



November 20-22, 2019 | Amsterdam | NL

## Let's Build! Clinical modelling with FHIR

Track: Clinical Modeling  
Session lead: **David Hay**  
Email: [david.hay25@gmail.com](mailto:david.hay25@gmail.com)

### Purpose

Use modules from clinFHIR to show how FHIR uses resources to build a Problem list, and how to adapt a core resource type to store additional information

### Prerequisites

- A modern browser and OS (we use chrome on Mac)
- An internet connection
- Some knowledge of FHIR Conformance Resources and profiling

### Exercises (details on next pages)

1. Create a graph to represent a patients Problem list, and save it to the data server
2. View the resources on the Data server using Patient Viewer
3. Create a model to represent a modified (aka profiled) Condition in Logical Modeler
4. Update the graph to add a condition using that model and save to the FHIR server.  
The view in the Patient Viewer

### Configuration

There are instructions in the exercises to set the required servers for this tutorial, in summary these are:

#### clinFHIR servers

- |                      |                             |
|----------------------|-----------------------------|
| • Data server        | Public Hapi R4 server       |
| • Conformance Server | nzProjectR4                 |
| • Terminology server | Ontoserver R4 (terminology) |

#### Graph Builder

Use the DevDays event

### Additional information

- Notes on Graph Builder: <http://bit.ly/2Kub06Z>



**Exercise 1: Create a Problem List graph**

The task is to create a graph that represents the Problem List for John Doe, a 19 year old male who has both asthma and diabetes. The list was created by Dr Mary Welby.

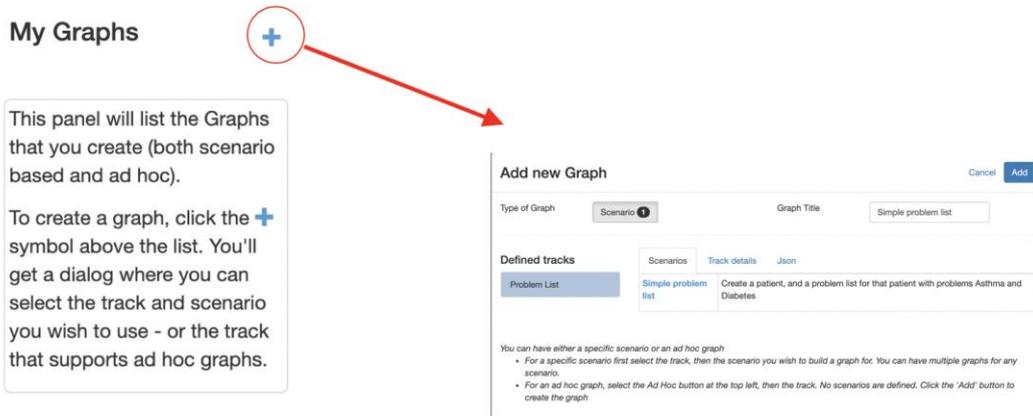
**Step 1: Start the graph builder.**

Start the graph builder by loading clinFHIR from <http://clinfhir.com/> and selecting the Graph Builder option. When prompted, select the devdays event. Click the login icon at the upper right and use your email to login.

*Log in using this icon  at the upper right. If this is your first login then enter your email and a password and click 'login'. You'll get a message dialog, then another one asking if you want to register. Select 'yes'.*

**Step 2: Create a new graph**

Click the plus symbol above the empty graph list. In the dialog that is displayed, select the 'Problem List' track, then the 'Simple problem list' scenario. (You may want to try the other scenarios at a later time). Optionally give the list a different name. Click the Add button at the upper right.



**Step 3: Add resources to the graph.**

Using the palette to the upper right, select the resources you will need. Give each resource a name so that you can easily locate it in the graph builder (This name is just used in the Graph Builder – it is not part of the resource). You will need the following resources:

November 20-22, 2019 | Amsterdam | NL

- A Patient (Joan Doe)
- A Practitioner (Mary Welby).
- A List (the Problem list itself)
- 2 Condition resources



If the resource type you want is not in the list, then click the ‘>Select Direct<’ link to get a complete list of types – you can any type you want to.

