FHIR for Developers

Rik Smithies

Updated for FHIR DSTU!
INTRODUCTION
Who am I?

- **Name:** Rik Smithies
- **Company:** Independent Consultant (NProgram Ltd., UK)
- **Background:**
  - Technical Committee Chair, HL7 UK (and former Chair)
  - HL7 International Co-Chair (Clinical Statement), Help Desk moderator, eLearning course tutor
  - Architect, analyst and software developer. 20 years in healthcare IT
- **Contact:**
  - rik@nprogram.co.uk
  - www.nprogram.co.uk
Introduce ourselves

- About your organization...
- HL7 (v2/v3) background?
- How did you hear about FHIR?
- Platform of choice (.NET, Java, Ruby, …)?
- Familiar with HTTP, XML, JSON, REST?
- Persistence technologies used?
Contents of this tutorial

- Deconstructing FHIR
- FHIR RESTful service interface
- Beyond REST
- Searching
- FHIR on the Wire
- Distribution for developers
- Building a FHIR server
What perspective?
Looking at FHIR data modeling concepts from a software engineering perspective

DECONSTRUCTING FHIR
“Resources” are:

- Small logically discrete units of exchange
- Defined behaviour and meaning
- Known identity / location
- Smallest unit of transaction
- “of interest” to healthcare

- V2: Sort of like Segments
- V3: Sort of like CMETs
Structure of a Resource

- Metadata
  - Resource
  - Narrative
  - Elements
  - Extensions
  - Extensions
FHIR makes composition and references explicit:

- **References** are in between Resources. **No context conduction across references** – safe retrieval as individual resources.
- **Composition** is within a Resource: Components have no meaning outside resource, no identity, no separate access path except through resource
Composition of a Resource

Resource Root

MedicationDispense (Resource)

- identifier : Identifier 0..1
- status : code 0..1 <<MedicationDispenseStatus>>
- patient : Resource(Patient) 0..1
- dispenser : Resource(Practitioner) 0..1
- authorizingPrescription : Resource(MedicationPrescription) 0..1

Resource Component

Dispense

- dispense : 0..

Simple & Complex elements (may be repeating)

Substitution

- type : CodeableConcept 1..1 <<MedicationIntendedSubstitutionType>>
- reason : CodeableConcept 0..* <<MedicationIntendedSubstitutionReason>>
- responsibleParty : Resource(Practitioner) 0..1
Composition

MedicationDispense (Resource)
- identifier : Identifier 0..1
- status : code 0..1 <<MedicationDispenseStatus>>
- patient : Resource(Patient) 0..1
- dispenser : Resource(Practitioner) 0..1
- authoringPrescription : Resource(MedicationPrescription) 0..*

Dispense
- dispense
  - status value="completed"/
  - quantity
    - <value value="10"/>
    - <units value="ml"/>
    - <system value="http://unitsofmeasure.org"/>
    - <code value="ml"/>
  - whenPrepared value="2012-05-30T16:20:00"/
  - whenHandedOver value="2012-05-31T10:20:00"/

Substitution
- type
  - <coding>
    - <system value="/MedDispSubType"/>
    - <code value="NoSub"/>
    - <display value="No substitution made or expected"/>
It’s all about combining resources . . .

http://lab.hospitalA.org/Observation/3ff27

http://lab.hospitalA.org/DiagnosticReport/4445

http://hospitalA.org/Practitioner/87

http://hospitalA.org/Organization/1

http://pat.registry.org/Patient/223
Example: part of DiagnosticReport

<!-- first, various administrative/context stuff -->
<status value="final"/> <!-- all this report is final -->
<issued value="2011-03-04T11:45:33+11:00"/>

<subject>
  <reference value="Patient/pat2"/>
</subject>

<performer>
  <reference value="Organization/1832473e-2fe0-452d-abe9-3cdb9879522f"/>
  <display value="Acme Laboratory, Inc"/>
</performer>

<identifier>
  <system value="http://acme.com/lab/reports"/>
  <value value="5234342"/>
</identifier>
“How do we know where an object made up of other objects begins and ends?”

“In any system with persistent storage of data, there must be a scope for a transaction that changes data and a way of maintaining the consistency of the data”
“Business” identifiers

```yaml
DiagnosticReport (Resource)
name : CodeableConcept 1..1 <<DiagnosticReportNames>>
status : code 1..1 <<DiagnosticReportStatus>>
issued : dateTime 1..1
subject : Resource(Patient | Group | Device | Location) 1..1
performer : Resource(Practitioner | Organization) 1..1
identifier : Identifier 0..1
```

```yaml
Patient (Resource)
identifier : Identifier 0..*
name : HumanName 0..1
telecom : Contact 0..*
gender : CodeableConcept 0..1 <<AdministrativeGender>>
birthDate : dateTime 0..1
deceased[x] : boolean | dateTime 0..1
address : Address 0..*
maritalStatus : CodeableConcept 0..1 <<MaritalStatus>>
multipleBirth[x] : boolean | integer 0..1
photo : Attachment 0..*
communication : CodeableConcept 0..* <<Language>>
careProvider : Resource(Organization | Practitioner) 0..*
managingOrganization : Resource(Organization) 0..1
active : boolean 0..1
```

© 2014 HL7 © International. Licensed under Creative Commons. HL7 & Health Level Seven are registered trademarks of Health Level Seven International. Reg. U.S. TM Office.
A Resource’s identity

- In fact: an URL
  
  ![Diagram](http://server.org/fhir/Patient/1)

  - resource type
  - endpoint
  - identifier

Note: This URL resolves to the current version of a resource
Resource metadata

Metadata

Patient
MRN 22234
“Ewout Kramer”
30-11-1972
Amsterdam

Resource Identities
http://fhir.hl7.org/Patient/23E455A3B
http://fhir.hl7.org/Patient/23E455A3B/_history/4

Last updated
2013-12-23T23:33:01+01:00

http://hl7.org/fhir/tag
http://example.org/fhir/Status#Test
http://hl7.org/fhir/tag/profile
http://hl7.org/fhir/Profile/us-core
The FHIR Elements

- Composite Datatypes
  - (HumanName, Quantity, Period, Address, Identifier)

- Constrained Types
  - (Quantity: Distance, Count, Duration, Money)

- Primitives
  - (integer, boolean, string, instant)

- Derived Primitives
  - (oid, uuid, code, id)
## Start at the bottom: Primitives

<table>
<thead>
<tr>
<th>Type</th>
<th>xs:</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>boolean</td>
<td>Values can be either true or false</td>
</tr>
<tr>
<td>integer</td>
<td>int</td>
<td>A signed 32-bit integer</td>
</tr>
<tr>
<td>decimal</td>
<td>decimal</td>
<td>A rational number. <strong>A true decimal</strong>, with inbuilt precision (e.g. Java BigDecimal)</td>
</tr>
<tr>
<td>base64Binary</td>
<td>base64Binary</td>
<td>A stream of bytes, base64 encoded</td>
</tr>
<tr>
<td>instant</td>
<td>dateTime</td>
<td>An instant in time - <strong>known at least to the second and always includes a timezone.</strong></td>
</tr>
<tr>
<td>string</td>
<td>string</td>
<td>A sequence of <strong>Unicode</strong> characters.</td>
</tr>
<tr>
<td>uri</td>
<td>anyURI</td>
<td>A Uniform Resource Identifier Reference.</td>
</tr>
<tr>
<td>date</td>
<td>union of date, gYearMonth, gYear</td>
<td>A date, or <strong>partial date</strong> as used in human communication. No time zone.</td>
</tr>
<tr>
<td>dateTime</td>
<td>union of dateTime, date, gYearMonth, gYear</td>
<td>A date, date-time or <strong>partial date</strong> as used in human communication. If hours and minutes are specified, a <strong>time zone must be populated.</strong></td>
</tr>
</tbody>
</table>
Derived primitives

