Introduction to the HL7® Fast Healthcare Interoperability Resources (FHIR®) standard
Introduction to the HL7® FHIR® standard

➢ This is the first of three webinars
  • An overview of FHIR and beginners guide to getting started using FHIR

➢ Two subsequent webinars will look in more depth at
  • The services provided by the National Clinical Terminology Service (NCTS)
  • FHIR terminology services
In this webinar

1. What is FHIR?
   - Basic concepts

2. Playing with FHIR
   - FHIR specification introduction
   - Getting started
   - Useful tools & resources

3. FHIR and Clinical Terminology
   - Terminology Resources

4. FHIR Terminology Operations

HL7, CDA, and FHIR are the registered trademarks of Health Level Seven International and the use does not constitute endorsement by HL7.
What is FHIR®?
What is FHIR®?

➢ Grahame Grieve wrote the initial strawman version of FHIR between May and August 2011, it was called ‘Resources for Health (RFH)’

➢ FHIR today is a standard with a mix of normative and non-normative content and currently at Release 4.0.1
  • The normative content is primarily within the infrastructural resource of FHIR; Patient and Observation are the only clinical resource currently normative
  • FHIR Release 5 is in preview and continues to be developed by the community – new resource, more normative resource, implementation guides, cross version support are some of the focuses for release 5

➢ FHIR is more than just a transport format – unlike its HL7 predecessors, it defines how to send and receive content as well as the format

➢ FHIR is a RESTful API – many of today’s most successful web companies are built around a RESTful architecture such as Facebook, LinkedIn, Twitter and thousands of others

➢ Extensibility was built into FHIR from day one, it focuses on the 80% and allows extensibility to cover the rest

➢ FHIR has a strong focus on implementation
  • Intended to be fast and easy to implement by developers
  • Foundation in Web standards: HTTP, JSON, XML, OAUTH
Messaging, Document and Resource paradigm – Health Level Seven® (HL7) standards

➢ HL7 V2 Messaging
  • An event happened and here is the data about that event

➢ CDA Document
  • A document that is published

➢ FHIR Resources
  • FHIR has a Messaging & Document paradigm but is primarily a Resource oriented paradigm
  • The resources represent granular clinical concepts, linked together to form a web of information and higher level concepts (such as a Pathology Report or Discharge Summary)
Playing with FHIR®
Where to start reading?

The FHIR specification: [http://hl7.org/fhir](http://hl7.org/fhir)

Start with the Documentation tab and RESTful API: [http://hl7.org/fhir/documentation.html](http://hl7.org/fhir/documentation.html)
Resource definitions

All FHIR resources are listed on the Resources tab: [http://hl7.org/fhir/resourcelist.html](http://hl7.org/fhir/resourcelist.html)

- Familiarise yourself with the layout of a FHIR Resource page
- The Patient resource is a friendly place to start
- All resource pages are structured the same way
- Concentrate on the structure diagram for the resource and the search parameters at the bottom of the page
Patient resource – structure diagram

Describes the structure of the resource content: [http://hl7.org/fhir/patient.html](http://hl7.org/fhir/patient.html)
### Patient resource – examples

View examples for each resource type, in various formats: [http://hl7.org/fhir/patient-examples.html](http://hl7.org/fhir/patient-examples.html)

<table>
<thead>
<tr>
<th>Example Name</th>
<th>id</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Person Example</td>
<td>example</td>
<td>XML JSON Turtle</td>
</tr>
<tr>
<td>Patient 1 for linking</td>
<td>pat1</td>
<td>XML JSON Turtle</td>
</tr>
<tr>
<td>Patient 2 for linking</td>
<td>pat2</td>
<td>XML JSON Turtle</td>
</tr>
<tr>
<td>Deceased patient (using time)</td>
<td>pat3</td>
<td>XML JSON Turtle</td>
</tr>
<tr>
<td>Deceased patient (using boolean)</td>
<td>pat4</td>
<td>XML JSON Turtle</td>
</tr>
<tr>
<td>Stock people (defined by HL7 publishing)</td>
<td>b248b1b2-1685-4b94-9938-37d7a5f94b51</td>
<td>XML JSON Turtle</td>
</tr>
</tbody>
</table>
FHIR® requests

The FHIR API follows the RESTful paradigm of managing state through SCRUD interactions.