**Step 4: connect the resources together**

This is the process of creating references from one resource to another – e.g. from Condition to Patient. Remember that references are always one way. For each reference:

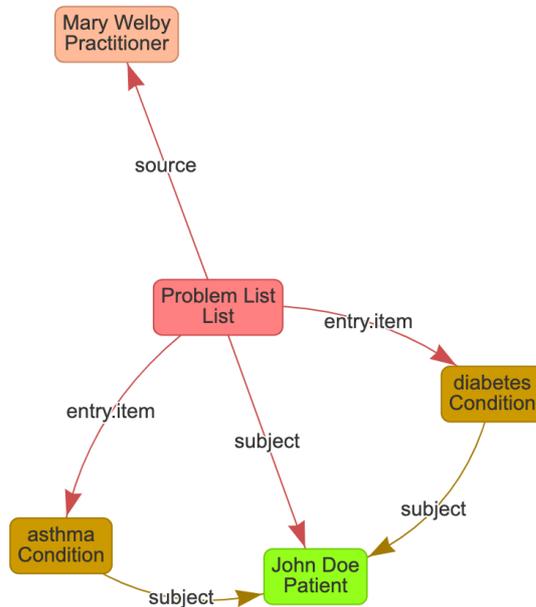
- Select the source resource (eg Condition) – the one that the reference is from. It pays to have the graph view open as you do this, as you will see the references appear in real-time.
- In the table editor, find the element that represents the reference (e.g. subject). It will have the possible types that you can link to in green. Click the type you want.
- If there is already a single resource of that type in the graph, then you will be asked if you want to link to it. If there is more than one (as will be the case when creating references from the List to the Conditions for example) then you will get a list, and you choose which one you want. If there are no resources of this type you can add one directly.

severity	0..1	CodeableConcept	
code	0..1	CodeableConcept	
bodySite	0..*	CodeableConcept	
subject	1..1	<a href="#">--&gt; Patient</a> <a href="#">--&gt; Group</a>	
encounter	0..1	<a href="#">--&gt; Encounter</a>	
onset[x]	0..1	<input type="radio"/> dateTime <input type="radio"/> Age <input type="radio"/> Period <input type="radio"/> Range <input type="radio"/> string	
abatement[x]	0..1	<input type="radio"/> dateTime <input type="radio"/> Age	

For items like List where you need to make copies of an element to make references to multiple resources, click the + icon to make the copy. The new item is always the last one in the set of elements of that name.

>	entry		0..*	BackboneElement
>	entry		0..*	BackboneElement
>	entry		0..1	CodeableConcept

At the end of the process the graph should look like this: (though the resource instances may be in different places. Of course, you can always move them around...).



### Step 5: Add the structured data to the resources

Now you can add the structured data to the resources. To do this:

- Select the resource in the list or graph.
- In the table editor, click the edit icon for the element you wish to add. (see image below). You'll get a dialog that is specific to the data type, and contains fields for the various parts of the element.
- Most are simple entry fields, but the CodeableConcept allows you to select from the terminology server using a filter string (like 'asthma') as shown in the image

**code (CodeableConcept)** Cancel Save

asthma  ValueSet Uri: <http://hl7.org/fhir/ValueSet/condition-code>  
 Binding: example  
 Terminology Svr: <https://r4.ontoserver.csiro.au/fhir/>

List	Json		
Asthma		195967001	<a href="http://snomed.info/sct">http://snomed.info/sct</a>
Acute asthma		304527002	<a href="http://snomed.info/sct">http://snomed.info/sct</a>
Allergic asthma		389145006	<a href="http://snomed.info/sct">http://snomed.info/sct</a>
Asthma finding		390798007	<a href="http://snomed.info/sct">http://snomed.info/sct</a>
Asthma resolved		162660004	<a href="http://snomed.info/sct">http://snomed.info/sct</a>
Bakers' asthma		34015007	<a href="http://snomed.info/sct">http://snomed.info/sct</a>

You don't have to do all the data for all the resources before saving – it is quite possible to add some data to some resources, save, then repeat. The same is for resources which can be added or removed at any time. In fact, we'll be coming back to this graph later in the session.