- Using the ISO date/time with timezone
  - “1951”, “1951-06” and “1951-06-04”
  - “1951-06-04T10:57:34.0321+01”
  - “1951-06-04T10:57:34.0321Z”
Derived primitives

- Based on uri(!): OID and UUID
  - urn:oid:1.2.3.4.5
  - urn:uuid:a5afddf4-e880-459b-876e-e4591b0acc11

- Based on string:
  - code (string of characters, may contain single spaces) - “4548-4”, “active”, “not known”
  - id ([a-z0-9\-\.]{1,36})
Level up: Composite Datatypes

```
<time>
    <value value="25"/>
    <units value="sec"/>
    <system value="http://unitsofmeasure.org"/>
    <code value="s"/>
</time>
```
Let’s take a look at the “Data Types” section of the FHIR specification at

http://www.hl7.org/implement/standards/fhir/datatypes.html
Coded types

Codes are defined in code systems

```xml
<problem>
  <system value="http://snomed.info/id" />
  <code value="128045006:363698007=56459004" />
</problem>
```
Coded types

- When used in a Resource, the modelers include **Bindings**

  **Bindings** specify which codes can be used

  4.15.3.1 Terminology Bindings

<table>
<thead>
<tr>
<th>Path</th>
<th>Definition</th>
<th>Type</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observation.name</td>
<td>Codes identifying types of simple observations</td>
<td>Example</td>
<td><a href="http://hl7.org/fhir/vs/observation-codes">http://hl7.org/fhir/vs/observation-codes</a></td>
</tr>
<tr>
<td>Observation.interpretation</td>
<td>Codes identifying interpretations of observations</td>
<td>Incomplete</td>
<td><a href="http://hl7.org/fhir/vs/observation-interpretation">http://hl7.org/fhir/vs/observation-interpretation</a></td>
</tr>
<tr>
<td>Observation.status</td>
<td>Codes providing the status of an observation</td>
<td>Fixed</td>
<td><a href="http://hl7.org/fhir/observation-status">http://hl7.org/fhir/observation-status</a></td>
</tr>
<tr>
<td>Observation.reliability</td>
<td>Codes that provide an estimate of the degree to which quality issues have impacted on the value of an observation</td>
<td>Fixed</td>
<td><a href="http://hl7.org/fhir/observation-reliability">http://hl7.org/fhir/observation-reliability</a></td>
</tr>
<tr>
<td>Observation.bodySite</td>
<td>Codes describing anatomical locations. May include laterality</td>
<td>Example</td>
<td><a href="http://hl7.org/fhir/vs/body-site">http://hl7.org/fhir/vs/body-site</a></td>
</tr>
<tr>
<td>Observation.method</td>
<td>Methods for simple observations</td>
<td>Example</td>
<td><a href="http://hl7.org/fhir/vs/observation-methods">http://hl7.org/fhir/vs/observation-methods</a></td>
</tr>
<tr>
<td>Observation.referenceRange meaning</td>
<td>Code for the meaning of a reference range</td>
<td>Example</td>
<td><a href="http://hl7.org/fhir/vs/referencerange-meaning">http://hl7.org/fhir/vs/referencerange-meaning</a></td>
</tr>
<tr>
<td>Observation.related.type</td>
<td>Codes specifying how two observations are related</td>
<td>Fixed</td>
<td><a href="http://hl7.org/fhir/observation-relationshiptypes">http://hl7.org/fhir/observation-relationshiptypes</a></td>
</tr>
</tbody>
</table>
Value Set http://hl7.org/fhir/vs/administrative-gender

The gender of a person used for administrative purposes

Formal definitions: XML (for browser) or JSON.

Administrative Gender Codes

This value set defines the set of codes that can be used to indicate the administrative gender of a person.

This value set includes codes defined in other code systems, using the following rules:

- Include these codes as defined in http://hl7.org/fhir/v3/AdministrativeGender

<table>
<thead>
<tr>
<th>Code</th>
<th>Display</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>Female</td>
<td>Female</td>
</tr>
<tr>
<td>M</td>
<td>Male</td>
<td>Male</td>
</tr>
<tr>
<td>UN</td>
<td>Undifferentiated</td>
<td>The gender of a person could not be uniquely defined as male or female, such as hermaphrodite.</td>
</tr>
</tbody>
</table>

- Include these codes as defined in http://hl7.org/fhir/v3/NullFlavor

<table>
<thead>
<tr>
<th>Code</th>
<th>Display</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNK</td>
<td>unknown</td>
<td>Description: A proper value is applicable, but not known. Usage Notes: This means the actual value is not known. If the only thing that is unknown is how to properly express the value in the necessary constraints (value set, datatype, etc.), then the OTH or UNC flavor</td>
</tr>
</tbody>
</table>
Go to some interesting value sets to look at them
- Publication metadata
- Concepts from 1 or more existing systems
- Additional concepts
Level up: resources

```xml
<Organization>
  <!-- Clinical Team "Gastroenterology" at Acme -->
  <identifier>
    <id value="Gastro"/>
  </identifier>
  <name value="Gastroenterology"/>
  <telecom>
    <system value="phone"/>
    <value value="+1 555 234 3523"/>
    <use value="mobile"/>
  </telecom>
  <telecom>
    <system value="email"/>
    <value value="gastro@acme.org"/>
    <use value="work"/>
  </telecom>
</Organization>
```
"Choice" properties

```xml
<Observation>
  <valueQuantity>
    <value value="107"/>
    <units value="mm[Hg]"/>
  </valueQuantity>
</Observation>

<Observation>
  <valueString
    value="Patient loves to sing"/>
</Observation>
```
<DiagnosticReport>

<subject>
  <reference value="Patient/pat2"/>
</subject>

<performer>
  <reference value="Organization/1832473e-2fe0-452d-abe9-3cdb9879522f"/>
  <display value="Acme Laboratory, Inc"/>
</performer>

</DiagnosticReport>
Quick look at extensions

Metadata

Resource

Narrative

Extensions

Elements

Extensions
Extensions

Patient
MRN 22234
“Ewout Kramer”
30-11-1972
Amsterdam

+ Haircolor BROWN

Organization
“ACME Hospital”
National Drive 322
Orlando, FL

+ Taxoffice Id NLOB33233

You can extend:
- Resources
- Elements of Resources
- FHIR Datatypes
Extending a multiple birth

