

FHIR API for .Net programmers - an introduction

Mirjam Baltus, Firely



HL7 FHIR DevDays 2020, Virtual Edition US, June 15–18, 2020 | @HL7 @FirelyTeam | #fhirdevdays | www.devdays.com/us

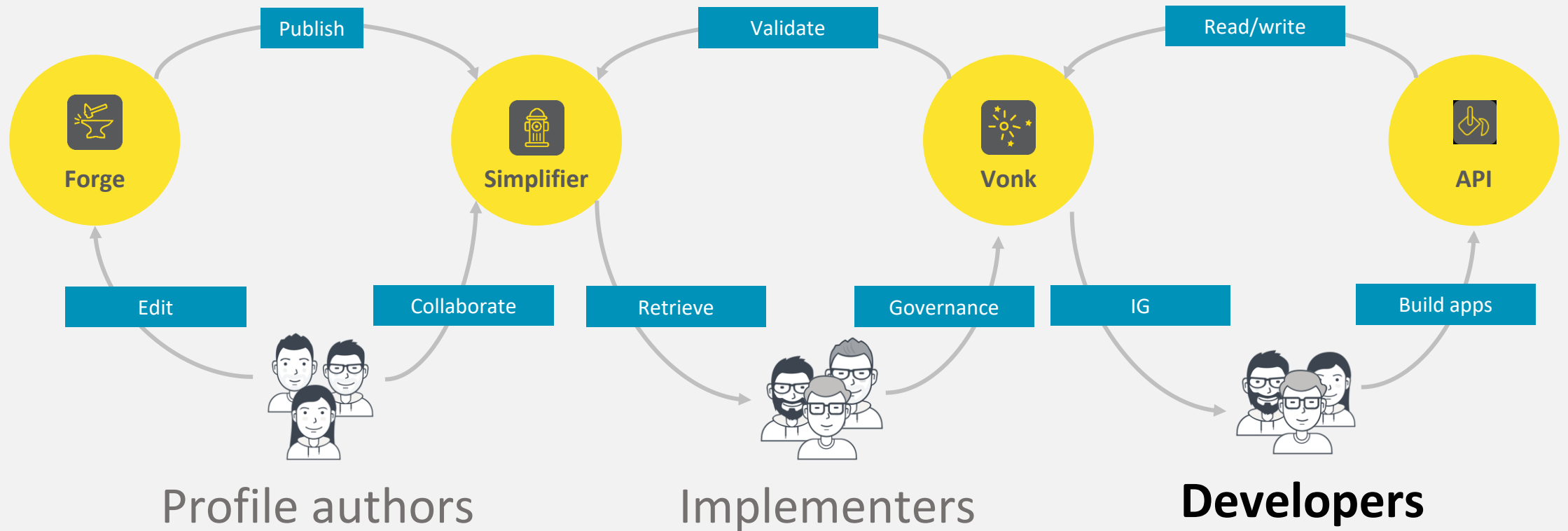
Who am I?

- Name: Mirjam Baltus
- Company: Firely, Amsterdam
- Background:
 - FHIR team since 2012
 - FHIR trainer
- Slides: bit.ly/MirjamsTraining

