



HL7 FHIR

For developers



Mirjam Baltus



- **Name:** Mirjam Baltus
- **Company:** Furore, Amsterdam
- **Background:**
 - Furore FHIR team
 - IT trainer
 - Support engineer



Rik Smithies



Independent Consultant
Technical Chair, HL7 UK
Health IT dev/architect (~20 years)
FHIR Certified Implementer
HL7 International Co-chair



Agenda



- Talk (me)
- Show (me → you)
- Do (you)

- Discuss (we)



FHIR BASICS RECAP

The Acronym



Fast

Relative – No technology can make integration as fast as we'd like

Healthcare

Interoperability

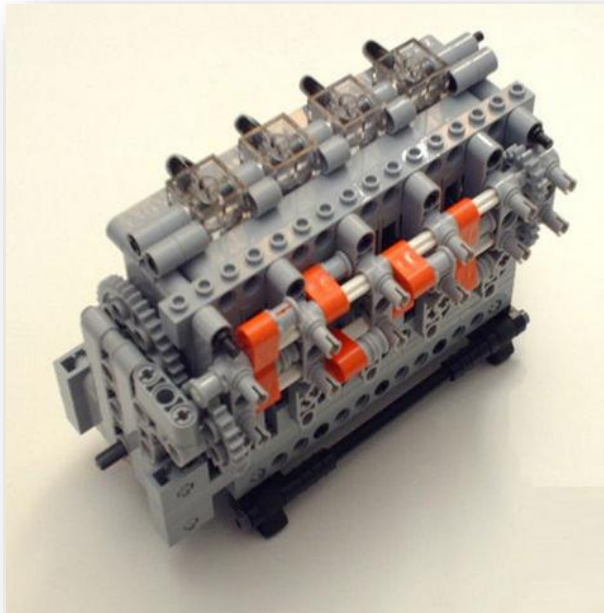
That's why we're here

Resources

Building blocks – more on these to follow



Resources



Patient



Prescription





1.2 Resource Index

This page is provided to help find resources quickly. There is also a more [detailed classification, ontology, and description](#). For background to the layout on the layers in this page, see the [Architect's Overview](#).

Categorized[Alphabetical](#)[R2 Layout](#)[By Maturity](#)[By Committee](#)

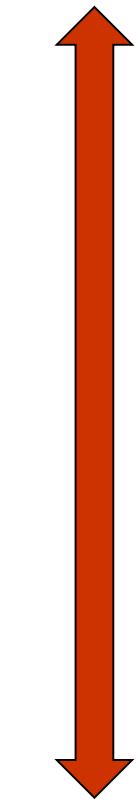
Foundation	Conformance	Terminology	Security	Documents	Other
	<ul style="list-style-type: none">• CapabilityStatement 3• StructureDefinition 5• ImplementationGuide 1• SearchParameter 3• MessageDefinition 0• OperationDefinition 4• CompartmentDefinition 1• StructureMap 2• GraphDefinition 0• DataElement 1	<ul style="list-style-type: none">• CodeSystem 5• ValueSet 5• ConceptMap 3• ExpansionProfile 2• NamingSystem 1	<ul style="list-style-type: none">• Provenance 3• AuditEvent 3• Consent 1	<ul style="list-style-type: none">• Composition 2• DocumentManifest 2• DocumentReference 3	<ul style="list-style-type: none">• Basic 1• Binary 5• Bundle 5• Linkage 0• Media 1• MessageHeader 3• OperationOutcome 5• Parameters 5• Subscription 3
	Individuals	Entities	Workflow	Management	
	<ul style="list-style-type: none">• Patient 5• Practitioner 3• PractitionerRole 2	<ul style="list-style-type: none">• Organization 3• HealthcareService 2• Endpoint 2	<ul style="list-style-type: none">• Task 2• Appointment 3• AppointmentResponse 3	<ul style="list-style-type: none">• Encounter 2• EpisodeOfCare 2• Flag 1	

<http://www.hl7.org/fhir/resourcelist.html>

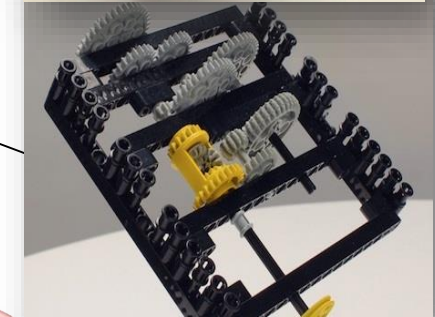
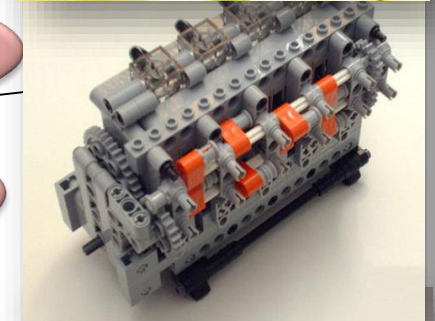
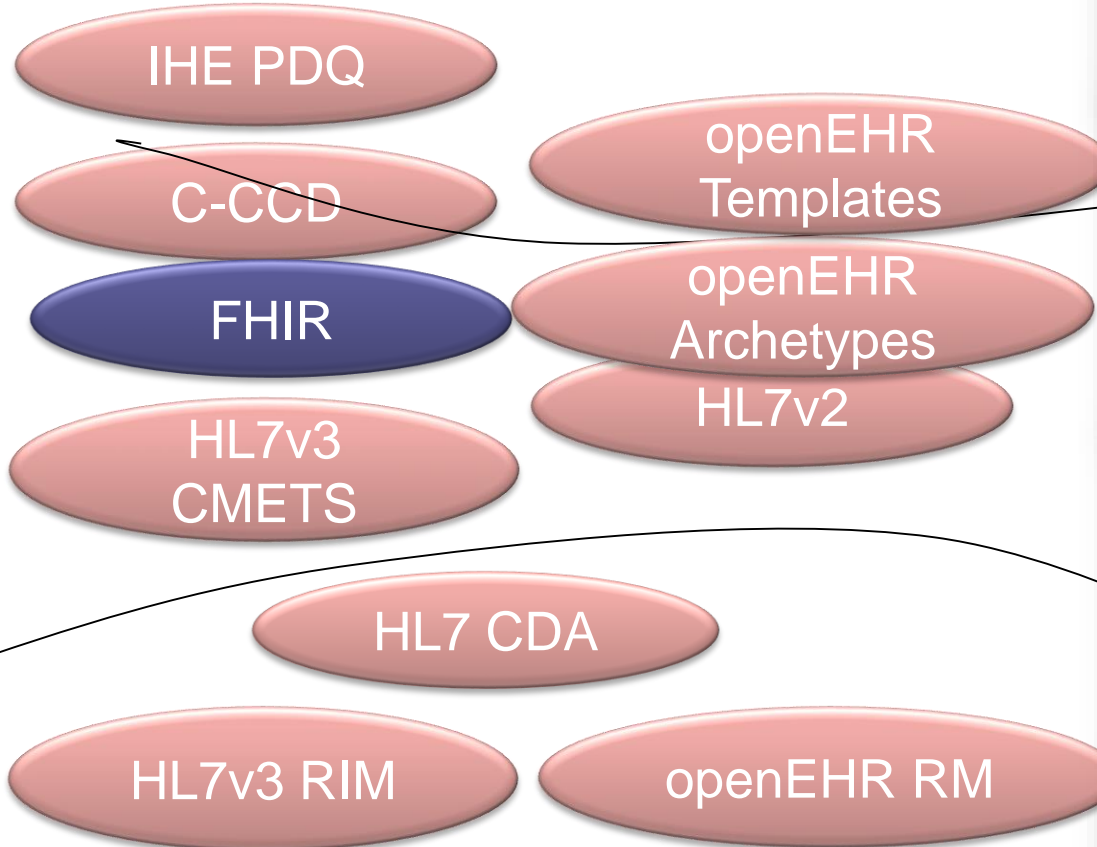