Key = location of formal definition

Value = value according to definition

```xml
<Patient xmlns="http://hl7.org/fhir">
  <!-- stuff -->
  <multipleBirthBoolean value="true">
    <extension url="http://hl7.org/fhir/Profile/us-core#birthorder">
      <valueDecimal value="2" />
    </extension>
  </multipleBirthBoolean>
  <!-- more stuff -->
</Patient>
```
Complex extensions

```xml
<Patient>
  <extension url="http://acme.org/fhir/Profile/main#trial-status" >
    <extension url="http://acme.org/fhir/Profile/main#trial-status-code" >
      <valueCode value="unsure" />
    </extension>
  </extension>
  <extension url="http://acme.org/fhir/Profile/main#trial-status-date" >
    <valueDate value="2009-03-14" />
  </extension>
  <extension url="http://acme.org/fhir/Profile/main#trial-status-who" >
    <valueResource>
      <reference value="Practitioner/example" />
    </valueResource>
  </extension>
</extension>
<!-- other data for patient -->
</Patient>
```
Quick look at narrative
<DiagnosticReport xmlns="http://hl7.org/fhir">
  <text>
    <status value="generated"/>
    <div xmlns="http://www.w3.org/1999/xhtml">
      <h3>CBC Report for Wile. E. COYOTE (MRN: 23453) issued 3-Mar 2011 11:45</h3>
      <pre>
        | Test           | Units | Value | Reference Range |
        |----------------|-------|-------|-----------------|
        | Haemoglobin    | g/L   | 176   | 135 - 180       |
        | Red Cell Count | x10*12/L | 5.9 | 4.2 - 6.0       |
        | Haematocrit    |       | 0.55+ | 0.38 - 0.52     |
        | Mean Cell Volume| fL  | 99+   | 90 - 98         |
        | Mean Cell Haemoglobin| pg | 36+ | 27 - 35         |
        | Platelet Count | x10*9/L | 444 | 150 - 450       |
        | White Cell Count| x10*9/L | 4.6 | 4.0 - 11.0      |
      </pre>
      <p>Acme Laboratory, Inc signed: Dr Pete Pathologist</p>
    </div>
  </text>
  <status value="final"/> <!-- all this report is final -->
  <issued value="2011-03-04T11:45:33+11:00"/>
</DiagnosticReport>
How FHIR uses RESTful principles to communicate Resources

REST SERVICE INTERFACE
Paradigms

FHIR supports 4 interoperability paradigms

- REST
- Documents
- Messages
- Services
REST?

“REpresentational State Transfer”
- Represent your data as “resources”
- Make “Resources” URI addressable
- Use HTTP to do CRUD operations
- Resources may be exchanged using different representations
Possibly distributed...

- **FHIR server @ pat.registry.org**
  - Patient/223
  - Patient

- **FHIR server @ lab.hospitalA.org**
  - DiagnosticReport/4445
  - Observation/3ff27
  - Observation

- **FHIR server @ hospitalA.org**
  - Organization/1
    - Organization
  - Practitioner/87
    - Practitioner

Managing

Performer
http://fhirblog.com/2014/01/24/modelling-encounters-with-fhir/
“Repository” model of healthcare

Hospital System
- Create
- Update
- Query

Lab System
- Create
- Update

FHIR server

Patient

Diagnostic Report

Observation

© 2014 HL7 © International. Licensed under Creative Commons. HL7 & Health Level Seven are registered trademarks of Health Level Seven International. Reg. U.S. TM Office.
Just a quick GET

GET /fhir/Patient/1 HTTP/1.1

HTTP/1.1 200 OK
Content-Type: application/xml+fhir; charset=utf-8
Content-Length: 787
Content-Location:
http://fhir.furore.com/fhir/Patient/1/_history/1
Last-Modified: Tue, 29 May 2012 23:45:32 GMT

<?xml version="1.0" encoding="UTF-8"?>
A Resource’s REST identity

- In fact: an URL

  - http://server.org/fhir/Patient/1

  Note: This URL resolves to the current version of a resource
Remember metadata?

**Metadata**

**Patient**

MRN 22234  
"Ewout Kramer"  
30-11-1972  
Amsterdam

**Resource Identities**

http://fhir.hl7.org/Patient/23E455A3B  
http://fhir.hl7.org/Patient/23E455A3B/_history/4

Last updated  
2013-12-23T23:33:01+01:00

http://hl7.org/fhir/tag  
http://example.org/fhir/Status#Test  
http://hl7.org/fhir/tag/profile  
http://hl7.org/fhir/Profile/us-core
Tag metadata

GET /fhir/Patient/1 HTTP/1.1

HTTP/1.1 200 OK
Content-Location: http://sever.om/fhir/Patient/1/_history/12
Last-Modified: Tue, 29 May 2012 23:45:32 GMT
Category: http://example.org/fhir/Status#Test; scheme=" http://hl7.org/fhir/tag"; label="Our test tag"

<table>
<thead>
<tr>
<th><a href="http://hl7.org/fhir/tag">http://hl7.org/fhir/tag</a></th>
<th>A general tag</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://hl7.org/fhir/tag/profile">http://hl7.org/fhir/tag/profile</a></td>
<td>A profile tag - a claim that the Resource conforms to the profile identified in the term</td>
</tr>
<tr>
<td><a href="http://hl7.org/fhir/tag/security">http://hl7.org/fhir/tag/security</a></td>
<td>A security label</td>
</tr>
</tbody>
</table>

© 2014 HL7 ® International. Licensed under Creative Commons. HL7 & Health Level Seven are registered trademarks of Health Level Seven International. Reg. U.S. TM Office.
Mapping (meta)data to HTTP

- Resource data
- Resource id
- Resource version
- Last update date
- Tags

- http body
- Url
- Content-Location header
- Last-Modified header
- Category header

© 2014 HL7 ® International. Licensed under Creative Commons. HL7 & Health Level Seven are registered trademarks of Health Level Seven International. Reg. U.S. TM Office.
One more look at the header

GET /fhir/Patient/1 HTTP/1.1

HTTP/1.1 200 OK
Content-Type: application/xml+fhir; charset=utf-8
Content-Length: 787
Content-Location: http://fhir.furore.com/fhir/Patient/1/_history/12
Last-Modified: Tue, 29 May 2012 23:45:32 GMT
For a specific version...

- We have the version-specific URL

```
http://server.org/fhir/ (continued)
```

```
Patient/1/_history/4
```

- base path
- resource type
- identifier
- version id
Support for versions

 abyss, v12 – 2012-12-04

 abyss, v13 – 2012-12-05

 abyss, v14 – 2012-12-08

 abyss, v15 – 2012-12-09

 /server.org/fhir/Patient/33/_history/12

 /server.org/fhir/Patient/33/_history/13

 /server.org/fhir/Patient/33/_history/14

 /server.org/fhir/Patient/33/_history/15

 /server.org/fhir/Patient/33
REST “representations”

GET /fhir/Patient/1?_format=json HTTP/1.1

HTTP/1.1 200 OK
Content-Type: application/json+fhir;charset=utf-8
Content-Length: 787

GET /fhir/Patient/1 HTTP/1.1
Accept: application/json+fhir

HTTP/1.1 200 OK
Content-Type: application/json+fhir;charset=utf-8
Content-Length: 787
Question

DO I REALLY HAVE TO IMPLEMENT VERSIONS?

NO, You are not required to keep history, and may return 410 (Gone) on a “vread” for any request for an older version than the current one!

DO I REALLY NEED TO SUPPORT THAT PRE-HISTORIC XML STUFF?
Conformance

• Which FHIR version?
• Which Resources?
• What search operations?
• What formats?
• Is this a test server?

• Who can I contact?
• What’s the name of the software?

• DO YOU SUPPORT HISTORY?
• DO YOU SUPPORT XML/JSON?

http://www.hl7.org/fhir/conformance.htm
REST in the spec

Let’s look at these operations in the specification....
Mapping to verbs

**create 2.1.10**
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 [service-url]/[resourcetype] (?_format=mimeType)
```

**read 2.1.6**
The read interaction accesses the current contents of a resource. The interaction is performed by an HTTP GET operation as shown:

```
GET [service-url]/[resourcetype]/{id} (?_format=mimeType)
```

**update 2.1.8**
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 [service-url]/[resourcetype]/{id} (?_format=mimeType)
```

**delete 2.1.9**
The delete interaction removes an existing resource. The interaction is performed by an HTTP DELETE operation as shown:

```
DELETE [service-url]/[resourcetype]/{id}
```
To create a resource

- You **POST** the contents to an url which indicates the resource type:
  - E.g. http://server.org/fhir/Patient

- Supply body’s format in **Content-Type** header

- Server returns **201 (Created)**.