Each action is defined by the HTTP method used.

<table>
<thead>
<tr>
<th>Action</th>
<th>HTTP Request Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>POST</td>
</tr>
<tr>
<td>Search &amp; Read</td>
<td>GET</td>
</tr>
<tr>
<td>Update</td>
<td>PUT</td>
</tr>
<tr>
<td>Delete</td>
<td>DELETE</td>
</tr>
</tbody>
</table>
# FHIR® requests – HTTP components

Each FHIR request is comprised of the following HTTP components:

| **URL** | Defines the destination FHIR server and target FHIR resource(s)  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Defines request parameters (URL encoded)</td>
</tr>
</tbody>
</table>
|         | http://test.fhir.org/r4/Patient?family=Chalmers&given=Peter  
|         | http://test.fhir.org/r4/Patient/0064869c-76e2-4413-ac64-d016a5f837 |
| **Method** | GET, POST, PUT, etc. (SCRUD)                                     |
| **Headers** | Additional instructions, e.g. content negotiation and authorisation |
| **Accept** | application/fhir+json                                           |
| **Content-Type** | application/xml                                                |
| **Authorisation** | Bearer {token}                                                  |
| **Body** | A FHIR document being sent to the server (if appropriate)  
|         | in a specific format (e.g. XML or JSON)                         |
FHIR® requests – HTTP components

- **URL**: https://stu3.test.pyrohealth.net/fhir/Patient?family=Chalmers&given=Peter
- **Method**: GET
- **Headers**:
  - Content-Type: application/fhir+json
  - Accept: application/fhir+json
- **Body**:
  ```json
  "resourceType": "Bundle",
  "type": "searchset",
  "total": 17,
  "link": [
    {
      "relation": "first",
      "url": "https://stu3.test.pyrohealth.net/fhir/Patient?family=Chalmers&given=Peter&page=1"
    }
  ]
  ```
FHIR® requests – content negotiation

Content negotiation dictates the format exchanged in requests and responses:

<table>
<thead>
<tr>
<th>HTTP Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Type</td>
<td>The format to be SENT to the FHIR server. This is the format of the body of the REQUEST.</td>
</tr>
<tr>
<td>Accept</td>
<td>The format to be RECEIVED from the FHIR server. This is the requested format for the body of the RESPONSE.</td>
</tr>
</tbody>
</table>

Both need to be set by you in your HTTP Post client to interact with a FHIR server.
FHIR® requests – content negotiation

HTTP Request

URL: GET https://test.fhir.org/fhir

Header
Accept: application/fhir+json
Content-Type: application/fhir+xml

Body

The content type I am giving the FHIR server is in XML format.
I accept content returned to me from the FHIR server in JSON format.

HTTP Response

Header
Content-Type: application/fhir+json

Body

The content type I am giving the FHIR server is in XML format.
I accept content returned to me from the FHIR server in JSON format.
# FHIR® requests – content negotiation

![Screenshot of FHIR requests content negotiation](image)

## URL

```
https://stu3.test.pyrohealth.net/fhir/Patient?family=Chalmers&given=Peter
```

## Method

- GET

## Headers

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Type</td>
<td>application/fhir+xml</td>
</tr>
<tr>
<td>Accept</td>
<td>application/fhir+json</td>
</tr>
</tbody>
</table>

## Body

```
{
  "resourceType": "Bundle",
  "type": "searchset",
  "total": 17,
  "link": [
    {
      "relation": "first",
      "url": "https://stu3.test.pyrohealth.net/fhir/Patient?family=Chalmers&given=Peter&page=1"
    }
  ]
}
```
## FHIR® response

The outcome of the request is defined in:

<table>
<thead>
<tr>
<th>HTTP Response Code</th>
<th>Eg. For an UPDATE request</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 “OK”</td>
<td>Existing resource UPDATED</td>
</tr>
<tr>
<td>201 “Created”</td>
<td>New resource CREATED</td>
</tr>
<tr>
<td>400 “Bad Request”</td>
<td>Resource could not be parsed or failed basic FHIR validation rules</td>
</tr>
</tbody>
</table>

http://hl7.org/fhir/http.html#update

**Body**

Either:

