Java and .Net FHIR client

James Agnew and Mirjam Baltus

Redmond, June 10 – 12 | @HL7 @FirelyTeam | #fhirdevdays | www.devdays.com/us

Who are we?

- **Name:** James Agnew
- **Background:**
  - Software Developer by day
  - Project lead for HAPI for 14 years
  - Lead Architect: Centre for Global eHealth Innovation
  - CTO: Smile CDR

- **Name:** Mirjam Baltus
- **Background:**
  - Firely team
  - FHIR trainer & Support

Prerequisites

- This tutorial assumes you are at least comfortable with the following:
  - FHIR
  - Java or C#
  - An IDE (Eclipse or IntelliJ IDEA recommended for Java, Visual Studio for .Net)
  - Apache Maven (build system) or NuGet Package manager

FHIR: A Quick Recap (1)

- FHIR offers a model for Healthcare concepts

- For example, Patient:

Structure

Name	Flags	Card.	Type	Description & Constraints
Patient			DomainResource	Information about an individual or animal receiving health care services Elements defined in Ancestors: id, meta, implicitRules, language, text, c
identifier	Σ	0..*	Identifier	An identifier for this patient
active	?! Σ	0..1	boolean	Whether this patient's record is in active use
name	Σ	0..*	HumanName	A name associated with the patient
telecom	Σ	0..*	ContactPoint	A contact detail for the individual
gender	Σ	0..1	code	male | female | other | unknown AdministrativeGender (Required)
birthDate	Σ	0..1	date	The date of birth for the individual
deceased[x]	?! Σ	0..1		Indicates if the individual is deceased or not
deceasedBoolean			boolean	
deceasedDateTime			dateTime	
address	Σ	0..*	Address	Addresses for the individual
maritalStatus		0..1	CodeableConcept	Marital (civil) status of a patient Marital Status Codes (Extensible)
multipleBirth[x]		0..1		Whether patient is part of a multiple birth
multipleBirthBoolean			boolean	
multipleBirthInteger			integer	
photo		0..*	Attachment	Image of the patient

FHIR: A Quick Recap (2)

- The standard specifies a RESTful API for interacting with that model
- For example, a search:

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



```
GET http://hapi.fhir.org/R4/Patient?birthdate=lt2016-11-13&_pretty=true
```
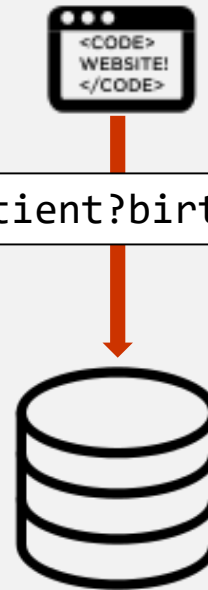
FHIR: A Quick Recap (3)

- FHIR has had several releases
  - FHIR DSTU1 (Released 2014)
  - FHIR DSTU2 (Released 2015)
  - FHIR R3 (Released 2017)
  - FHIR R4 (Released 2019)

- The concepts from this presentation apply to all versions
- For our purposes: DSTU / STU / R are interchangeable

API background

Java

- HAPI started in 2001 as an HL7 v2 Library
- Built to support a simple web portal, now used in applications around the world

```
HL7 v2 - http://hl7api.sourceforge.net
FHIR    - http://hapifhir.io
```

- HAPI is now the Java Reference Implementation

HAPI FHIR - Meet our new mascot!

.Net

- Has been developed with the FHIR standard from 2011
- Hl7.Fhir.<version>
- The official Reference Implementation for .Net

- Source: https://github.com/ewoutkramer/fhir-net-api
- Documentation: http://docs.simplifier.net/fhirnetapi/index.html

Code examples

- See the documentation of your API

- Take a look at:

    https://github.com/firelyteam/fhirstarters

The model

Structures JARs

- HAPI supports multiple versions of FHIR via different "structures JARs"