- Returns only the newly assigned version id URL in the **Location** header.
To update a resource

- Use **PUT** on the resource’s URL, with the new contents in the body
- Tell server the body’s format (xml/json) in the **Content-Type** header
- Server returns 200 and the URL to new version in the **Content-Location** header.
Using PUT to create

- Server might/might not allow you to PUT to an id that does not yet exist.
- If it does: Server returns 201 and resource gets created at that location → *client determines resource’s id!*
- If it does not: server returns 405 (Method not allowed)
Version-aware updates

- Server requires client to send Content-Location header with a version-specific URL
- Server uses this to check whether you are updating the latest version.
- Server will then return 409 (Conflict) if it has been updated by someone else in the meantime
What’s a ‘deleted’ Resource?

- Trying read operations will return in a 410 (Gone) result in stead of 404 (Not Found)
- The resource will not be returned by the search operation.
- You can “undelete” by doing an update with fresh content
- Just a “marker” in a resource’s history
Version history - deletions

33, v12 – 2012-12-04
/server.org/fhir/Patient/33/_history/12

33, v13 – 2012-12-05
/server.org/fhir/Patient/33/_history/13

33, v14 – 2012-12-08
/server.org/fhir/Patient/33/_history/14

33, v15 – 2012-12-09
/server.org/fhir/Patient/33/_history/15

33, v16 – 2012-12-10
/server.org/fhir/Patient/33/_history/16

DELETION

/server.org/fhir/Patient/33

© 2014 HL7 ® International. Licensed under Creative Commons. HL7 & Health Level Seven are registered trademarks of Health Level Seven International. Reg. U.S. TM Office.
Version history - revival

/server.org/fhir/Patient/33/_history/13
33, v13 – 2012-12-05

/server.org/fhir/Patient/33/_history/14
33, v14 – 2012-12-08

/server.org/fhir/Patient/33/_history/15
33, v15 – 2012-12-09

/server.org/fhir/Patient/33/_history/16
33, v16 – 2012-12-10

/server.org/fhir/Patient/33/_history/17
33, v17 – 2012-12-11

/server.org/fhir/Patient/33
How resources are made into classes in the supplied reference implementations

RESOURCES IN CODE
Reference implementations

- Contents
  - Model – classes generated from the spec
  - Parsers – Parsers generated from the spec
  - Serializers – Serializers generated from the spec
  - FhirClient
  - Validation (currently Java only)

- Java – Everything on the downloads page
- .NET – NuGet “FHIR”, or GitHub “fhir-net-api”
[FhirResource("DiagnosticReport")]

public partial class DiagnosticReport : Resource {

    public Code<ObservationStatus> Status {...}

    public Instant Issued {...}

    public ResourceReference Subject {...}

    public ResourceReference Performer {...}

    public Identifier ReportId {...}

    public List<DiagnosticReportRequestDetailComponent>
        RequestDetail { ...}

    requestDetail
// Create a file-based reader for Xml
XmlReader xr = XmlReader.Create(
    new StreamReader(@"publish\observation-example.xml");

// Parse the Observation from the stream
var obs = (Observation)FhirParser.ParseResource(xr);

// Modify some fields of the observation
obs.Status = Observation.ObservationStatus.Amended;
obs.Value = new Quantity() { Value = 40, Units = "g" };  

// Serialize the in-memory observation to Json
var jsonText = FhirSerializer.SerializeResourceToJson(obs);
XmlParser xml = new XmlParser();

Observation obs = (Observation)xml.parse(new FileInputStream("observation.xml"));

obs.setStatusSimple(ObservationStatus.amended);

Quantity newValue = new Quantity();
newValue.setValueSimple(new BigDecimal(40));
newValue.setUnitsSimple("g");
obs.setValue( newValue );

ByteArrayOutputStream bos = new ByteArrayOutputStream();
JsonComposer comp = new JsonComposer();
comp.compose(bos, obs, true);
String json = bos.toString("UTF8");
Using FHIR Client in C#

```csharp
var client = new FhirClient(
    new Uri("http://fhir.com/svc/fhir"));

var patEntry = client.Read<Patient>("1");
var pat = patEntry.Resource;
var restId = patEntry.Id;
var tags = patEntry.Tags;

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

client.Update<Patient>(patEntry);
```
Using FHIR Client in Java

FHIRClient client = new FHIRExampleClient();
client.initialize("http://spark.furore.com/fhir");

AtomEntry<Patient> pe = client.read(Patient.class, "1");
Patient p = pe.getResource();

HumanName hn = new HumanName();
hn.getFamily().add(Factory.newString_("Kramer"));
hn.getGiven().add(Factory.newString_("Ewout"));
p.getName().add(hn);

client.update(Patient.class, p, "1");
How FHIR uses Atom to communicate sets of resources

BUNDLES
Communicating lists

- We need to communicate lists of Resources
  - Search result
  - History
  - Documents or messages
  - Multiple-resource inserts ("batches")

- So, we need an industry-standard to represent lists, and a place to put our metadata
Bundles

- Atom RFC 4287 + Tombstones RFC 6721

- Poll-based protocol for keeping up-to-date with newsfeeds (RSS and Atom)

- You can “subscribe” to a FHIR feed and get updates
New reports in the mail
<feed xmlns="http://www.w3.org/2005/Atom">
  <title>Search on 'Patient'</title>
  <id>urn:uuid:1df558f6-0e10-452f-b0a4-e60ddac91211</id>
  <updated>2014-01-20T13:00:21.5416894Z</updated>
  <author>
    <name>Spark Search Engine</name>
  </author>
  <totalResults>598</totalResults>
  <link rel="next" href="http://spark.furore.com/fhir/_snapshot">
    <link rel="fhir-base" href="http://spark.furore.com/fhir"/>
  </link>
  <entry>
    <title type="text">Patient 3212416</title>
  </entry>
  <entry>
    <!--Etcetera-->
  </entry>
</feed>
Resource metadata

Metadata

Patient
MRN 22234
“Ewout Kramer”
30-11-1972
Amsterdam

Resource Identity
http://fhir.hl7.org/Patient/23E455A3B

Last updated
2013-12-23T23:33:01+01:00

http://hl7.org/fhir/tag
http://example.org/fhir/Status#Test
http://hl7.org/fhir/tag/profile
http://hl7.org/fhir/Profile/us-core
<entry>
  <title type="text">Patient resource with id 3212416</title>
  <id>http://hl7.org/fhir/Patient/3212416</id>
  <updated>2014-01-18T19:48:05.7634661Z</updated>
  <category term="http://hl7.org/fhir/Profile/us-core"
    scheme="http://hl7.org/fhir/tag/profile"/>
  <author>
    <name>(unauthenticated)</name>
  </author>
  <link rel="self" href="http://hl7.org/Patient/3212416/_history/2406"/>
  <content type="text/xml">
    <Patient xmlns="http://hl7.org/fhir">
      <Patient xmlns="http://www.w3.org/1999/xhtml">Patient HEATHER CHEN (00
        Address: 4491 HILL HAVEN DRIVE, NEW ORLEANS LA 70334</div>
      </Patient>
    </content>
  <summary type="xhtml">
  <div xmlns="http://www.w3.org/1999/xhtml">Patient HEATHER CHEN (00
    Address: 4491 HILL HAVEN DRIVE, NEW ORLEANS LA 70334</div>
  </summary>
</entry>
Multiple versions of entries

```xml
<entry>
  <title type="text">Patient 3</title>
  <id>http://hl7.org/fhir/Patient/3</id>
  <updated>2014-01-18T19:48:05.7634661Z</updated>
  <link rel="self" href="http://hl7.org/Patient/3/_history/40"/>
</entry>

<entry>
  <title type="text">Patient 3</title>
  <id>http://hl7.org/fhir/Patient/3</id>
  <updated>2014-01-19T10:18:25.3466121Z</updated>
  <link rel="self" href="http://hl7.org/Patient/3/_history/48"/>
</entry>
```