Cover all usecases - (n)ever

Specific



Generic



The 80/20 rule

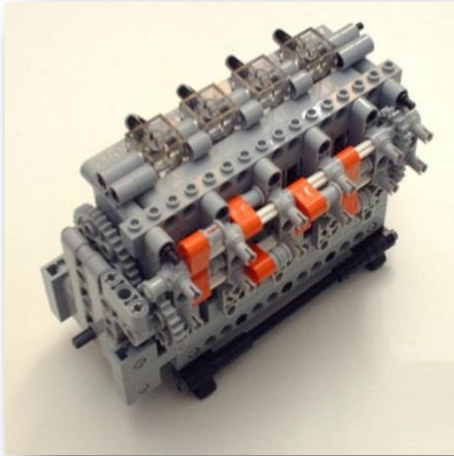


- Design for the 80%, not 100%
 - Only include data elements in the artifacts if 80% of all implementers of that artifact will use the data element
- Allow easy extension for the remaining 20% of elements
 - which often make up 80% of current specs
 - Vocabulary approach to extension definition

V3/OpenEHR are designed to cover the 100%.



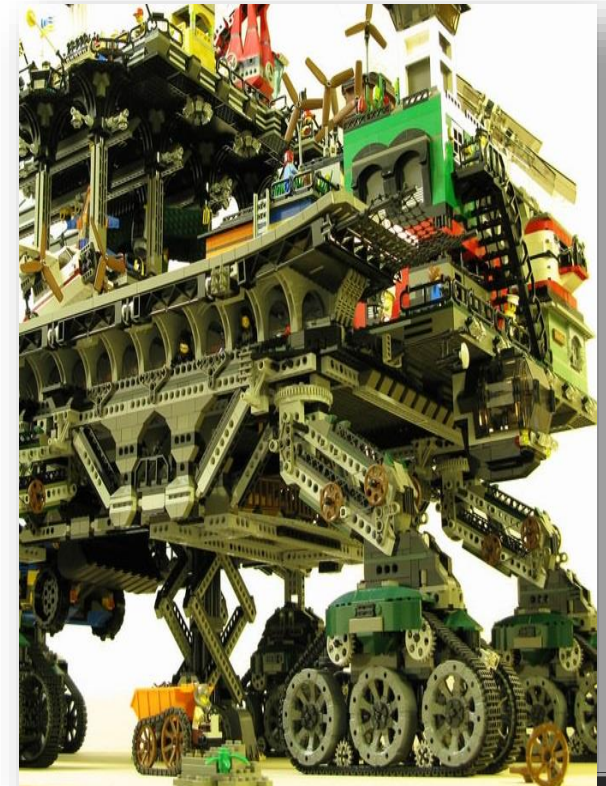
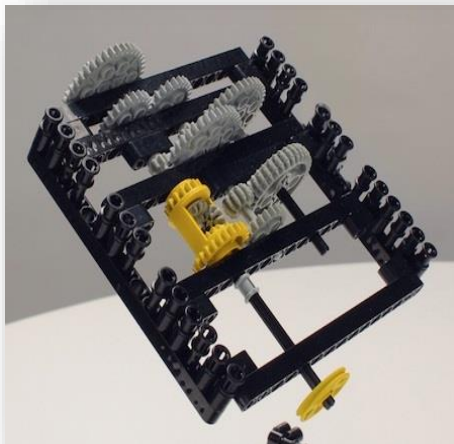
Extensibility



+



=



Structure of a Resource

(XML example)



```
<Patient xmlns="http://hl7.org/fhir">
```

```
  <id value="patient1"/>
```

```
  <meta>
```

```
    <versionId value="v2"/>
```

```
  </meta>
```

```
  <text>
```

```
    <status value="generated"/>
```

```
    <div xmlns="http://www.w3.org/1999/xhtml">
```

```
      <p>Henry Levin the 7th</p>
```

```
    </div>
```

```
  </text>
```

```
  <extension url="http://hl7.org/fhir/StructureDefinition/patient-birthTime">
```

```
    <valueDateTime value="1932-09-24T14:35:45-05:00"/>
```

```
  </extension>
```

```
  <identifier>
```

```
    <use value="usual"/>
```

```
    <system value="http://acme.org/identifiers/mrn"/>
```

```
    <value value="12345"/>
```

```
  </identifier>
```

```
  <name>
```

```
    <family value="Levin"/>
```

```
    <given value="Henry"/>
```

```
  </name>
```

```
  <gender value="male"/>
```

```
  <birthDate value="1932-09-24"/>
```

```
  <managingOrganization>
```

```
    <reference value="Organization/123xyz"/>
```

```
    <display value="Good Health Clinic"/>
```

```
  </managingOrganization>
```

```
  <active value="true"/>
```

```
</Patient>
```

▶ Metadata

▶ Human
Readable
Summary

▶ Extension with
reference to its
definition

▶ Standard Data
Content:

- MRN
- Name
- Gender
- Date of Birth
- Provider



A Resource's identity



- In fact: a URL

➤ `http://server.org/fhir/Patient/1`

Diagram illustrating the components of the URL:

- `http://server.org` is labeled as the **endpoint**.
- `fhir` is labeled as the **resource type**.
- `Patient/1` is labeled as the **logical id**.