FHIR Version

HAPI Version

hapi-fhir-structures-dstu-2.5.jar
↳ ca.uhn.fhir.model.dstu.resources.Patient

Patient Resource Models

hapi-fhir-structures-dstu2-2.5.jar
↳ ca.uhn.fhir.model.dstu2.resources.Patient

hapi-fhir-structures-dstu3-2.5.jar
↳ org.hl7.fhir.dstu3.model.Patient

hapi-fhir-structures-r4-3.0.0.jar
↳ org.hl7.fhir.r4.model.Patient
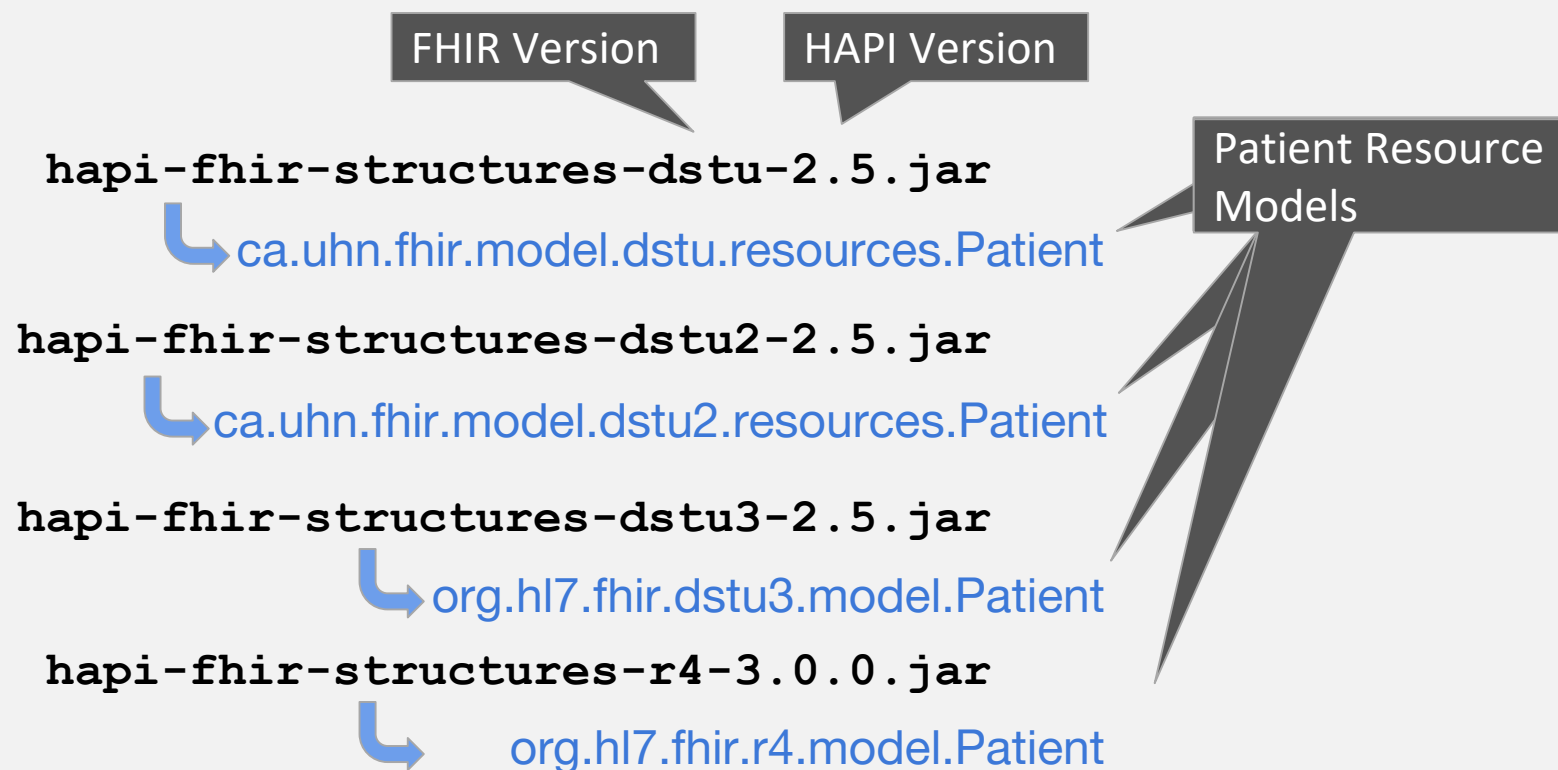
FHIR versions in .Net

- Different NuGet Packages
  - Hl7.Fhir.DSTU2
  - Hl7.Fhir.STU3
  - Hl7.Fhir.R4

- All have Hl7.Fhir.Model namespace

A FHIR resource

Name	Flags	Card.	Type	Description & Constraints
Observation	I		DomainResource	Measurements and simple assertions + If code is the same as a component code then the value element associated with the code SHALL NOT be present + dataAbsentReason SHALL only be present if Observation.value[x] is not present Elements defined in Ancestors: id, meta, implicitRules, language, text, contained, extension, modifierExtension
identifier	Σ	0..*	Identifier	Business Identifier for observation
basedOn	Σ	0..*	Reference(CarePlan | DeviceRequest | ImmunizationRecommendation | MedicationRequest | NutritionOrder | ProcedureRequest | ReferralRequest)	Fulfills plan, proposal or order
status	?! Σ	1..1	code	registered | preliminary | final | amended + ObservationStatus (Required)
category		0..*	CodeableConcept	Classification of type of observation Observation Category Codes (Preferred)
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
context		0..1	Reference(Encounter | EpisodeOfCare)	Healthcare event during which this observation is made
effective[x]	Σ	0..1		Clinically relevant time/time-period for observation
effectiveDateTime			dateTime	
effectivePeriod			Period	