Same id!
Atom Tombstones - Deletions

```xml
<feed xmlns="http://www.w3.org/2005/Atom">
  <title>History for Patient 1</title>
  <id>urn:uuid:9f395ee0-19d2-4760-baf6-097fda52d914</id>
  <link href="http://server.org/Patient/1/_history" rel="self"/>
  <at:deleted-entry xmlns:at="http://purl.org/atompub/tombstones/1.0"
    ref="http://server.org/fhir/Patient/1" when="2012-06-19T11:27:12Z">
    <link rel="self" href="http://server.org/fhir/Patient/1/_history/2"/>
  </at:deleted-entry>

  <entry>
    <title>Patient resource with id 1</title>
    <link href="http://server.org/Patient/1/_history/1" rel="self"/>
    <id>http://server.org/fhir/Patient/1</id>
    <updated>2012-06-04T11:23:56Z</updated>
  </entry>

</feed>
```
Atom in JSON

- There’s not yet a way to render Atom in JSON, though there are initiatives, all ugly.
- So, we had to (sorry) roll our own…
- …very straightforward, single-purpose
- Atom JSON solution
- (Note: MIME type is still application/json!)
```json
{
    "title": "Search result",
    "updated": "2012-09-20T12:04:45Z",
    "id": "urn:uuid:50ea3e5e-b6a7-4f55-956c-caef491bbc08",
    "link": [ { "rel": "self", "href": "http://server.org/fhir/Patient?format=json" } ],
    "entry": [
        { "title": "Resource of type Patient, with id = 1 and version = 1",
          "link": [ { "rel": "self", "href": "http://server.org/fhir/Patient/1/_history/1" } ],
          "id": "http://fhir.furore.com/fhir/Patient/1",
          "updated": "2012-05-29T23:45:32Z",
          "published": "2012-09-20T12:04:47Z",
          "author": [ { "name": "Grahame Grieve / HL7 publishing committee" } ],
          "content":
            { "Patient": {} }
        }
    ]
}
```
Bundles

- For both Java and C#, reference has custom-built Atom parser
- For .NET, you *could* use the framework’s SyndicationFeed
  - A bit more low-level
  - No support for deleted-entries (even parse problems)
  - Incompatible with WinRT (Win8 mobile apps)
Bundles in C#

- Abstraction on top of Atom parser
- Bundle = feed, BundleEntry = entry.

```csharp
Bundle result = new Bundle() { Title = "Demo bundle" };

result.Entries.Add(new ResourceEntry<Patient>()
    { LastUpdated=DateTimeOffset.Now, Content = new Patient() });
result.Entries.Add(new DeletedEntry()
    { Id = new Uri("http://..."), When = DateTime.Now });

var bundleXml = FhirSerializer.SerializeBundleToXml(result);
```
Bundles in Java

```java
AtomFeed feed = new AtomFeed();
feed.setTitle("Demo bundle");

AtomEntry pat = new AtomEntry();
pat.setUpdated(Calendar.getInstance());
pat.setResource(new Patient());
feed.getEntryList().add(pat);

AtomEntry del = new AtomEntry();
del.setUpdated(Calendar.getInstance());
del.setDeleted(true); del.setId("http://nu.nl/fhir");
feed.getEntryList().add(del);

ByteArrayOutputStream bos = new ByteArrayOutputStream();
AtomComposer comp = new AtomComposer();
comp.compose(bos, feed, true);
String xml = bos.toString("UTF8");
```
Example: Keeping in sync

- History of **all resources** on server
  - http://server.org/fhir/_history

- History of **all patient resources** on server
  - http://server.org/fhir/Patient/_history

- History of **specific patient** on server
  - http://server.org/fhir/Patient/1/_history

- A history of all changes: updates and deletions, ordered by newest first

- Limit with `_since` and `_count`
And finally, the last REST operation (for now):
SEARCH FUNCTIONALITY
Getting “all” patients

- http://server.org/fhir/Patient

- Always returns a paged feed

- Use _count to indicate number of results per page

- Special case of the “real” search operation:

  http://server.org/fhir/Patient/_search?name=eve
  http://server.org/fhir/Patient?name=eve
Each resource has a set of “standard” search operations, so **not every element can be searched!**:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>Whether the patient record is active</td>
</tr>
<tr>
<td>address</td>
<td>an address in any kind of address/part of the patient</td>
</tr>
<tr>
<td>animal-breed</td>
<td>the breed for animal patients</td>
</tr>
<tr>
<td>animal-species</td>
<td>the species for animal patients</td>
</tr>
<tr>
<td>birthdate</td>
<td>the patient's date of birth</td>
</tr>
<tr>
<td>family</td>
<td>a portion of the family name of the patient</td>
</tr>
<tr>
<td>gender</td>
<td>gender of the patient</td>
</tr>
<tr>
<td>given</td>
<td>a portion of the given name of the patient</td>
</tr>
<tr>
<td>identifier</td>
<td>A patient identifier</td>
</tr>
<tr>
<td>language</td>
<td>language code (irrespective of use value)</td>
</tr>
<tr>
<td>name</td>
<td>a portion of either family or given name of the patient</td>
</tr>
</tbody>
</table>

Our last search used this one
Combining parameters

- Specifying multiple parameters finds resources matching all params ➔ “AND”
- Parameters may list multiple values ➔ “OR”

http://server.org/fhir/Patient/search?
birthdate=1972-11-30
&language=NL,FR
Each search parameter has a ‘type’

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>Whether the patient record is active</td>
</tr>
<tr>
<td>address</td>
<td>an address in any kind of address/part of the patient</td>
</tr>
<tr>
<td>animal-breed</td>
<td>the breed for animal patients</td>
</tr>
<tr>
<td>animal-species</td>
<td>the species for animal patients</td>
</tr>
<tr>
<td>birthdate</td>
<td>the patient’s date of birth</td>
</tr>
<tr>
<td>family</td>
<td>a portion of the family name of the patient</td>
</tr>
<tr>
<td>gender</td>
<td>gender of the patient</td>
</tr>
<tr>
<td>given</td>
<td>a portion of the given name of the patient</td>
</tr>
<tr>
<td>identifier</td>
<td>A patient identifier</td>
</tr>
<tr>
<td>language</td>
<td>language code (inherited from the given name of the patient)</td>
</tr>
<tr>
<td>name</td>
<td>a portion of either the family name or the given name of the patient</td>
</tr>
</tbody>
</table>
Ok I get it...or not?