You don't need to save the graph – this happens automatically.



November 20-22, 2019 | Amsterdam | NL

*In order to be able to find your patient in the Patient Viewer, create a distinct name – like your initials before the name – eg last name = dh-doe, first name john*

You should also take a look at the Resource instance and validation tab. This shows what the resource instance will look like ‘on the wire’ (i.e. when shared between systems) and there’s also a ‘validate’ link that allows you to check it using the FHIR validator.

*Technical note: The graph details are saved in the clinFHIR database. The resources are only sent to the server when you instruct it.*

### Step 6: Save the graph to the server

Finally, you can save your graph to the data server! This is done by selecting the ‘server’ tab, then pressing the ‘send to server’ button. This will cause the resources to be sent to the data server. If the resources are incomplete, they will still be saved – you can use the ‘Validate all’ link to check this first.

If there are any incomplete ones, selecting the resource in the tab will display the editor to the right so you can make changes directly.

List Graph Notes **Save to server** View Transaction bundle

Data server: <http://hapi.fhir.org/baseR4/> [Validate all](#) [Send to server](#)  
[Add to Patient Viewer Scenario list](#)

Type	Title	Id	Valid
Patient	John Doe	id1573280787846	
List	Problem List	id1573280812077	
Condition	asthma	id1573280820849	
Condition	diabetes	id1573280826119	

### Final note

While this exercise has a specific order, in practice you can use any order you want – for example you might want to add the structured data immediately after adding the resource instance – or save the graph to the server, add a new resource or reference then save again. Each time you save a resource, a new version is created – you can look at previous versions in the Patient Viewer if you want.

November 20-22, 2019 | Amsterdam | NL

**Exercise 2: View the data using the Patient Viewer**

The Patient Viewer module can display resources on any FHIR server – including those created and saved by the Graph Builder. There are a number of views – some of which are similar to the Graph Builder. Let’s look at the data you just saved to the data server.

**Step 1: Start the Viewer**

- Load clinFHIR (<http://clin.fhir.com>)
- Set the servers as shown to the right (Click the edit link at the upper right to change a server)
- Log in using this icon  at the upper right. You should be able to use the login you created for exercise 1
- Select the Patient Viewer module

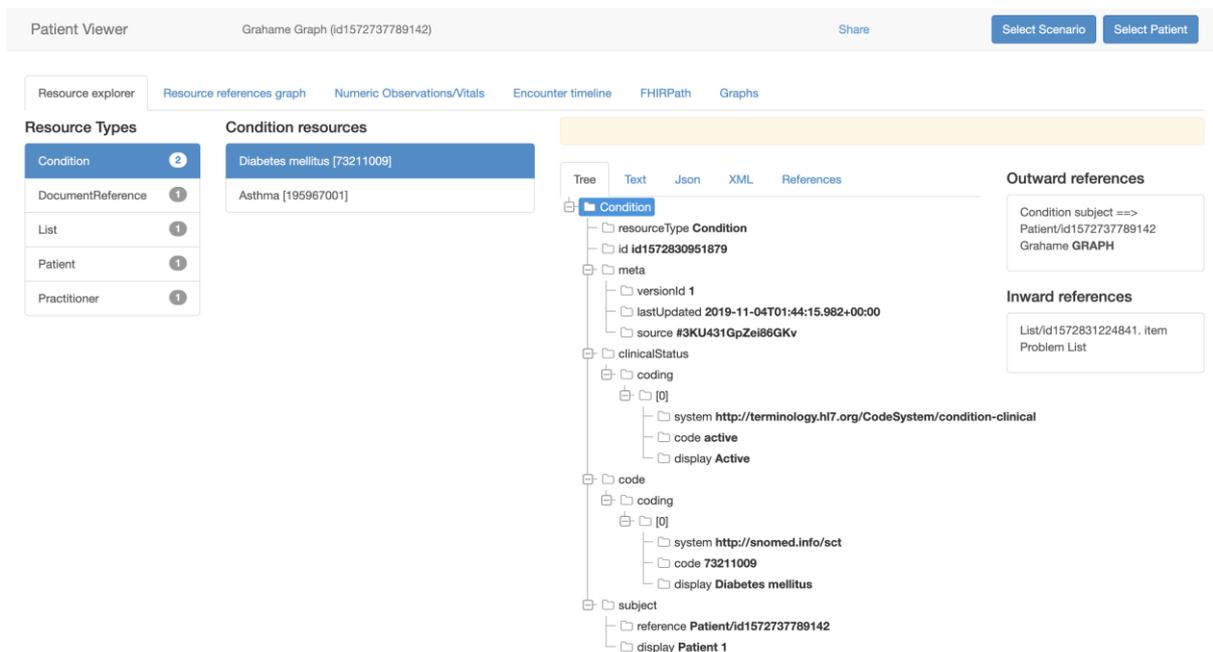
Current servers		Edit
<b>Data Server</b>	Public HAPI R4 server	
<b>Conformance Server</b>	nzProjectR4	
<b>Terminology Server</b>	OntoserverR4 (terminology)	
<a href="#">Add Server</a>		<a href="#">Set all the same as the Data Server</a>