"Business" identifiers



DiagnosticReport (DomainResource)

identifier : Identifier [0..*]

basedOn : Reference [0..*] CarePlan | ImmunizationRecommendation |
MedicationRequest | NutritionOrder | ProcedureRequest |
ReferralRequest

status : code [1..1] DiagnosticReportStatus!

category : CodeableConcept [0..1] Diagnostic Service Section ??

code : CodeableConcept [1..1] LOINC Diagnostic Report ?

subject : Reference [0..1] Patient | Group | Device | Location

context : Reference [0..1] Encounter | EpisodeOfCare

effective[x] : Type [0..1] dateTime | Period

issued : instant [0..1]

specimen : Reference [0..*] Specimen

result : Reference [0..*] Observation

imagingStudy : Reference [0..*] ImagingStudy | ImagingManifest

conclusion : string [0..1]

codedDiagnosis : CodeableConcept [0..*] SNOMED CT Clinical
Findings??

presentedForm : Attachment [0..*]

Patient (DomainResource)

identifier : Identifier [0..*]

active : boolean [0..1]

name : HumanName [0..*]

telecom : ContactPoint [0..*]

gender : code [0..1] AdministrativeGender!

birthDate : date [0..1]

deceased[x] : Type [0..1] boolean | dateTime

address : Address [0..*]

maritalStatus : CodeableConcept [0..1] Marital Status +

multipleBirth[x] : Type [0..1] boolean | integer

photo : Attachment [0..*]

generalPractitioner : Reference [0..*] Organization | Practitioner

managingOrganization : Reference [0..1] Organization



It's all about combining resources . . .





DiagnosticReport			DomainResource	
	identifier	Σ 0..*	Identifier	
	subject	Σ 0..1	Reference(Patient Group Device Location)	
	context	Σ 0..1	Reference(Encounter EpisodeOfCare)	
	issued	Σ 0..1	instant	
	performer	Σ 0..*	BackboneElement	
	role	Σ 0..1	CodeableConcept	
	actor	Σ 1..1	Reference(Practitioner Organization)	
	specimen	0..*	Reference(Specimen)	
	result	0..*	Reference(Observation)	



Resource Reference



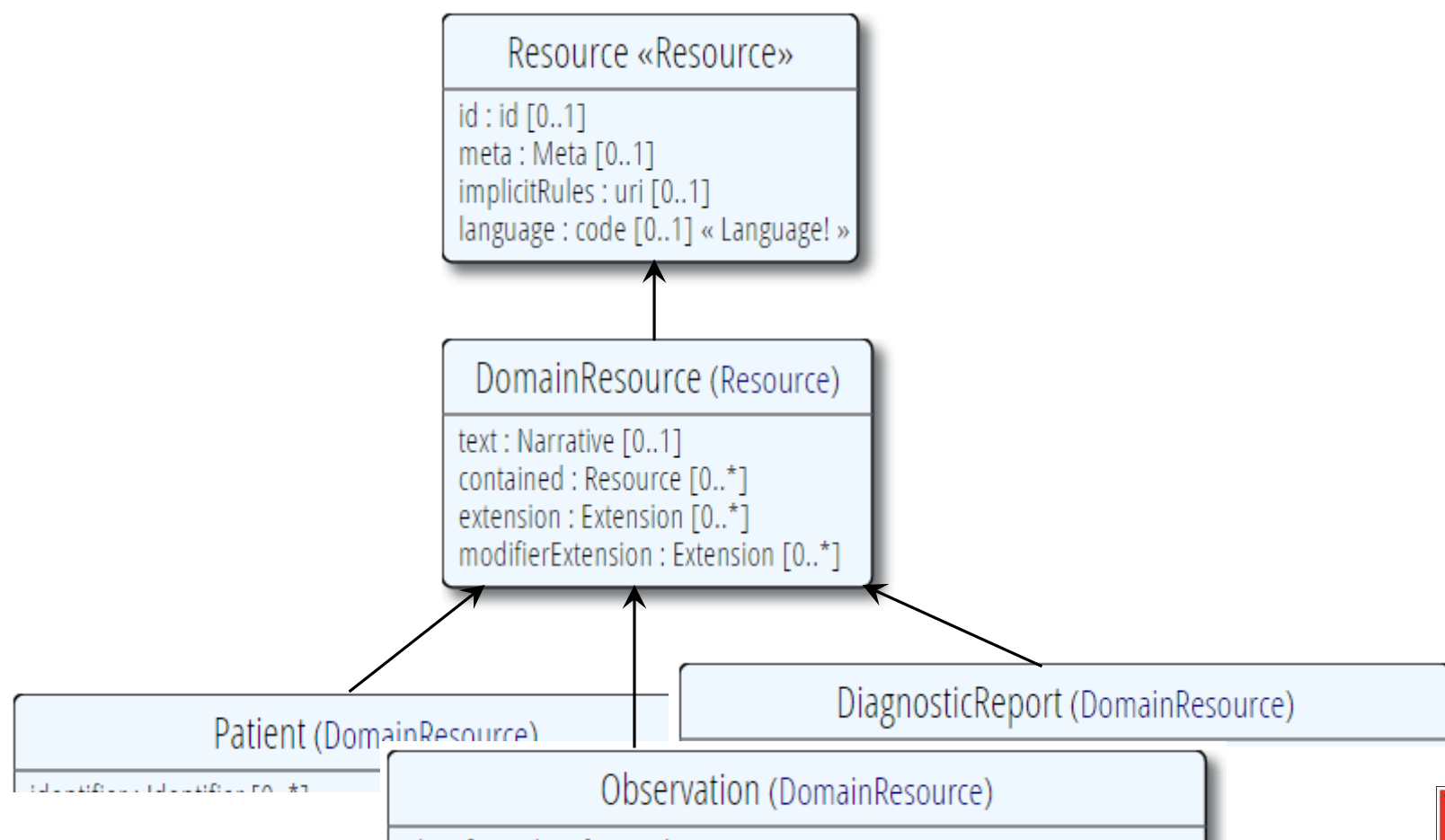
example snippet taken from a DiagnosticReport:

```
<subject>
  <reference value="Patient/f001"/>
  <display value="P. van den Heuvel"/>
</subject>
```

```
"subject": {
  "reference": "Patient/f001",
  "display": "P. van den Heuvel"
},
```