<table>
<thead>
<tr>
<th>URL</th>
<th>Hits</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://server.org/fhir/Patient/">http://server.org/fhir/Patient/</a></td>
<td>406</td>
</tr>
<tr>
<td><a href="http://server.org/fhir/Patient?gender=M">http://server.org/fhir/Patient?gender=M</a></td>
<td>234</td>
</tr>
<tr>
<td><a href="http://server.org/fhir/Patient?gender=F">http://server.org/fhir/Patient?gender=F</a></td>
<td>167</td>
</tr>
</tbody>
</table>

**Total:** $234 + 167 = 401$

<table>
<thead>
<tr>
<th>URL</th>
<th>Hits</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://server.org/fhir/Patient/">http://server.org/fhir/Patient/</a></td>
<td>406</td>
</tr>
<tr>
<td><a href="http://server.org/fhir/Patient?gender=M">http://server.org/fhir/Patient?gender=M</a></td>
<td>234</td>
</tr>
<tr>
<td><a href="http://server.org/fhir/Patient?gender=F">http://server.org/fhir/Patient?gender=F</a></td>
<td>167</td>
</tr>
</tbody>
</table>

**Total:** $234 + 167 + 5 = 406
Chained searches

- Patient has a search for “name”.
- Observation has a search for “subject” (the id of the Patient, Group or Device)
- How do I find Observations for a patient, searching using his name?
2 queries in 1

- You (as a client) don’t need to do separate operations, just one:

  http://server.com/fhir/Observation/_search?
  subject.name=jim

But note: this still only works on the **predefined search parameters**. You cannot just use any property of the resource.
More optimizations

- Say we do:
  - We get back: a Bundle with 0..* “Observations”

- Now, usually, wouldn’t we want the Patient information too?  => Need to do “N” queries for the Observation’s “subject”

- Quicker:
  
  `_include=Observation.subject`

  Returns both Observations + Patients
How FHIR supports messages and documents

BEYOND REST
FHIR supports 4 interoperability paradigms

- REST
- Documents
- Messages
- Services
Regardless of **paradigm**
the content **is the same**

Receive a lab result in a message…

…Package it in a discharge summary document...
The Document resource

A single Resource, very often a List
Documents – are bundles

Composition Resource
Section
Metadata
Attester
Section
Observation Resource
Device Resource
Prescription Resource
Patient Resource

<feed>
  <entry>
    <Composition />
  </entry>
  <entry>
    <Observation />
  </entry>
  <entry>
    <Device />
  </entry>
  <entry>
    <Prescription />
  </entry>
  <entry>
    <Patient />
  </entry>
</feed>
Tag as “Document”

```xml
<feed xmlns="http://www.w3.org/2005/Atom">
  <title>This is my first Bundle</title>
  <id>urn:uuid:9f395ee0-19d2-4760-baf6-097fda52d914</id>
  <updated>2014-03-10T12:42:08.6834841Z</updated>
  <category term="http://hl7.org/fhir/tag/document"
             scheme="http://hl7.org/fhir/tag" />
  <entry/>
</feed>
```

This Bundle is a Document
You can “drop” your document on
- http://server.org/fhir/Mailbox

No storage or disassembly is implied, your just posting a document in its entirety.

Servers can implement any specific functionality as required between trading partners when receiving such a document.
Communicating documents

- You can store your document using
- Storage, NO disassembly is implied, document (and signature) stays intact
- Search is supported (you search on it’s Message header – Composition)
V3 and FHIR

Hospital System A

Note: Documents are compositions

- No update semantics
- Context?
- Wholeness?

CDA to FHIR Document bridge

FHIR Document processor

FHIR Repository

V3 CDA Documents

FHIR Documents

© 2014 HL7 © International. Licensed under Creative Commons. HL7 & Health Level Seven are registered trademarks of Health Level Seven International. Reg. U.S. TM Office.
Messages – are bundles

Message Resource
- source
- destination
- event

Observation Resource

Device Resource

Patient Resource

<feed>
  <entry>
    <MessageHeader />
  </entry>
  <entry>
    <Observation />
  </entry>
  <entry>
    <Patient />
  </entry>
  <entry>
    <Device />
  </entry>
</feed>
Tag as “Message”

```xml
<feed xmlns="http://www.w3.org/2005/Atom">
  <title>This is my first Bundle</title>
  <id>urn:uuid:9f395ee0-19d2-4760-baf6-097fda52d914</id>
  <updated>2014-03-10T12:42:08.6834841Z</updated>
  <entry></entry>
</feed>
```

This Bundle is a Message
Sending messages

- Again, REST not necessary, but...
- There is an explicit REST endpoint:
  - http://server.org/fhir/Mailbox
- No storage implied. Might be a router, converted to v2, etc. etc.
- The server can process them based on the event code and return the response as another message (again a bundle).
V2 and FHIR

Note: Messages are events, REST exposes a “repository” Model of data
The Binary Endpoint

http://server.org/fhir/Binary/

- Accepts any kind of content
- Stores the content as is, along with the content type provided by the HTTP headers.
- Acts just like the normal Resource endpoints (but there is no search)
Useful for Attachments

Media (Resource)

- type: code 1..1 <<MediaType>>
- subtype: CodeableConcept 0..1 <<MediaSubtype>>
- identifier: Identifier 0..*
- dateTime: dateTime 0..1
- subject: Resource(Patient | Practitioner | Group | Device | Specimen) 0..1
- requester: Resource(Practitioner) 0..1
- operator: Resource(Practitioner) 0..1
- view: CodeableConcept 0..1 <<MediaView>>
- deviceName: string 0..1
- height: integer 0..1
- width: integer 0..1
- frames: integer 0..1
- length: integer 0..1
- content: Attachment 1..1

/Binary/23344
INSIDE THE FHIR DISTRIBUTION
2.0 Implementation

Exchange Frameworks
Define how Resources are exchanged.

- RESTful API (HTTP)
- Search / Query
- Documents
- Messaging
- Services (SOA)

Support
Implementation Support.

- Downloads - Schemas, Code, Tools
- Managing Resource Identity
- Push vs Pull
- Support Links

Using FHIR
Making use of FHIR.

- Common Use Cases
- Profiles Defined as part of FHIR
- Security & Security Labels
- Integrated Examples

2.0.1 Implementers Safety Check List

FHIR is as simple to implement as we know how to make it. However, due to the nature of healthcare, and healthcare processes, and cultural concerns, there are a number of features in FHIR that implementers are obliged to consider in order to implement safe systems.

This section is a check list to help implementers be sure that they’ve considered all the parts of FHIR that impact on their system design with regard to safety.
Browsing the site

- REST API
- Data Types
- XML & JSON
- Codes / Terminologies
- Resource List
- Stack Overflow
- Public Test servers
The FHIR distribution

- Under Implementation-Support-Downloads:
  - The XSD schema’s / schematrons
  - The Java / C# / Delphi zips (model, serializers, parsers, etc., both code & binary)
  - All xml + json examples
  - Full spec for offline reading (always have your FHIR with you)

Note! .NET implementation has moved to GitHub, distribution by NuGet (.NET) and Maven (Java)
In the FHIR SVN

- All you need to build FHIR (/build)
- All presentations (/presentations)
- Source of the publication process (/build/tools, we use Eclipse + Java 1.6)
- Archived older versions of FHIR (/archive)

- You can download only the /build
- Then run publish.bat & wait
The FHIR SVN

- The “build” SVN tree, the “full” SVN tree
  - [http://gforge.hl7.org/svn/fhir](http://gforge.hl7.org/svn/fhir)
  - User ‘anonymous’, blank password

- Note: you **have to run the publisher** to be able to build the C# and Delphi source. Without that…they won’t compile!
## “Source” of FHIR

### Table: FHIR Resources

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
<td>Card.</td>
<td>Type</td>
<td>Short Name</td>
</tr>
<tr>
<td>Patient</td>
<td></td>
<td>Resource</td>
<td>Information about a person or animal receiving health care services</td>
</tr>
<tr>
<td>Patient.link</td>
<td>0..*</td>
<td>Resource(Patient)</td>
<td>Other patients linked to this resource</td>
</tr>
<tr>
<td>Patient.active</td>
<td>0..1</td>
<td>boolean</td>
<td>Whether this patient record is in active use</td>
</tr>
<tr>
<td>Patient.identifier</td>
<td>0..*</td>
<td>Identifier</td>
<td>An identifier for the person as this patient</td>
</tr>
<tr>
<td>Patient.details</td>
<td>0..1</td>
<td>Demographics</td>
<td>Patient demographics</td>
</tr>
<tr>
<td>Patient.contact</td>
<td>0..*</td>
<td>=Contact</td>
<td>A contact party (e.g. guardian, partner, friend) for the patient</td>
</tr>
<tr>
<td>Patient.contact.relationship</td>
<td>0..*</td>
<td>CodeableConcept</td>
<td>The kind of relationship</td>
</tr>
</tbody>
</table>

---

Straight from the HL7 SVN “code” repository at gforge.hl7.org

© 2014 HL7 ® International. Licensed under Creative Commons. HL7 & Health Level Seven are registered trademarks of Health Level Seven International. Reg. U.S. TM Office.
Generator writers!