**Step 2: Select the Patient**

Click the ‘select patient’ button to the upper right, then enter the name of your patient. All matching patients are displayed in a list (this is why you needed to enter the patient name when you created the graph).

**Step 3: View the resources**

There are a number of different ways of viewing the data – feel free to experiment. A simple way is to select the resource type in the list to the left, then an instance of that type. You’ll get a number of views of that resource in tabs to the right as shown in this screen shot:



The screenshot shows the Patient Viewer interface for a 'Graham Graph (id1572737789142)'. The 'Resource Types' list on the left has 'Condition' selected. The 'Condition resources' list shows 'Diabetes mellitus [73211009]' and 'Asthma [195967001]'. The main view displays a JSON tree for the selected 'Condition' resource, including details like 'resourceType Condition', 'id id1572830951679', 'meta' (versionId 1, lastUpdated 2019-11-04T01:44:15.982+00:00, source #3KU431GpZei86GKv), 'clinicalStatus', 'coding' (system http://terminology.hl7.org/CodeSystem/condition-clinical, code active, display Active), 'code' (system http://snomed.info/sct, code 73211009, display Diabetes mellitus), and 'subject' (reference Patient/id1572737789142, display Patient 1). On the right, 'Outward references' and 'Inward references' are also visible.

**Exercise 3: Modify a resource type**

In this exercise you will use the Logical Modeler to modify the base Condition resource to add a new element and remove others. This is called ‘profiling’ in FHIR, though we won’t be creating actual profiles – however the work we do can be used to do so. We’ll call this a ‘logical model’.

*Note that you must be logged in as described in exercise 2 or you won’t be able to create or edit models.*

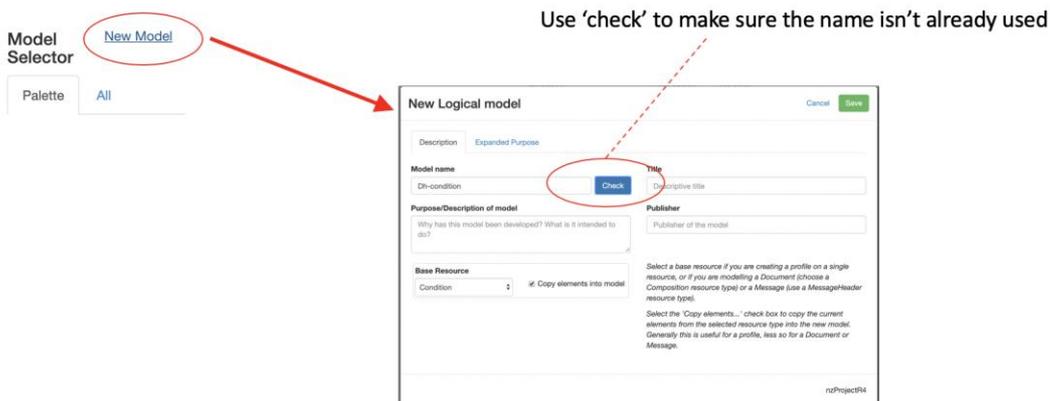
**Step 1: Load the Logical Modeler**

- Load clinFHIR
- Set the servers as shown to the right (They are the same as the Viewer – and the Graph builder, through the track you selected).
- Select the Logical Modeler