Technical hierarchy



Communicating resources

(with REST)



create

The create interaction creates a new resource in a server assigned location. The create interaction is performed by an HTTP POST operation as shown:

```
POST [base]/[type] {?_format=[mime-type]}
```

read

The read interaction accesses the current contents of a resource. The interaction is performed by an HTTP GET operation as shown:

```
GET [base]/[type]/[id] {?_format=[mime-type]}
```

update

The update interaction creates a new current version for an existing resource or creates a new resource if no resource already exists for the given id. The update interaction is performed by an HTTP PUT operation as shown:

```
PUT [base]/[type]/[id] {?_format=[mime-type]}
```

delete

The delete interaction removes an existing resource. The interaction is performed by an HTTP DELETE operation as shown:

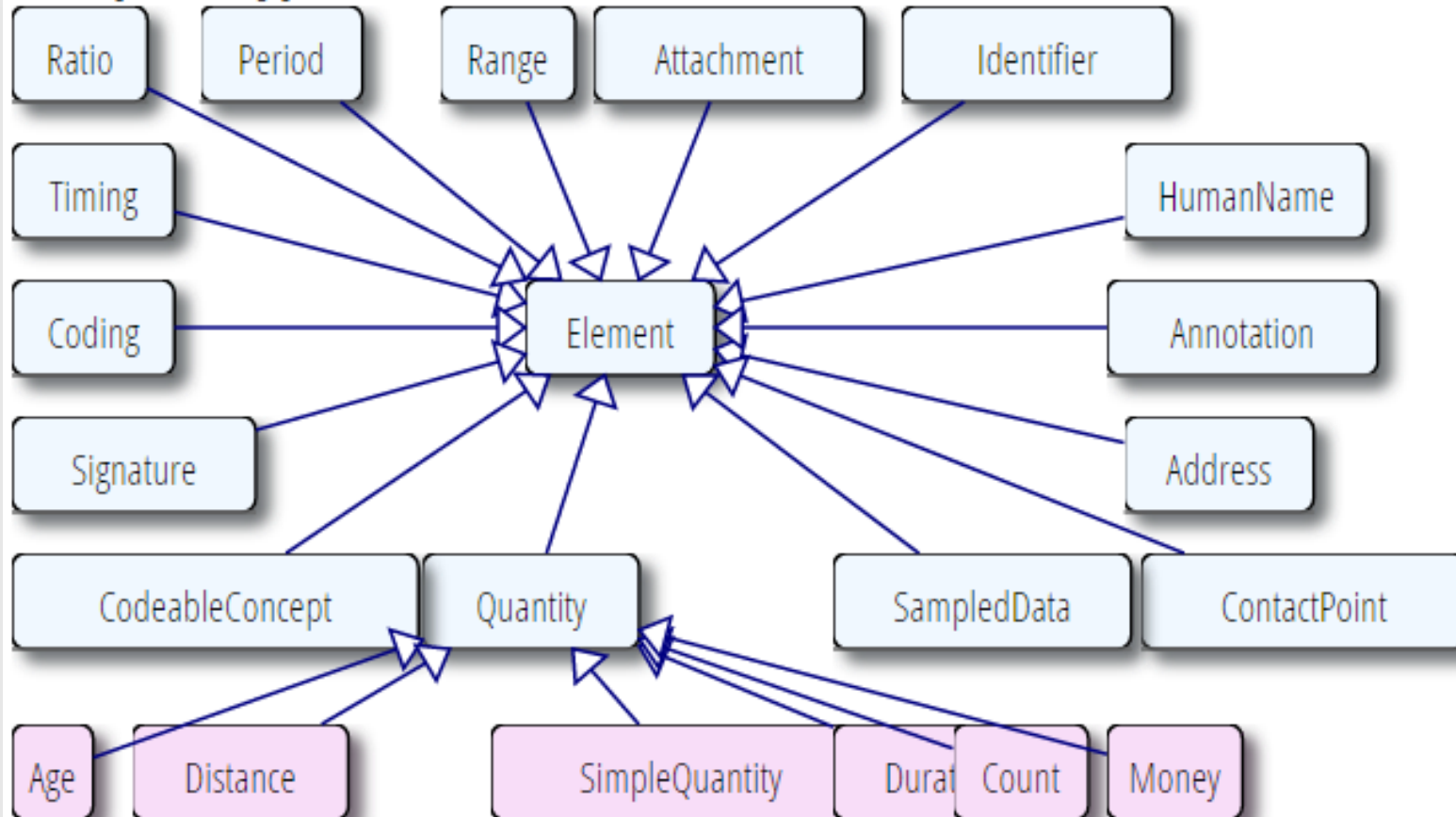
```
DELETE [base]/[type]/[id]
```



Datatypes, complex



Complex Types



Complex Datatypes



Quantity

value : decimal [0..1]
comparator : code [0..1] « QuantityComparator! »
unit : string [0..1]
system : uri [0..1]
code : code [0..1]

```
<time>  
  <value value="25" />  
  <unit value="sec" />  
  <system value="http://unitsofmeasure.org" />  
  <code value="s" />  
</time>
```

HumanName

use : code [0..1] NameUse!
text : string [0..1]
family : string [0..1]
given : string [0..*]
prefix : string [0..*]
suffix : string [0..*]
period : Period [0..1]

```
<name>  
  <family value="Everyman" />  
  <given value="Adam" />  
  <given value="A." />  
</name>
```



"Choice" properties



Observation (DomainResource)

```
<Observation>
```

```
  <valueQuantity>
```

```
    <value value="107"/>
```

```
    <units value="mm[Hg]" />
```

```
  </valueQuantity>
```

```
</Observation>
```

```
<Observation>
```

```
  <valueString
```

```
    value="Patient loves to sing" />
```

```
</Observation>
```

effective[x] : Type [0..1] dateTime | Period

issued : instant [0..1]

performer : Reference [0..*] Practitioner | Organization | Patient |

value[x] : Type [0..1] Quantity | CodeableConcept | string | boolean |
Range | Ratio | SampledData | Attachment | time | dateTime | Period

dataAbsentReason : CodeableConcept [0..1] Observation Value Absent
Reason

interpretation : CodeableConcept [0..1] Observation Interpretation





QUESTIONS?