- There’s a file called eCoreDefinitions.xml that the C# generator runs of. It has all details from the definitions.

- There are Profiles for each resource, basically describing the “unconstrained” resources.
Short introduction to

PROFILES AND VALIDATION
The need for Profiles

- Many different contexts in healthcare, but a single set of Resources
- Need to be able to describe restrictions based on use and context
- Allow for these usage statements to:
  - Authored in a structured manner
  - Published in a repository
  - Used as the basis for validation, code, report and UI generation.
Constraining cardinality

Limit cardinality to 1..2 (e.g. to at maximum your organizations’ identifier + the national one)

Limit names to just 1 (instead of 0..*)

Forbid any telecom elements

Note: something that’s mandatory in the core definition cannot be made optional in a profile
Limit value domains

```
deceased[x] : boolean | dateTime 0..1
address : Address 0..*
maritalStatus : CodeableConcept 0..1 <<MaritalStatus>>
multipleBirth[x] : boolean | integer 0..1
photo : Attachment 0..*
communication : CodeableConcept 0..* <<Language>>
provider : Resource(Organization) 0..1
link : Resource(Patient) 0..1
active : boolean 0..1
```

- If deceased is given, it must be a dateTime, not a boolean
- Use our national codes for MaritalStatus
- Use another profiled Resource
- Fix value: Only allow “active” Patients

© 2014 HL7 ® International. Licensed under Creative Commons. HL7 & Health Level Seven are registered trademarks of Health Level Seven International. Reg. U.S. TM Office.
Tagging a Resource

Patient

- MRN 22234
- “Ewout Kramer”
- 30-11-1972
- Amsterdam

http://hl7.org/fhir/tag/security
“I’m a VIP - My information cannot yet be disclosed”

http://hl7.org/fhir/tag
“This is TEST data! Don’t use!”

http://hl7.org/fhir/tag/profile
“I’m a Patient as defined in the Norwegian Profile – see http://hl7.no/Profiles/patient-no”
2.1.12 validate 🌍

The validate interaction checks whether the attached content would be acceptable as an update to an existing resource. The interaction is performed by an HTTP POST command as shown:

```
POST [base]/[type]/_validate{/[id]}
```
Validation

- When receiving an XML resource
  - 1. Validate using schema
  - 2. Run schematrons

- When receiving JSON
  - 1. Parse the JSON
  - 2. Serialize to XML
  - 3. Validate using schema
  - 4. Run schematrons

- There’s a validation pack for Java
(Distributed) validation

App’s server

Profile X
Profile Y

Store & Validate

Country validation server

Profile X

Validate Y

Download & Validate

Profile Y

Profile Y

Profile Y
Operation Outcome

When something goes wrong….return the OperationOutcome Resource!

```
<OperationOutcome>
  <text>
    <status value="additional"/>
    <div xmlns="http://www.w3.org/1999/xhtml">
      <p>W is not a recognized code for Gender.</p>
    </div>
  </text>
  <issue>
    <severity value="error"/>
    <type>
      <system value="http://test.org/issueCodeSystem"/>
      <code value="V15"/>
      <display value="InvalidCode"/>
    </type>
    <location value="/Person[1]/gender[1]"/>
  </issue>
</OperationOutcome>
```
FHIR in practice
BUILDING A FHIR SERVER
Some possible uses
Repository model

Vendor Neutral Repository
Overview of a server

HTTP / REST interface

Encoding/decoding, param validation, syntax validation

Implement service operations as described in spec

Fhir Service

Indexer / Search

Storage

© 2014 HL7 ® International. Licensed under Creative Commons. HL7 & Health Level Seven are registered trademarks of Health Level Seven International. Reg. U.S. TM Office.
From wire to store

REST interface

<table>
<thead>
<tr>
<th>JSON/XML</th>
<th>JSON/XML</th>
<th>JSON/XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>FHIR Parser</td>
<td>FHIR Parser</td>
<td>FHIR Parser</td>
</tr>
<tr>
<td>POCO/POJO</td>
<td>POCO/POJO</td>
<td>POCO/POJO</td>
</tr>
<tr>
<td>O-R Map</td>
<td>Serialize</td>
<td>Serialize</td>
</tr>
<tr>
<td>DBMS</td>
<td>NoSql (Xml/Json)</td>
<td>DBMS</td>
</tr>
</tbody>
</table>

Storage

© 2014 HL7 ® International. Licensed under Creative Commons. HL7 & Health Level Seven are registered trademarks of Health Level Seven International. Reg. U.S. TM Office.
Briefest intro to JSON

- JSON ("JavaScript Object Notation"): it’s JavaScript, not markup!

```javascript
{
    "Country": {
        "name": "the Netherlands",
        "population": 16696000,
        "popDensity": 447.9
    }
}
```

- `var report = eval("{{...}}")`;
  `alert(report.LabReport.status);`

- Very easy parsing for JavaScript clients. But actually, use `JSON.parse()` instead ;-)
## Xml and JSON are different

<table>
<thead>
<tr>
<th>Xml</th>
<th>JSON</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;XXX xmlns=&quot;urn:foo&quot;&gt;</code></td>
<td><code>{ “B”: { “a” : “c” }, </code></td>
</tr>
<tr>
<td><code>   &lt;B a=&quot;c&quot; /&gt;</code></td>
<td><code>  “C”: [ “One”, “Two” ],</code></td>
</tr>
<tr>
<td><code>   &lt;C&gt;One&lt;/C&gt;</code></td>
<td>`   “D” : “One”,</td>
</tr>
<tr>
<td><code>   &lt;C&gt;Two&lt;/C&gt;</code></td>
<td>`   “div” : {</td>
</tr>
<tr>
<td><code>   &lt;D&gt;One&lt;/D&gt;</code></td>
<td>`     “text-before”:“Not ”,</td>
</tr>
<tr>
<td><code>   &lt;div&gt;Not &lt;b&gt;so&lt;/b&gt; easy&lt;/div&gt;</code></td>
<td>`     b:“so”,</td>
</tr>
<tr>
<td><code>&lt;/XXX&gt;</code></td>
<td>`     “tekst-after”:“easy”}</td>
</tr>
</tbody>
</table>

- How to retain namespaces?  
- How do you identify attributes?  
- Bridge difference in datatypes?

- Distinguish single elements from lists with one element?  
- Mixed content?
<DocumentReference xmlns="http://hl7.org/fhir">
  <masterIdentifier>
    <system value="urn:ietf:rfc:3986"/>
    <value value="urn:oid:1.3.6.1.4.1.21367:"
  </masterIdentifier>
  <created value="2005-12-24T09:35:00+11:00"/>
  <confidentiality>
    <coding>
      <system value="http://ihe.net/xds/co:"
      <code value="1.3.6.1.4.1.21367.2006.">
      <display value="Clinical-Staff"/>
      </coding>
    </confidentiality>
</DocumentReference>
Xml and Json in FHIR

```json
{
    "resourceType" : "DocumentReference",
    "masterIdentifier" : {
        "system" : "urn:ietf:rfc:3986",
        "value" : "urn:oid:1.3.6.1.4.1.21367.2005.3.7"
    },
    "created" : "2005-12-24T09:35:00+11:00",
    "confidentiality" : [
        {
            "coding" : [
                {
                    "system" : "http://ihe.net/xds/connectathor",
                    "code" : "1.3.6.1.4.1.21367.2006.7.101",
                    "display" : "Clinical-Staff"
                }
            ],
            "display" : "Clinical-Staff"
        }
    ]
}
```
Handling both

- You need “meta” knowledge of the definition to distinguish lists / attributes
- The Java/C# API’s can easily interconvert
- Digital Signatures (in json?) are a problem when converting → store the original
- JAXB / XmlSerializer / DataContract would need extensive customization / additional @annotation / [attributes]. Contact us!
A document-oriented store can store Resources as a “whole” document.

