to Healthcare Documents (MHD) – “XDS on FHIR”
- Patient Demographics Query for Mobile / Patient Identifier Cross Ref for Mobile - “PIX/PDQ on FHIR”
- Query for Existing Data for Mobile (QEDm) – “QED on FHIR”

No longer published as PDFs, these are full-fledged, computable IGs. Example:

<https://profiles.ihe.net/ITI/PIXm/index.html>

If your company is invested on IHE profiles and wants to open a “FHIR-door” or ‘FHIR-façade’ (this is how it is technically described), then these are the specs you were looking for.

Picture taken from John Moehrke’s (IHE on FHIR blog post):



## Smart-On-FHIR

Web Site: <https://docs.smarthealthit.org/>

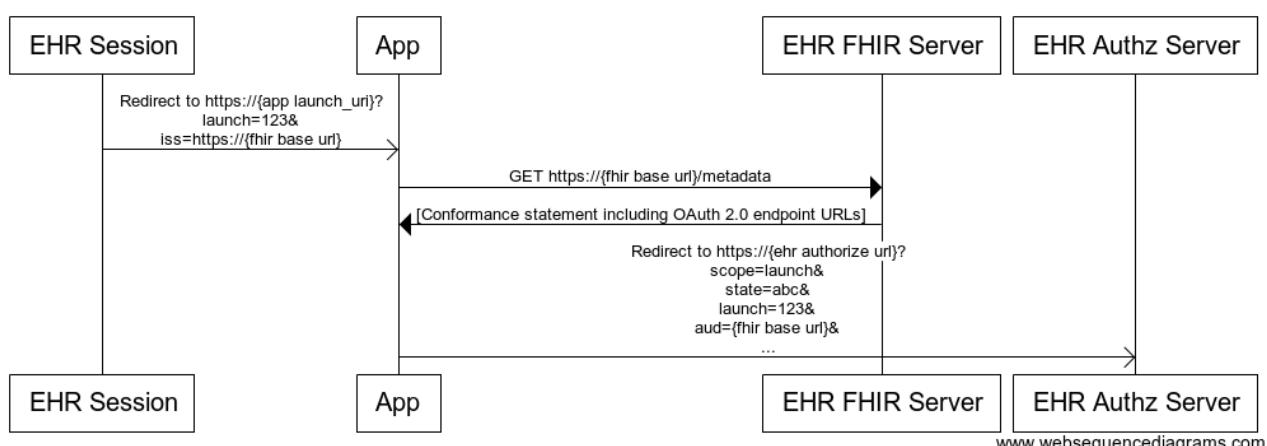
**Sponsors:** Originally created by Boston Children's Hospital Computational Health Informatics Program and the Harvard Medical School Department for Biomedical Informatics, it's now FHIR based and a FHIR standard with 2 releases.

**Components:** Smart-On-FHIR was at the beginning "Smart Healthcare", and was aimed to create 'an app platform for healthcare'. Mid-way of the development of their API, FHIR emerged, and they embraced it wholeheartedly. The main idea is to connect a marketplace of Apps for Healthcare to any hospital EHR (You can see the App Gallery here: <https://apps.smarthealthit.org/>)

It's a set of specifications for: **Structured Data:** basically, FHIR +Argonaut , **Scopes/Permissions:** oAuth2 (which patient/encounter/physician) is in context in the EHR, and **Authentication:** Simple Sign In through OpenID and **UI Integration:** Runs inside browser widgets or inline frames.

But what they provide is much more than an implementation guide: they created a set of open-source libraries that enables App creators (HTML5/Javascript/iOS/Python) to connect to FHIR-enabled servers. The libraries cover the topics described above.

**Adoption:** Boston Children's Hospital, DukeMedicine, CoxHealth, Geisinger, Hospital Corporation of America, Intermountain Healthcare, Ochsner, Allscripts, Cerner, Epic, Apple.



## FHIR Official Implementation Guides

All implementation guides **balloted and published officially by HL7 International** can be found here: <http://www.fhir.org/guides/registry>

All profiles (constraints on resources, extensions, valuesets, etc.) balloted and published by HL7 International can be found here: <https://registry.fhir.org/>

## Public Regional Implementation Guides

Public Regional (HL7 Affiliate created, from France, Finland, Norway, UK, etc.) implementation guides are usually found at Simplifier, at <https://simplifier.net/search?category=ImplementationGuide>

There may be other FHIR IGs but they are not advertised or they are private, so we cannot provide any more information about them.

## This week's assignment

We will validate instances of Berzerkistan patients against a FHIR server.  
If you made it through here, you are almost done. Go!

## Unit Summary and Conclusion

In this unit we discussed the **basic concepts of profiling and conformance** when using FHIR. We create profiles **to constraint standards to our specific scenarios and use cases**, and document our decisions, allowing automatic conformance validation when possible.

In FHIR you can constraint: **elements in resources** (optionality, repetitions), **vocabulary** - codes in coded elements- (specific vocabulary, including representing the actual code systems and value sets), **behavior** (which crud/non-crud methods and/or searches our FHIR server actually supports), and also which **extensions** are needed at each level (data type, resource) and if they are mandatory or not.

Profiling in FHIR is achieved by creating FHIR resources. These are some of FHIR resources used to document constraints: ImplementationGuide, CapabilityStatement, OperationDefinition, SearchParameter, StructureDefinition, ConceptMap, NamingSystem, DataElement, ValueSet, CodeSystem.

Some of the tools available to create and test profiles are Trifolia, Forge, Simplifier (constraint registry) and clinFHIR. Not all the tools support editing all the kind of conformance resources.

Conformance validation can be achieved by using the XML Schema for the resources (simple validation against the Schema), or by using FHIR servers' validation functionality. We also explored some widely used or relevant Implementation Guides.

Profiling in FHIR is still being developed, but we can start to dream with computable implementation guides and adequate tooling to create them.

## Additional Reading Material

### Information about FHIR

There are several places where you can get information about FHIR.

- The specification itself is available on-line at [www.hl7.org/FHIR](http://www.hl7.org/FHIR). It is fully hyperlinked & very easy to follow. It is highly recommended that you have access to the specification as you are reading this module, as there are many references to it - particularly for some of the details of the more complex aspects of FHIR.
- All the subjects detailed in this unit are deeply documented in the FHIR specification. If you can't explain something, this should be the first source-of-truth.
- The root HL7 wiki page for FHIR can be found at <http://wiki.hl7.org/index.php?title=FHIR> . The information here is more for those developing resources, but still very interesting. Some wiki information is more historical and may not reflect the most recent version of the specification.
- The team uses the 'stack overflow' site (<http://stackoverflow.com/questions/tagged/hl7-fhir>) as a place to answer implementation-related questions - and therefore have both question and answer available for reference. You can use the HL7 Help Desk if you are member.