How FHIR communicates sets of resources

BUNDLES

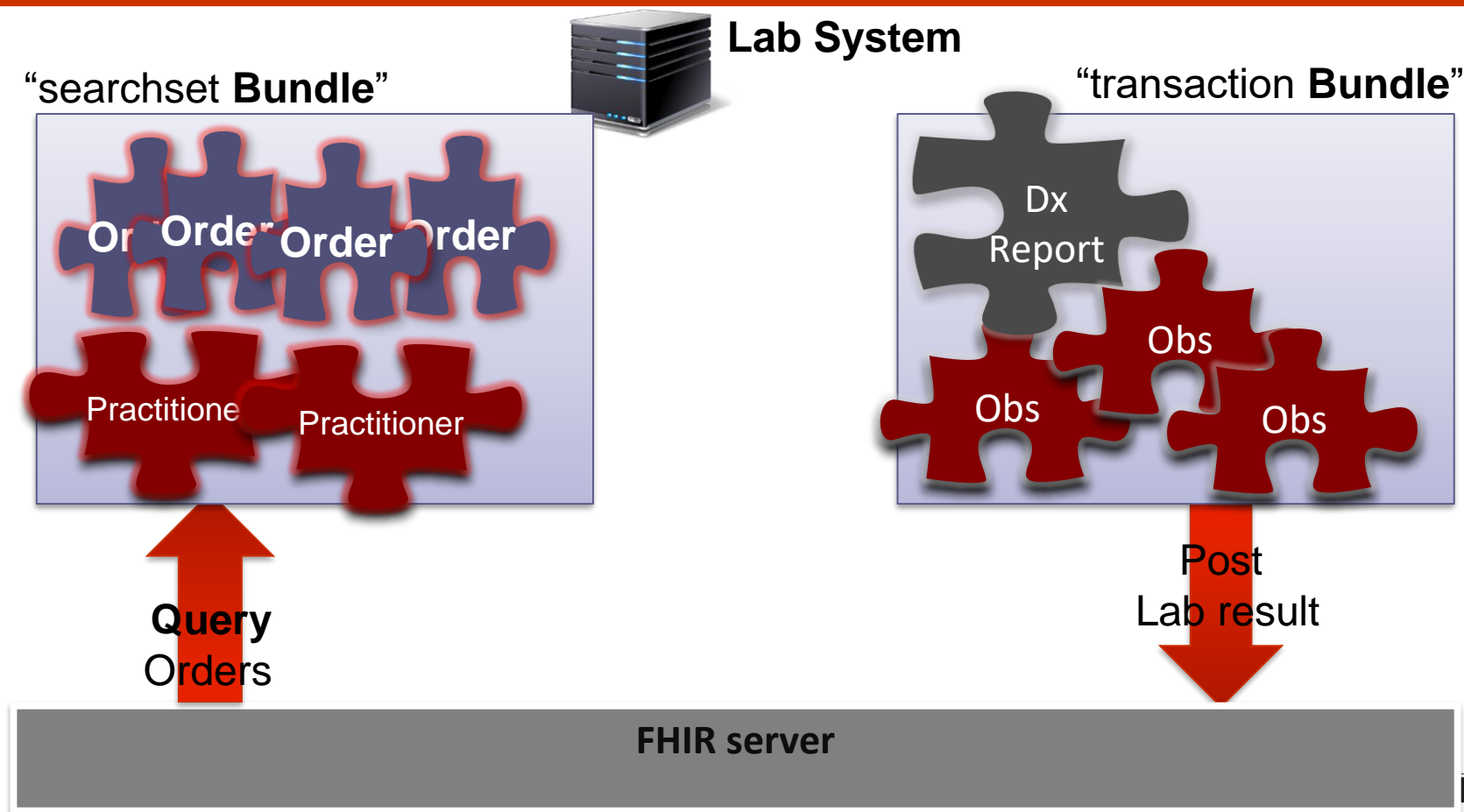
Communicating sets



- We need to communicate sets of Resources
 - Search result
 - History
 - Multiple-resource inserts (“batches”)
 - Transactions




Bundling resources



Bundle..is a resource!



2.36 Resource Bundle - Content

FHIR Infrastructure 
Work Group

Maturity
Level: 5

Trial
Use

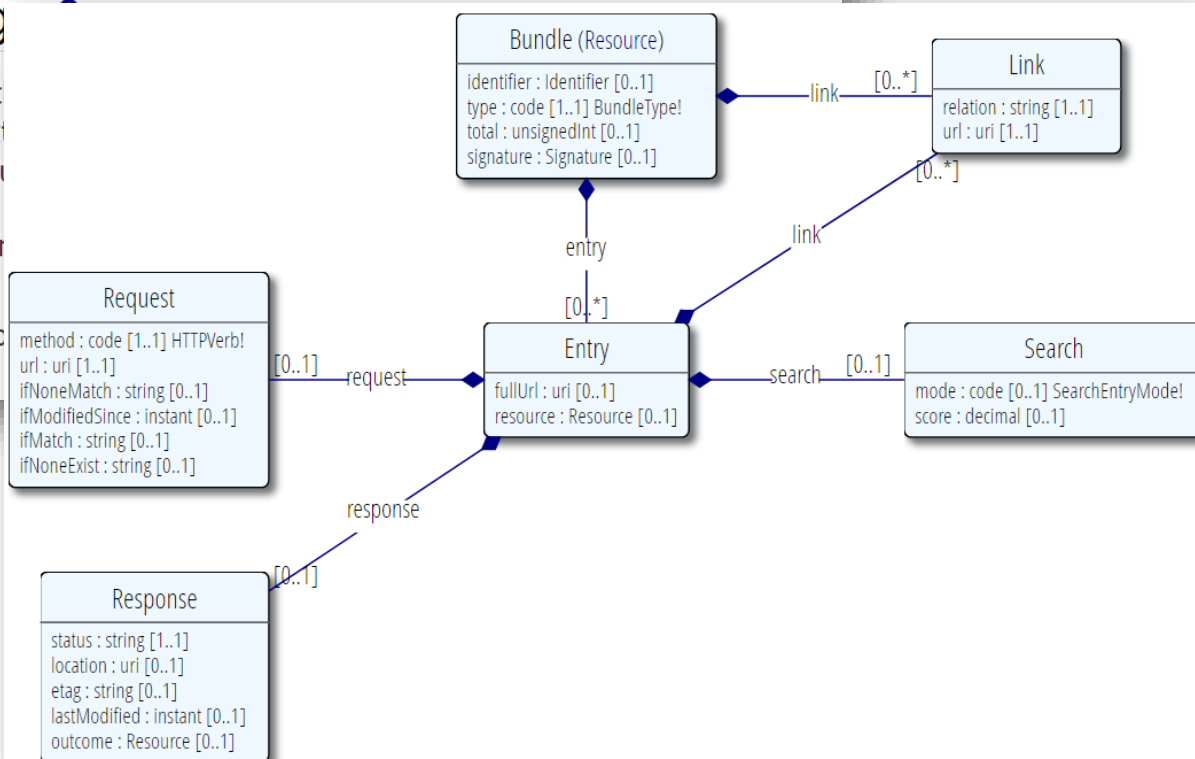
Compartments: Not linked to any defined compartments

A container for a collection of resources.