E.g. MongoDB stores documents in JSON:

```javascript
post = {
  author: "mike",
  text: "my blog post...",
  tags: ["mongodb", "intro"]
};

db.posts.save(post);

db.posts.find({ author: "mike" });
db.posts.find().sort({date: -1}).limit(10);
```
private void insertUpdatedRecord(string collectionName, ResourceRecord oldVersion,)
{
    var coll = getDbCollectionForResource(collectionName);

    // TODO: Note that this update assumes that we insert a new record with a new version in a multi-user environment is not guaranteed to be the latest version number.
    coll.Insert(newVersion);

    // TODO: Mark old record as superceded, this really needs a transaction ;-) the
    coll.Update(Query.EQ(ID_LABEL, oldVersion.RecordId),
    Update.Set(STATE_LABEL, RecordState.Superceded));
}

var resourceIdQuery = Query.EQ("_id", resource.CollectionName);
var newId = coll.FindAndModify(resourceIdQuery, null, Update.Inc("next",1), true, true);

return newId.ModifiedDocument["next"].AsInt32;
Batch: needs transactions

Transaction

The transaction interaction submits a set of resources to be updated, created or deleted on the server. This interaction allows multiple resources to be updated/created in a single transaction. The interaction is performed by an HTTP POST command as shown:

```
POST [service-url] (?_format=mimeType)
```

The content of the post submission is a resource bundle.
# RDBMS: BLOB + Index

<table>
<thead>
<tr>
<th>Id</th>
<th>ResourceId</th>
<th>VersionId</th>
<th>LastUpdate</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2012-12-19</td>
<td><code>&lt;Patient ...&gt;</code></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2012-12-20</td>
<td><code>&lt;Patient ...&gt;</code></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2012-12-20</td>
<td><code>&lt;Observation ...&gt;</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Id</th>
<th>Param</th>
<th>Value</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Patient.Name</td>
<td>“Ewout”</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Patient.DoB</td>
<td>“1972-11-30”</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Obs.Code</td>
<td>“234332”</td>
<td>SNOMED</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>VersionIdHigh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient</td>
<td>2</td>
</tr>
<tr>
<td>Lab</td>
<td>3</td>
</tr>
</tbody>
</table>

Separate tables for current and history?

Need to index only latest version!
Both implementations use a separately maintained index for search?

Yes, here’s why…
Predictable search

Each resource has a fixed set of search parameters:

<table>
<thead>
<tr>
<th>Name / Type</th>
<th>Description</th>
<th>Paths</th>
</tr>
</thead>
<tbody>
<tr>
<td>_id : token</td>
<td>The logical resource id associated with the resource (must be supported by all servers)</td>
<td></td>
</tr>
<tr>
<td>date : date</td>
<td>obtained date/time. If the obtained element is a period, a date that falls in the period</td>
<td>Observation.applies[x]</td>
</tr>
<tr>
<td>name : token</td>
<td>The name of the observation type or component</td>
<td>Observation.name type</td>
</tr>
<tr>
<td>name-value :</td>
<td>Both name and value</td>
<td></td>
</tr>
<tr>
<td>composite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>performer :</td>
<td>who/what performed the observation</td>
<td>Observation.performer</td>
</tr>
<tr>
<td>reference</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reliability :</td>
<td>The reliability of the observation</td>
<td>Observation.reliability</td>
</tr>
<tr>
<td>token</td>
<td></td>
<td></td>
</tr>
<tr>
<td>status : token</td>
<td></td>
<td>Observation.status</td>
</tr>
</tbody>
</table>
## Types of parameters

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>integer</td>
<td>Search parameter must be a simple whole number.</td>
</tr>
<tr>
<td>string</td>
<td>Search parameter is a simple string, like a name part. <strong>Search is case-insensitive and accent-insensitive.</strong> May match just the start of a string.</td>
</tr>
<tr>
<td>date</td>
<td>Search for an exact match on a date (parameters look like 1956-05-27T12:34:12+04:00 or shorter)</td>
</tr>
<tr>
<td>token</td>
<td>Search parameter on a coded element or identifier. May be used to search <strong>through the text, displayname, code and code/codesystem</strong> (for codes) and label, system and key (for identifier).</td>
</tr>
<tr>
<td>reference</td>
<td>A pair of resource type and resource id, separated by &quot;/&quot;. Matches when the resource reference resolves to a resource of the given type and id.</td>
</tr>
</tbody>
</table>

*Note: you need to escape the query-string!!*
Partial/combined match

The search parameter description shows that the match certainly isn’t always exact, and can even combine the contents of several elements:

<table>
<thead>
<tr>
<th>Name / Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>address : string</td>
<td>an address in any kind of address/part of the patient</td>
</tr>
<tr>
<td>family : string</td>
<td>a portion of the family name of the patient</td>
</tr>
<tr>
<td>given : string</td>
<td>a portion of the given name of the patient</td>
</tr>
<tr>
<td>name : string</td>
<td>a portion of either family or given name of the patient</td>
</tr>
<tr>
<td>phonetic : string</td>
<td>a portion of either family or given name using some kind of phonetic matching algorithm</td>
</tr>
<tr>
<td>telecom : string</td>
<td>the value in any kind of telecom details of the patient</td>
</tr>
</tbody>
</table>
Prepare your data!

```
<details>
  <name>
    <use value="official"/>
    <family value="von"/>
    <family value="Hochheim-Weilenfels"/>
    <given value="Regina"/>
    <given value="Johanna"/>
    <given value="Maria"/>
    <prefix value="Gräfin"/>
  </name>
  <name>
    <use value="maiden"/>
    <family value="Hochheim"/>
  </name>
  <telecom>
    <system value="phone"/>
    <value value="555-555-2004"/>
    <use value="work"/>
  </telecom>
</details>
```

patient/search?
name=johan&
name=grafin&
telecom=5552004

“Groom” – prepare for searching
THE END IS NEAR...
Balloting plans

- First Draft Standard for Trial Use ballot (DSTU) complete
  - FHIR is now published as a DSTU
  - Will provide a semi-stable platform for implementers while still allowing non-backward-compatible change for Normative version if implementation experience dictates
  - Additional DSTU versions roughly annually to make fixes, introduce new resources

- Normative is around 3 years out
  - We want *lots* of implementation experience before committing to backward compatibility
Next Steps for you

- Read the spec: http://hl7.org/fhir
- Try implementing it
- Come to a (European?) Connectathon!

- fhir@lists.hl7.org
- #FHIR
- Implementor’s Skype Channel
- FHIR Developer Days (November 24 – 26), Amsterdam
- StackOverflow: hl7 fhir tag
International HL7 FHIR Developer Days
November 24-26, 2014 in Amsterdam

- Education
  - 14 tutorials
    - pick & choose

- Connectathon
  - Meet fellow developers
  - Put FHIR to the test

- Networking
  - FHIR experts and authors on hand

http://fhir.furore.com/devdays
QUESTIONS?