firely



The Firely FHIR ecosystem

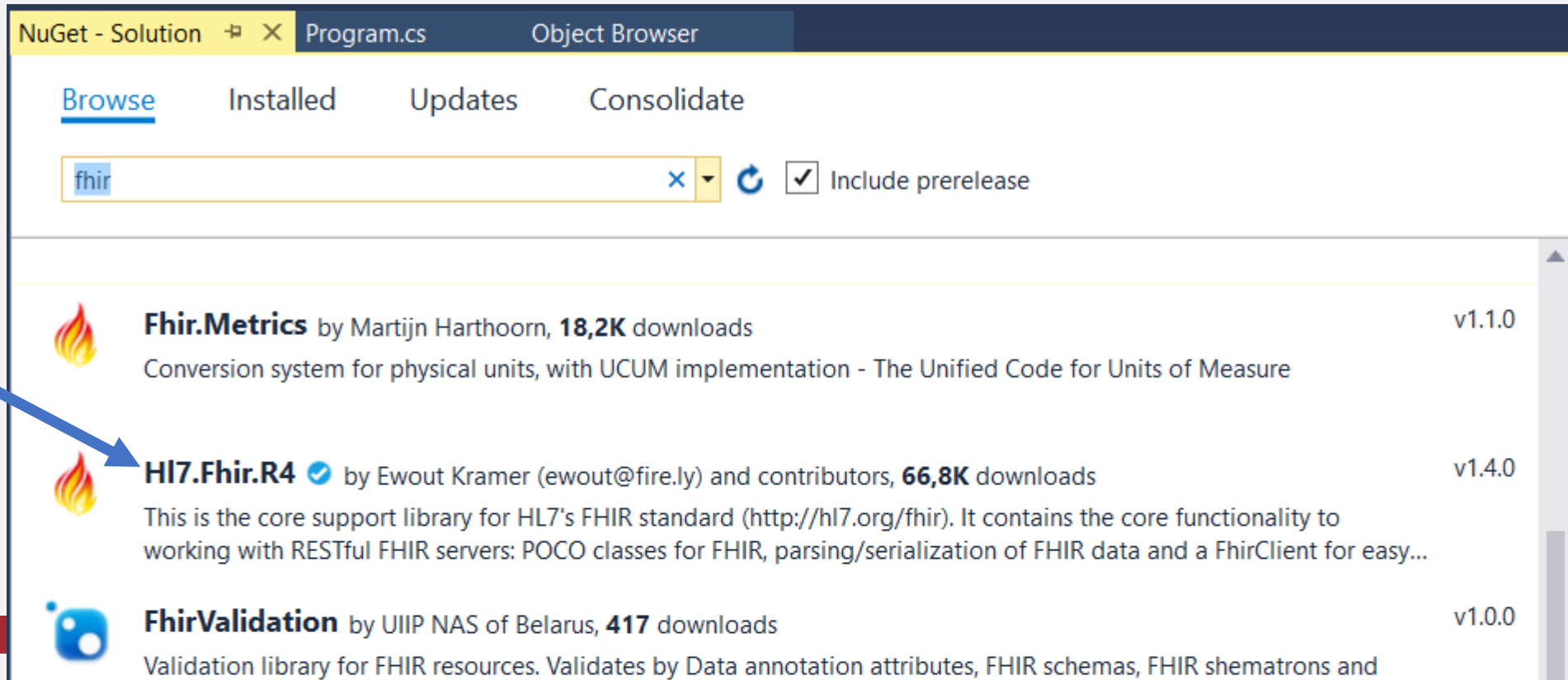


Using the Reference Implementation for .Net

HL7.Fhir API

First step





- Adding the HL7.Fhir package to your solution
 - NuGet Package manager, HL7.Fhir.R4 package



NuGet - Solution Program.cs Object Browser

Browse Installed Updates Consolidate

fhir Include prerelease

	Fhir.Metrics by Martijn Harthoorn, 18,2K downloads Conversion system for physical units, with UCUM implementation - The Unified Code for Units of Measure	v1.1.0
	HL7.Fhir.R4  by Ewout Kramer (ewout@fire.ly) and contributors, 66,8K downloads This is the core support library for HL7's FHIR standard (http://hl7.org/fhir). It contains the core functionality to working with RESTful FHIR servers: POJO classes for FHIR, parsing/serialization of FHIR data and a FhirClient for easy...	v1.4.0
	FhirValidation by UIIP NAS of Belarus, 417 downloads Validation library for FHIR resources. Validates by Data annotation attributes, FHIR schemas, FHIR shematrons and	v1.0.0