2.36.1 Scope and Usage

One common operation performed with a single instance with containing context together. These resource bundles are used for:

- Returning a set of resources that match a query (see [RESTful Search](#))
- Returning a set of versions of resources (see [History](#))



An example Bundle



```
<Bundle xmlns="http://hl7.org/fhir">
  <id value="urn:uuid:1d2de686-03d8-4451-afc0-193104c3464e" />
  <meta>
    <lastUpdated value="2015-04-26T20:25:15.6271425Z" />
  </meta>
  <type value="searchset" />
  <total value="1" />
  <link>
    <relation value="self" />
    <url value="http://fhir-dstu2-nprogram.azurewebsites.net/Patient/" />
  </link>
  <entry>
    <resource>
      <Patient xmlns="http://hl7.org/fhir">
        ...
      </Patient>
    </resource>
  </entry>
  <entry>
    <resource>
      <Organization xmlns="http://hl7.org/fhir">
        ...
      </Organization>
    </resource>
  </entry>
</Bundle>
```



Link types



Code	Description
current	The URL that defines this item
first	First set of records in a query set
last	Last set of records in a query set
next	Next set of records in a query set
previous	Previous set of records from this one
self	Conveys an identifier for the link's context.



SEARCH FUNCTIONALITY

Basic search



■ Syntax

```
GET [base]/[resourcetype]?name=value&...
```

■ Getting all patients

```
GET http://acme.org/fhir/Patient
```

■ Example:

```
GET http://acme.org/fhir/Patient?name=eve
```



Search parameters



Each resource has a set of “standard” search parameters, so **not every element can be searched!**

Name	Type	Description	Paths
active	token	Whether the patient record is active	Patient.active
address	string	An address in any kind of address/part of the patient	Patient.address
birthdate	date	The patient's date of birth	Patient.birthDate
family	string	A portion of the family name of the patient	Patient.name.family
gender	token	Gender of the patient	Patient.gender
given	string	A portion of the given name of the patient	Patient.name.given
identifier	token	A patient identifier	Patient.identifier
language	token	Language code (irrespective of use value)	Patient.language
link	reference	All patients linked to the given patient	Patient.link
name	string	A portion of either family or given name of the patient	Patient.name

Parameter
Type

Our last search
used this one

Parameter types



- For each type there are specific prefixes and modifiers

number, date and quantity:

Search for `[param]=gt[value]` retrieves resources where `[param]` has a value greater than `[value]`

`http://acme.org/fhir/Patient?name:contains=eve`

Ok I get it...or not?



<code>http://server.org/fhir/Patient/</code>	106 hits
<code>http://server.org/fhir/Patient?gender=male</code>	34 hits
<code>http://server.org/fhir/Patient?gender=female</code>	67 hits

Total: 34 + 67 = 101

<code>http://server.org/fhir/Patient/</code>	106 hits
<code>http://server.org/fhir/Patient?gender=male</code>	34 hits
<code>http://server.org/fhir/Patient?gender=female</code>	67 hits
<code>http://server.org/fhir/Patient?gender:missing=true</code>	5 hits

Total: 34 + 67 + 5 = 106



Search result parameters

■ Including other resources

GET [base]/DiagnosticReport?
_include=DiagnosticReport:subject:Patient

GET [base]/Patient?
_revinclude=Observation:subject

Source
resource type

Parameter of
type *reference*

Type of target
resource (optional)

Chained searches



- How do I find Observations for a patient, searching using his name?

```
GET [base]/Observation?subject.name=jim
```

Note: this still only works on the predefined search parameters. You cannot just use any property of the resource.



QUESTIONS?



The APIs

.Net and Java
(more are available)















THE MODEL

```
using Hl7.Fhir.Model;
```

```
hapi-fhir-structures-dstu3-2.4.jar  
└─ org.hl7.fhir.dstu3.model
```

A FHIR Resource



Name	Flags	Card.	Type	Description & Constraints
 Observation	I		DomainResource	Measurements and simple assertions <i>SHALL only be present if Observation.value[x] is not present</i> <i>Component code SHALL not be same as observation code</i>
 identifier		0..*	Identifier	Unique Id for this particular observation
 status	?! Σ	1..1	code	registered preliminary final amended + ObservationStatus (Required)
 category		0..1	CodeableConcept	Classification of type of observation Observation Category Codes (Example)
 code	Σ	1..1	CodeableConcept	Type of observation (code / type) LOINC Codes (Example)
 subject	Σ	0..1	Reference(Patient Group Device Location)	Who and/or what this is about
 encounter		0..1	Reference(Encounter)	Healthcare event during which this observation is made
 effective[x]	Σ	0..1		Clinically relevant time/time-period for observation
 effectiveDateTime			dateTime	
 effectivePeriod			Period	
 issued	Σ	0..1	instant	Date/Time this was made available
 performer	Σ	0..*	Reference(Practitioner	Who is responsible for the observation



A FHIR Resource in C#



```
public partial class Observation : Hl7.Fhir.Model.DomainResource
```

status	?! Σ	1..1	code	registered preliminary final amended + ObservationStatus (Required)
--------	------	------	------	------------------------------------------------------------------------------

```
/// <summary>  
/// Codes providing the status of an observation.  
/// (url: http://hl7.org/fhir/ValueSet/observation-status)  
/// </summary>  
public enum ObservationStatus {Registered, Preliminary, Final, ...}
```

```
var obs = new Observation();  
obs.Status = Observation.ObservationStatus.Preliminary;
```