1. A version of the FHIR resource  
   or
2. An OperationOutcome FHIR resource (eg, error conditions)
## Getting started

### Performing a FHIR request – tools:

<table>
<thead>
<tr>
<th>Client</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Basic HTTP client</td>
<td>1. Existing test server or reference implementation</td>
</tr>
<tr>
<td>2. API development tools</td>
<td>2. Existing FHIR endpoints</td>
</tr>
<tr>
<td>3. HTTP client libraries</td>
<td>3. Create own server instance</td>
</tr>
<tr>
<td>4. FHIR client library</td>
<td></td>
</tr>
</tbody>
</table>


Non-developers would need some familiarity with either XML or JSON, and the basic use of a HTTP Post client.
**FHIR® server capabilities**

A compliant FHIR server will produce a CapabilityStatement resource defining the functionality actually supported by the server.

- HTTP GET: `{baseUrl}/metadata`
- See [http://hl7.org/fhir/http.html#capabilities](http://hl7.org/fhir/http.html#capabilities)

<table>
<thead>
<tr>
<th>Instance Level Interactions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>read</td>
<td>Read the current state of the resource</td>
</tr>
<tr>
<td>vread</td>
<td>Read the state of a specific version of the resource</td>
</tr>
<tr>
<td>update</td>
<td>Update an existing resource by its id (or create it if it is new)</td>
</tr>
<tr>
<td>patch</td>
<td>Update an existing resource by posting a set of changes to it</td>
</tr>
<tr>
<td>delete</td>
<td>Delete a resource</td>
</tr>
<tr>
<td>history</td>
<td>Retrieve the change history for a particular resource</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type Level Interactions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>create</td>
<td>Create a new resource with a server assigned id</td>
</tr>
<tr>
<td>search</td>
<td>Search the resource type based on some filter criteria</td>
</tr>
<tr>
<td>history</td>
<td>Retrieve the change history for a particular resource type</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Whole System Interactions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>capabilities</td>
<td>Get a capability statement for the system</td>
</tr>
<tr>
<td>batch/transaction</td>
<td>Update, create or delete a set of resources in a single interaction</td>
</tr>
<tr>
<td>history</td>
<td>Retrieve the change history for all resources</td>
</tr>
<tr>
<td>search</td>
<td>Search across all resource types based on some filter criteria</td>
</tr>
</tbody>
</table>
Performing a SEARCH request

Example (1): Give me all Patient resources with a specified name

➢ HTTP GET:

• http://test.fhir.org/r4/Patient?family=Chalmers = Give me all Patient resources where the family name contains “chalmers”

• http://test.fhir.org/r4/Patient?family=Chalmers&given=Peter,James = Give me all Patient resources where the family name contains “chalmers” AND the given name contains “Peter” OR “James”

8.1.11 Search Parameters

Search parameters for this resource. The common parameters also apply. See Searching for more information about searching in REST, messaging, and services.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Expression</th>
<th>In Common</th>
</tr>
</thead>
<tbody>
<tr>
<td>family</td>
<td>string</td>
<td>A portion of the family name of the patient</td>
<td>Patient.name.family</td>
<td>1 Resources</td>
</tr>
<tr>
<td>gender</td>
<td>token</td>
<td>Gender of the patient</td>
<td>Patient.gender</td>
<td>3 Resources</td>
</tr>
<tr>
<td>general-practitioner</td>
<td>reference</td>
<td>Patient’s nominated general practitioner, not the organization that manages the record</td>
<td>Patient.generalPractitioner</td>
<td></td>
</tr>
<tr>
<td>(Practitioner, Organization)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>given</td>
<td>string</td>
<td>A portion of the given name of the patient</td>
<td>Patient.name.given</td>
<td>1 Resources</td>
</tr>
<tr>
<td>Identifier</td>
<td>token</td>
<td>A patient identifier</td>
<td>Patient.identifier</td>
<td></td>
</tr>
<tr>
<td>language</td>
<td>token</td>
<td>Language code (irrespective of use value)</td>
<td>Patient.communication.language</td>
<td></td>
</tr>
<tr>
<td>link</td>
<td>reference</td>
<td>All patients linked to the given patient</td>
<td>Patient.link.other (Patient, RelatedPerson)</td>
<td></td>
</tr>
</tbody>
</table>

At the bottom of each Resource page in the FHIR specification is the search parameters for that resource.