Current servers		Edit
<b>Data Server</b>	Public HAPI R4 server	?
<b>Conformance Server</b>	nzProjectR4	?
<b>Terminology Server</b>	OntoserverR4 (terminology)	?
Add Server		Set all the same as the Data Server

**Step 2: Create a new model**

Click the ‘new model’ to create a new model on the conformance server. Select the Condition resource as the base. You’ll need to provide a model name – try {initials}-Condition. Click the ‘check’ button to make sure no-one else has used this name.



**Step 3: Remove unwanted elements**

Select an item in the Designer tree, then click the ‘Remove Element’ link at the bottom right. For example, remove:

- Abatement
- Stage



- Evidence

### Step 4: Add the new one

To add a new element, first select the first node in the tree (also called the root node).

Next click the 'Add Element' link that appears to the bottom right. A dialog will appear where you can enter the details of the new element. See the image below for an example.

Note that it is important to set the FHIR mapping path to 'Condition.extension' and to add the url for the extension in the model. It doesn't really matter (for now) what it is – but it is needed for the graph builder to know that this element needs to be represented as an extension – as we'll see in exercise 4.

Here's a screenshot:

### Step 5: Make a note of the model Url

When you use this model in the Graph Builder for the next exercise, you're going to need the url for model. This is found by selecting the 'Model' tab to the right, then make a note of the Model Url (just copy it and paste into a text editor. It will be something like:

November 20-22, 2019 | Amsterdam | NL

<http://home.clinfhir.com:8054/baseR4/StructureDefinition/DhCondition>

*Technical note: This url is actually the RESTful location of the StructureDefinition resource that defines the logical model. It is not a canonical url. Right now, that definition does not exist.*

#### Step 6: Save the model

Unlike the Graph Builder you need to specifically save the model using the 'Save' link that appears to the upper right when you have changed the model.



**Exercise 4: Use the logical model in the problem list graph**

Now that you have a customized model, you can create instances of that model in your graph. The process is very similar to using ‘core’ types – the only difference being that you select the model from a different tab, and you need the url to the model. The act of connecting the resource instance to others and populating the structured data is exactly the same.

**Step 1: Load your graph**

Use the instructions in exercise 1 to load the graph (of course, you don’t need to create a new graph – just select the graph from the list).

**Step 2: Select the logical model**

In the palette to the upper right, select the tab labelled ‘Logical Models’, then the link ‘>Select Direct<’ in the tab contents. A dialog will be shown where you enter the url to the model and give it a description. Click ‘Add’ then you will be prompted for a name to use in the graph (exactly as when you add a core resource instance) and the new instance is added to the graph.

**Add Logical Model**
Cancel **Add**

From Server

Link

Description

The defined conformance server for this track is: <http://home.clinfhir.com:8054/baseR4/>

**Important!** At the moment, the Logical Model must be one developed by the clinFHIR Logical Modeller, and one that is based on a Core FHIR resource, as the builder needs this information to create the references graph. Importing from external profiles is planned.

(You can also select the model from the dropdown – though that can get quite large)

**Step 3: Connect to the other resources and add structured data**

This is exactly the same as for any other resource instance. You should see the new element you added in the table editor, and if you add data to that element and look at the resource instance json, you should see that it is represented as an extension.

**Step 4: Save the updated graph**

See exercise 1. Remember, a graph can be saved any number of times. However, do note that if you delete a resource instance from the graph, it won’t be deleted from the server.

**Step 5: View in the patient viewer.**

See exercise 2

Have fun and remember to ask for help if you get stuck! clinFHIR will still be available after devdays has finished – use chat.fhir.org to ask for help.