A FHIR Resource class in C#

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

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

```csharp
    /// <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 = ObservationStatus.Preliminary;`

A FHIR Resource class in HAPI

- Resource definition classes implement IBaseResource

- Enumerations are available for required value sets

```
// Create a resource instance
Observation obs = new Observation();
obs.setStatus(ObservationStatus.FINAL);
```
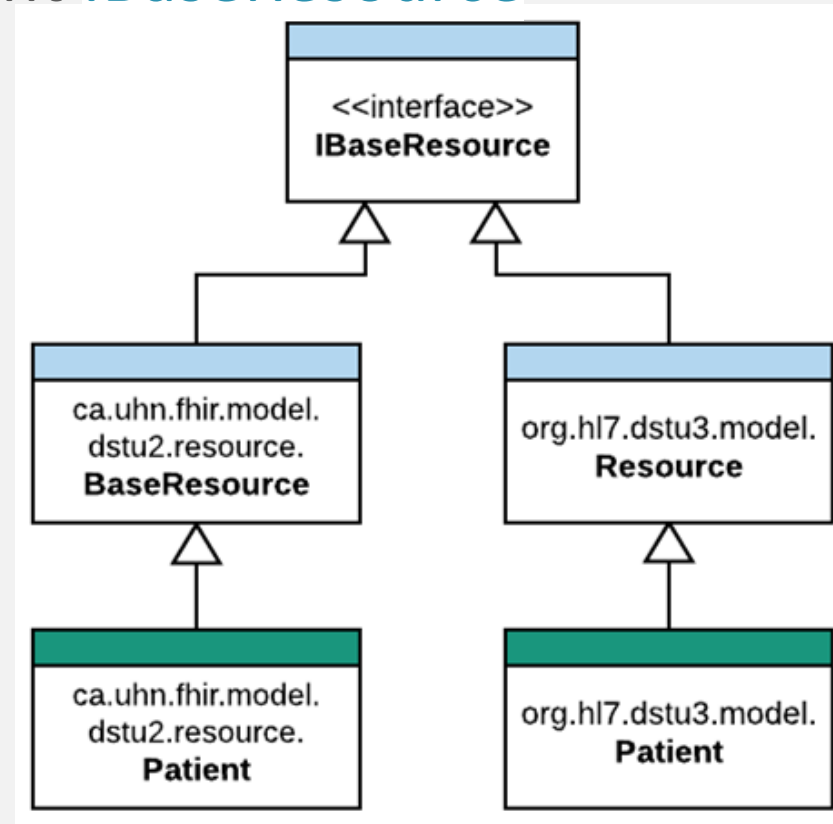
Datatypes

In Java

- Primitive classes are named [name]Type

- Primitive types include:
  StringType, BooleanType

- Composite types include:
  Address, Ratio, HumanName

In C#

Datatypes example

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

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

Using datatypes

C#

```csharp
// Add an "identifier" element
var identifier = new Identifier("http://example.org", "123456");
obs.Identifier.Add(identifier);
```

Java

```java
// Add an "identifier" element
Identifier identifier = obs.addIdentifier();
identifier.setSystem("http://example.org").setValue("123456");
```

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.Fhi

public bool? Active {  }
```

```
var pat = new Patient();

pat.ActiveElement = new FhirBoolean(true);

pat.Active = true;
```

| Patient | setActive(boolean value) |
| Patient | setActiveElement(BooleanType value) |

Why would you use the non-primitive version?

★
```
var name = new HumanName();
```

```
public string Family { get; set; }
```

★
```
name.Family = "Baltus-Bakker";
```

- Adding extensions cannot be done on primitives!

★
```
name.FamilyElement.AddExtension(
    "http://hl7.org/fhir/StructureDefinition/humanname-partner-name",
    new FhirString("Baltus"));
```

Choice properties


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

Resource components


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

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

Parsing/serializing

```
using Hl7.Fhir.Serialization;
```

```
import ca.uhn.fhir.context.FhirContext;
```

The context (HAPI)

- The starting point for much of the HAPI-FHIR API is the **FhirContext** class

- FhirContext acts as a factory for the rest of the API, including the two parsers:
  - XmlParser
  - JsonParser

- FhirContext is designed to be created once and reused (important for performance!)