Some servers may not support all search parameters. Refer to the servers provided capability statement.
Performing a SEARCH request

Example (2): Give me all Patient resources where a Medicare number identifier is found with the value 2950156481

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Parameter Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
<td>Token</td>
</tr>
</tbody>
</table>

Token Format

\[[parameter]=[system]|[code]\]

- HTTP GET:

- There are 8 search parameter types: http://hl7.org/fhir/search.html
Further training

**FHIR Drills**


This interactive web site will walk you through the basics of how to interact with a FHIR server:

- Create a test patient resource in a FHIR server,
- Read that resource from the FHIR server
- Update that resource in the FHIR server
- Delete that resource from the FHIR server

Note: this site still used FHIR STU3, however the knowledge you learn is still applicable to FHIR R4.
FHIR® and Clinical Terminology
The FHIR specification focuses on improving the delivery and maintainability of terminology.

Some FHIR servers, like CSIRO’s Ontoserver™, are dedicated to terminology functions.

There are 3 core FHIR resources used to manage terminology:

<table>
<thead>
<tr>
<th>Code System</th>
<th>A catalogue of codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value Set</td>
<td>A collection of codes for a specific use</td>
</tr>
<tr>
<td></td>
<td>Constructed from one or more Code Systems</td>
</tr>
<tr>
<td>Concept Map</td>
<td>Maps codes from one Value Set to another</td>
</tr>
</tbody>
</table>
CodeSystem & ValueSet examples

I may define a CodeSystem which provides an ACME identifier for all ACME hospitals around the world.

From this CodeSystem I could select all the codes for the Australian hospitals that I want to be displayed in my Patient Administration Systems (PAS) and create a ValueSet for that purpose.

And in Western Australia, I want only W.A. hospitals to be selectable for pathology episodes in my Laboratory Information System so this could be another ValueSet.
ConceptMap

Translates codes from one ValueSet to another, usually in **one** direction only.

Example: Translate the ACME hospital codes to Healthcare Provider Identifier – Organisation (HPI-O) numbers.
FHIRE® Terminology Operations
FHIR® RESTful API operations

FHIR also acts as an operations framework.

➢ A FHIR server may support specific functions (or actions) that can be performed.
➢ Read the specification for more information: [http://hl7.org/fhir/operationslist.html](http://hl7.org/fhir/operationslist.html)

➢ In the URL, an operation is signified by a dollar sign ($)

➢ Operations may be performed at different levels:

<table>
<thead>
<tr>
<th>Base Operation</th>
<th>{baseUrl}$/MyOperation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Operation</td>
<td>{baseUrl}/ConceptMap/$MyOperation</td>
</tr>
<tr>
<td>Resource Instance Operation</td>
<td>{baseUrl}/ConceptMap/12345/$MyOperation</td>
</tr>
</tbody>
</table>
FHIR® terminology operations

Here are 3 key FHIR operations used in terminology

**Concept Translation**
Translate a code from one value set to another, based on existing value set and concept map resources. The operation returns a set of parameters including a 'result' for whether there is an acceptable match, and a list of possible matches.

```
```

**ValueSet Expansion**
Get a list of codes from a value set matching a specific criteria.

```
GET [base]/ValueSet/23/$expand?filter=blah
```

**ValueSet Validation**
Confirm that a code exists in a Value Set.

```
```
Questions?
Contact us

Help Centre  1300 901 001

Email  help@digitalhealth.gov.au

Website  healthterminologies.gov.au
developer.digitalhealth.gov.au
digitalhealth.gov.au

twitter.com/AuDigitalHealth

OFFICIAL