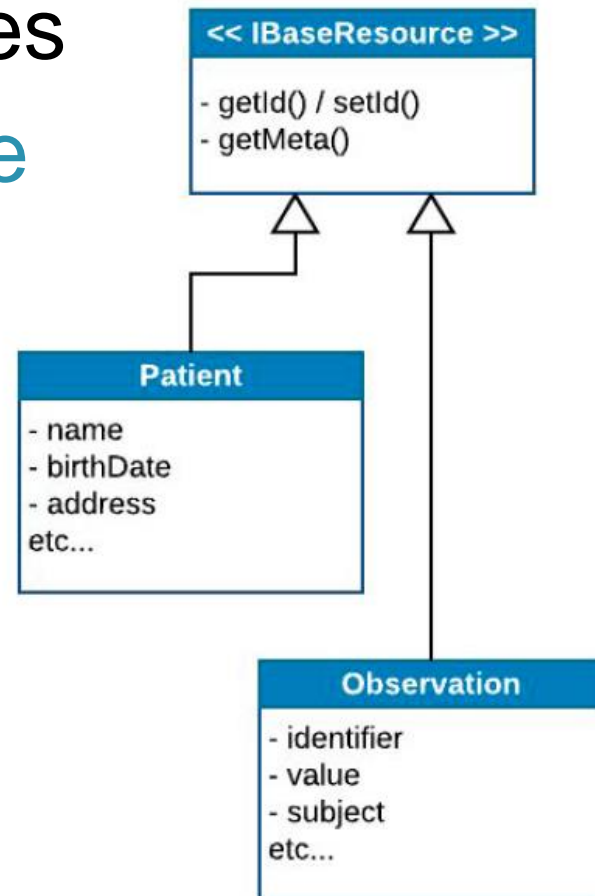
A FHIR Resource in Java



Resource definition classes implement **IBaseResource**

// Create a resource instance

```
Observation obs = new Observation();  
obs.setStatus(ObservationStatus.FINAL);
```



Datatypes (C#)



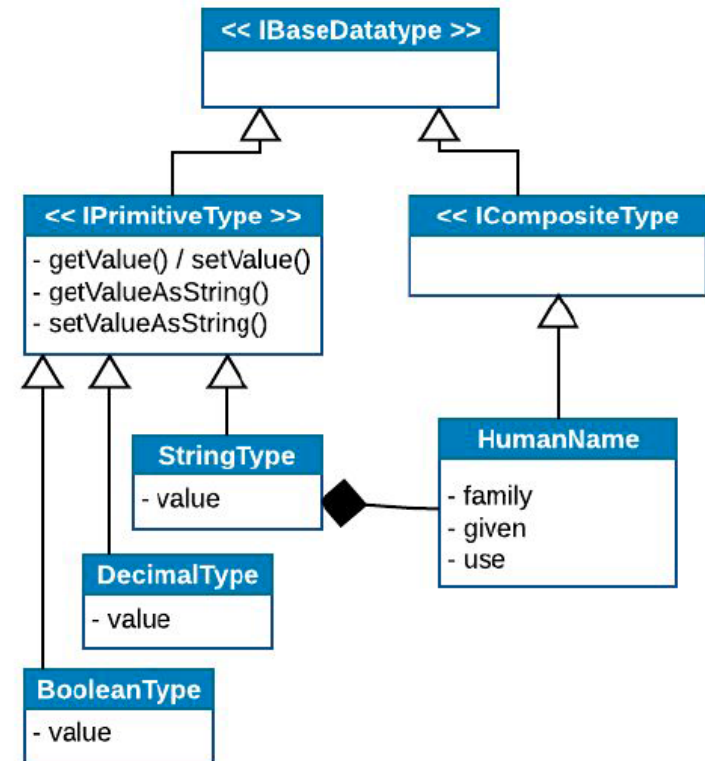
- Primitive classes are named FhirType
 - For datatypes with same name in C#
- Primitive types include:
FhirString, FhirBoolean
- Composite types include:
Address, Ratio, HumanName



Datatypes (Java)



- Primitive classes are named [name]Type
- Primitive types include: StringType, BooleanType
- Composite types include: Address, Ratio, HumanName



Primitives are not really primitive...



Patient (DomainResource)

identifier : Identifier [0..*]
active : boolean [0..1]

```
/// <summary>
```

```
/// Whether this patient's record is in active use
```

```
/// </summary>
```

```
public Hl7.Fhir.Model.FhirPatient
```

```
var pat = new Patient();
```

```
public bool? Active { get; set; }
```

```
pat.ActiveElement = new FhirBoolean(true);
```

```
pat.Active = true;
```

Patient

setActive(boolean value)

Patient

setActiveElement(BooleanType value)



A FHIR Resource




 code	Σ	1..1	CodeableConcept	Type of observation (code / type) LOINC Codes (Example)
----------------------------------------------------------------------------------------	---	------	-----------------	------------------------------------------------------------

```
public CodeableConcept Code { get; set; }
```

Observation

setCode(CodeableConcept value)

 identifier		0..*	Identifier	Unique Id for this particular observation
----------------------------------------------------------------------------------------------	--	------	------------	----------------------------------------------

```
public List<Identifier> Identifier{ get; set; }
```






List<Identifier>

getIdentifier()



A FHIR Resource in C#



 value[x]	Σ	0..1	Actual result
 valueQuantity			Quantity
 valueCodeableConcept			CodeableConcept
 valueString			string
 valueRange			Range

```
public Element Value { get; set; }
```



Observation

```
setValue(Type value)
```



A FHIR Resource in C#



 referenceRange	I	0..*	BackboneElement	Provides guide for interpretation <i>Must have at least a low or a high or text</i>
 low	T	0..1	SimpleQuantity	Low Range, if relevant

Observation (DomainResource)
identifier : Identifier [0..*] status : code [1..1] « ObservationStatus! » category : CodeableConcept [0..1] « Observation Category ?? » code : CodeableConcept [1..1] « LOINC ?? » subject : Reference [0..1] « Patient Group Device Location » encounter : Reference [0..1] « Encounter » effective[x] : Type [0..1] « dateTime Period »

ReferenceRange
low : Quantity(SimpleQuantity) [0..1] high : Quantity(SimpleQuantity) [0..1] meaning : CodeableConcept [0..1] « Observation Reference Range M...?? » age : Range [0..1] text : string [0..1]

[0..*]
referenceRange

[0..*]
referenceRange

```
public partial class ReferenceRangeComponent : BackboneElement
{ ... }
```

```
public static class Observation.ObservationReferenceRangeComponent
extends BackboneElement
implements IBaseBackboneElement
```