Parsing/serializing example Java

```java
// Create a context
FhirContext ctx = FhirContext.forDstu3();
// 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);
```

Parsing/serializing example C#

```csharp
// 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 an XML string
var serializer = new FhirXmlSerializer();
var xmlText = serializer.SerializeToString(new_obs);
```

Working with REST

```
using Hl7.Fhir.Rest;
```

```
import ca.uhn.fhir.rest.client.api.IGenericClient;
```

REST recap

- FHIR defines basic CRUD operations that can be performed on a FHIR compliant server *(*not a complete list)*

Name	HTTP URL
type create	POST http://base/[type]
instance read	GET http://base/[type]/[id]
instance update	PUT http://base/[type]/[id]
instance delete	DELETE http://base/[type]/[id]
type search	GET http://base/[type]?[params]

Using the FHIR client

- See Publicly Available FHIR Servers for available test servers

C#

```
var client = new FhirClient("https://vonk.fire.ly");
```

Java

```
// Create a client
String serverBaseUrl = "http://fhirtest.uhn.ca/baseDstu3";
IGenericClient client = ctx.newRestfulGenericClient(serverBaseUrl);
```

Clients: Two Distinct Flavours in HAPI FHIR

Annotation

```java
public interface SampleClient extends IRestfulClient {

    @Create
    MethodOutcome create(@ResourceParam Patient thePatient);

    @Read
    Patient read(@IdParam IdType theId);

}
```

- You create an annotated interface for your specific needs
- Similar to JAX-RS or Spring REST (but does not use these frameworks)

Generic/Fluent

```java
MethodOutcome outcome = client
    .create()
    .resource(pat)
    .execute();
```

- Use chained method calls to do anything you need
- This is generally easier and more popular

Clients: Two Distinct Flavours in HAPI FHIR

Annotation

```
public interface SampleClient extends IRestfulClient {

  @Create
  MethodOutcome create(@ResourceParam Patient thePatient);

  @Read
  Patient read(@IdParam IdType theId);

}
```

Docs:
http://hapifhir.io/
doc_rest_client_annotation.html

Generic/Fluent

```
MethodOutcome outcome = client
    .create()
    .resource(pat)
    .execute();
```

Docs:
http://hapifhir.io/doc_rest_client.html

Create interaction example Java

```java
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());
```

Create interaction example C#

```csharp
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.fire.ly");

// 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);
```

Searching

- FHIR defines a powerful search mechanism

- Searches are specially crafted URLs to express queries such as:
  - Find a Patient with the given Identifier
  - Find all Patients with given gender and DOB
  - Find all lab reports for a given patient identifier with an "abnormal" interpretation

- Searching is powerful!
  - Learn about it at http://hl7.org/fhir/search.html

Searching (2)

- For now, let's imagine a search for a Patient named "Test" whose birthdate is before 2014

GET http://fhirtest.uhn.ca/baseDstu3/Patient?name=Test&birthdate=lt2014-05-10

Example search in Java

Example search in C#

```csharp
// Build a search and execute it
var q = new SearchParams()
            .Where("name=Test")
            .LimitTo(100);
q.Add("birthdate", "lt2014-01-01");

Bundle results = client.Search<Patient>(q);

// How many resources did we find?
Console.WriteLine("Number of results: " + results.Total);

// Print the ID of the first one
Console.WriteLine("First result id: " + results.Entry.First().Resource.Id);
```

Client interceptors (Java)

- Interceptors "wrap" each client operation and can:
  - Examine outgoing requests before they happen
  - Change outgoing requests before they happen
  - Examine incoming responses after they happen

- Interceptors recently changed and no longer require an interface
  - For example:
    BasicAuthInterceptor and **BearerTokenAuthInterceptor**
    - Add credentials to request

- See http://hapifhir.io/doc_rest_client_interceptor.html

Old Interceptor

New Interceptor

```
public class CorsInterceptor extends
InterceptorAdapter {

  @Override
public boolean incomingRequestPreProcessed(
    HttpServletRequest theRequest,
    HttpServletResponse theResponse) {

  return true;
}
```

```
@Interceptor
public class AuthorizationInterceptor {

|
@Hook(Pointcut.SERVER_INCOMING_REQUE
ST_PRE_HANDLED)
 public boolean checkAuthorization(
      RestOperationTypeEnum theOperation,
      IServerInterceptor.ActionRequestDetails
theProcessedRequest) {
   return true;
}
```

Adding headers to the request in C#

```csharp
        client.OnBeforeRequest +=
            (object sender, BeforeRequestEventArgs e) =>
            {
                e.RawRequest.Headers.Add("some_key", "some_value");
            };


    client.OnAfterResponse +=
        (object sender, AfterResponseEventArgs e) =>
        {
            Console.WriteLine(e.RawResponse.StatusCode);
        };
```

Questions?

Starter Projects (.NET & HAPI FHIR)
https://github.com/FirelyTeam/fhirstarters

And now for some coding

Grab a quick coffee
and
return for the Let's Build session!