HL7.Fhir.STU3/R4/R5(beta)

- Core contents
 - Model – classes generated from the spec
 - REST functionality – FhirClient
 - Parsers and Serializers
 - Helper functions
- Source on GitHub: <https://github.com/FirelyTeam/fhir-net-api>
- Some documentation on: <http://docs.simplifier.net/fhirnetapi>

The model

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

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 in C# - classes and enums

```
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 = ObservationStatus.Preliminary;`

A FHIR Resource in C# - datatypes and lists

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

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

★ `obs.Code = new CodeableConcept("http://example.org", "EX123", "Example code 123");`

 identifier	Σ	0..*	Identifier	Business Identifier for observation
--	---	------	------------	-------------------------------------

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

★ `obs.Identifier.Add(new Identifier("http://example.org", "123456"));`

A FHIR Resource in C# - components

component	Σ	0..*	BackboneElement	Component results
code	Σ	1..1	CodeableConcept	Type of component observation (code / type) LOINC Codes (Example)
value[x]	Σ	0..1		Actual component result
valueQuantity			Quantity	
valueCodeableConcept			CodeableConcept	
valueString			string	
valueBoolean			boolean	
valueInteger			integer	
valueRange			Range	

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

- ★ `var component = new Observation.ComponentComponent();`
`// fill the values`
- ★ `obs.Component.Add(component);`

A FHIR Resource in C# - choice properties

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

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

- ★ `var qty = new Quantity {`
- ★ `Value = 25,`
- ★ `Unit = "sec",`
- ★ `System = "http://unitsofmeasure.org",`
- ★ `Code = "s" };`

- ★ `component.Value = qty;`

A FHIR Resource in C# - (non) primitives

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

```

/// <summary>
/// registered | preliminary | final | amended +
/// </summary>

```

```
public Hl7.Fhir.Model.ObservationStatus? Status { ... }
```

```
public Code<Hl7.Fhir.Model.ObservationStatus> StatusElement { ... }
```

★ `obs.Status = ObservationStatus.Preliminary;`

`// is the same as`

★ `obs.StatusElement = new Code<ObservationStatus>(ObservationStatus.Preliminary);`

Why would you use the non-primitive version?

★ `var name = new HumanName();`

```
public IEnumerable<string> Given { get; set; }
```

★ `name.Given = new string[] { "Mirjam" }; // or`

★ `name.GivenElement.Add(new FhirString("Mirjam"));`

★ `name.Family = "Baltus-Bakker";`

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

- Adding extensions cannot be done on primitives!

★ `name.FamilyElement.AddExtension(`

```
    "http://hl7.org/fhir/StructureDefinition/humanname-partner-name",
```

```
    new FhirString("Baltus"));
```

REST interactions

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

Using the FHIR Client

- For a list of test servers, see [Publicly Available FHIR Servers](#)

★ `var client = new FhirClient("https://vonk.fire.ly/r4");`

`// client options`

★ `client.PreferredFormat = ResourceFormat.Xml;`

★ `client.PreferredReturn = Prefer.ReturnRepresentation;`

C(RUD)

```
★ var obs = new Observation();
★ obs.Status = ObservationStatus.Preliminary;
★ obs.Code = new CodeableConcept("http://example.org", "EX123",
                                "Example code 123");
// fill in mandatory fields, plus other fields you have data for

// send the observation to the server to be created
★ var result = client.Create<Observation>(obs);

// note that this could generate an error,
// so setup error handling to catch exceptions
```

(C)RUD

```
// read a resource from the server
```

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

```
// update a resource on the server
```

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

```
★ client.Update<Patient>(pat);
```

```
// delete a resource from the server
```

```
★ client.Delete(pat); // or
```

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

Bundles and searches

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

Making queries

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

Paging through a Bundle

```
★ 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);
★ }
```

Questions?

ORGANIZED BY



PARTNER



HOST SPONSOR