C# example



```
static void Main(string[] args)
{
    // Create a resource instance
    var pat = new Patient();

    // Add a "name" element
    var name = new HumanName();
    name = name.WithGiven("Homer").WithGiven("J.").AndFamily("Simpson");

    pat.Name.Add(name);

    // Add an "identifier" element
    var id = new Identifier("http://acme.org/MRNs", "7000135");
    pat.Identifier.Add(id);
}
```



Java example



```
1: public class Example01_CreateAPatient {
2:     public static void main(String[] theArgs) {
3:         // Create a resource instance
4:         Patient pat = new Patient();
5:
6:         // Add a "name" element
7:         HumanName name = pat.addName();
8:         name.addFamily("Simpson").addGiven("Homer").addGiven("J");
9:
10:        // Add an "identifier" element
11:        Identifier identifier = pat.addIdentifier();
12:        identifier.setSystem("http://acme.org/MRNs").setValue("7000135");
13:
14:        // Model is designed to be chained
15:        pat.addIdentifier().setSystem("http://acme.org/MRNs").setValue("12345");
16:    }
17: }
```





REST INTERACTIONS

Using the FHIR Client



- See Publicly Available FHIR Servers for available test servers

```
var client = new FhirClient("http://acme.org/fhir");
```

```
// client options
```

```
client.PreferredFormat = ResourceFormat.Xml;
```

```
client.ReturnFullResource = true;
```



Create example C#



```
var pat = new Patient();
pat.Name.Add(new HumanName()
    .WithGiven("Homer").WithGiven("J.").AndFamily("Simpson"));
pat.Identifier.Add(new Identifier("http://acme.org/MRNs", "7000135"));
pat.Gender = AdministrativeGender.Male;

// Create a client
var client = new FhirClient("http://vonk.furore.com");

// Use the client to store a new resource instance
var outcome = client.Create<Patient>(pat);

// Print the ID of the newly created resource
Console.WriteLine(outcome.Id);
```



Create Example Java



```
public static void main(String[] theArgs) {  
    Patient pat = new Patient();  
    pat.addName().addFamily("Simpson").addGiven("Homer").addGiven("J");  
    pat.addIdentifier().setSystem("http://acme.org/MRNs").setValue("7000135");  
    pat.setGender(AdministrativeGender.MALE);  
  
    // Create a context  
    FhirContext ctx = FhirContext.forDstu3();  
  
    // Create a client  
    String serverBaseUrl = "http://fhirtest.uhn.ca/baseDstu3";  
    IGenericClient client = ctx.newRestfulGenericClient(serverBaseUrl);  
  
    // Use the client to store a new resource instance  
    MethodOutcome outcome = client.create().resource(pat).execute();  
  
    // Print the ID of the newly created resource  
    System.out.println(outcome.getId());  
}
```

Read/update/delete example C#



```
var pat = client.Read<Patient>("Patient/1");

pat.Name.Add(new HumanName()
    .WithGiven("Ewout").AndFamily("Kramer"));

client.Update<Patient>(pat);

client.Delete(pat);
client.Delete("Patient/12345");
```



Read/Update example Java

// Use the client to read back the new instance using the ID

```
Patient patient = client.read(Patient.class, "190002");
```

// Change the gender and send an update to the server

```
patient.setGender(AdministrativeGender.FEMALE);
```

```
MethodOutcome outcome = client.update().resource(patient).execute();
```


Search example C#



```
var q = new SearchParams()  
    .Where("name=Ewout")  
    .Include("Patient:organization")  
    .LimitTo(10)  
    .SummaryOnly()  
    .OrderBy("birthdate",  
            Hl7.Fhir.Rest.SortOrder.Descending);  
  
q.Add("gender", "male");  
  
Bundle result = client.Search<Patient>(q);
```



Search example Java



```
public static void main(String[] theArgs) {
    FhirContext ctx = FhirContext.forDstu3();
    IGenericClient client = ctx.newRestfulGenericClient("http://fhirtest.uhn.ca/baseDstu3");

    // Log requests and responses (very verbose for testing!)
    client.registerInterceptor(new LoggingInterceptor(true));

    // Build a search and execute it
    Bundle response = client.search()
        .forResource(Patient.class)
        .where(Patient.NAME.matches().value("Test"))
        .and(Patient.BIRTHDATE.before().day("2014-01-01"))
        .count(100)
        .returnBundle(Bundle.class)
        .execute();

    // How many resources did we find?
    System.out.println("Responses: " + response.getTotal());

    // Print the ID of the first one
    System.out.println(response.getEntry().get(0).getResource().getId());
}
```

27

http://fhirtest.uhn.ca/baseDstu3/Patient/82599/_history/1

Paging through Bundle (C# example)



```
while (result != null)
{
    foreach (var e in result.Entry)
    {
        Patient p = (Patient)e.Resource;
        // do something with the resource
    }

    result = client.Continue(result, PageDirection.Next);
}
```





PARSING/SERIALIZING

Parsing/Serializing C#



```
// Create a file-based reader for JSON
JsonTextReader reader =
    new JsonTextReader(new StreamReader(@"input.json"));

var parser = new FhirJsonParser();
var new_obs = parser.Parse<Observation>(reader);

// Serialize an in-memory observation to a JSON string
var jsonText =
    FhirSerializer.SerializeResourceToJson(new_obs);
```



Parsing/Serializing Java



// Create a JSON parser

```
IParser parser = ctx.newJsonParser();
```

```
Patient pat = parser.parseResource(Patient.class, resourceBody);
```

```
List<Identifier> identifiers = pat.getIdentifier();
```

```
String idSystemString = identifiers.get(0).getSystem();
```

```
String idValueString = identifiers.get(0).getValue();
```

```
System.out.println(idSystemString + " " + idValueString);
```

```
parser.setPrettyPrint(true);
```

```
String encode = parser.encodeResourceToString(pat);
```

```
System.out.println(encode);
```

- Java examples and source:
<https://github.com/jamesagnew/hapi-fhir/>
- Docs:
<http://hapifhir.io/>
- C# source:
<https://github.com/ewoutkramer/fhir-net-api/>
- Starters:
<https://github.com/furore-fhir/fhirstarters>

Exercise



<https://github.com/furore-fhir/fhirstarters>

- NHS-event folder
 - NHS-exercise.pdf
 - Or try the non-coding exercises:
F054. REST_excercise.pdf
and
F074. Search_excercise.pdf

Events



FH'R Developer Days



Tutorials, guided hackathons, community track

15/17 Nov 2017

<http://www.fhirdevdays.com/>



The End – Questions?