ClinFHIR Tutorial

Viet Nguyen, MD
Stratametrics LLC
vietnguyen@Stratametrics.com
Initial Setup

- Chrome Browser
- Use the Zoom in/out function (CTRL-) if you don’t see buttons
- Keep a notepad handy to jot down information
<table>
<thead>
<tr>
<th>Main modules (open in new tab)</th>
<th>Experimental modules (open in new tab)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Patient Viewer</strong></td>
<td><strong>Server Query</strong></td>
</tr>
<tr>
<td>Display resources for a specific patient, using a number of different views such as a list by resource type, json &amp; tree views, encounters by condition, numeric Observation charting and graphical relationship views. There is also the option to add a new patient, and to create sample data for that patient.</td>
<td>Supports ad hoc queries against any FHIR server. Includes a simple query builder. The response can be displayed as Json or a Tree view, and FHIRPath is supported.</td>
</tr>
<tr>
<td><strong>Scenario Builder</strong></td>
<td><strong>Conformance Server</strong></td>
</tr>
<tr>
<td>The Scenario Builder is used to join together the resources needed to represent a specific clinical scenario. It can use Core Resource types, Profiles and Logical models as it does this. The intention is to help people understand how resources can tell a clinical story, and to validate that the resource types available (including profiles) are sufficient. Note that the builder still has issues with more complex resource types - this is a work in progress.</td>
<td>Can access any compliant FHIR server (must expose a Capability Statement)</td>
</tr>
<tr>
<td><strong>Logical Modeller</strong></td>
<td><strong>Terminology Server</strong></td>
</tr>
<tr>
<td>The Logical Modeller allows the creation of a model that represents a particular interoperability requirement in a format that is easy to use. It uses FHIR datatypes, and can be based on an existing resource type or completely 'ad hoc'. It is intended to act as a 'bridge' between Modeller and User, and can act as the basis for the generation of the profiling components required by FHIR.</td>
<td>Patient information is on the Data Server. Profiles on the Conformance server. ValueSets on the Terminology server.</td>
</tr>
<tr>
<td><strong>Implementation Guide Browser</strong></td>
<td><strong>Create a simple scenario</strong></td>
</tr>
<tr>
<td>Display the contents of an Implementation Guide, and the relationships between the contents of the Guide.</td>
<td>Create a simple scenario</td>
</tr>
<tr>
<td><strong>Extension Definition builder</strong></td>
<td><strong>Create an Information Model</strong></td>
</tr>
<tr>
<td>Views and builds extension definitions. These can be defined and applied to the Logical Model, which will allow them to be included in the generated Profile.</td>
<td>Create an Information Model</td>
</tr>
<tr>
<td><strong>CodeSystem builder</strong></td>
<td><strong>Create a Resources Model</strong></td>
</tr>
<tr>
<td>The CodeSystem defines a set of Concepts from which a ValueSet provides possible values for a resource element. The actual 'binding' between CodeSystem and element is done by the ValueSet. This component allows you to build (and edit) a CodeSystem, and optionally builds the ValueSet as well.</td>
<td>Create a Resources Model</td>
</tr>
<tr>
<td><strong>ValueSet explorer</strong></td>
<td><strong>ValueSets are stored on the Terminology Server</strong></td>
</tr>
<tr>
<td>Lets you view existing ValueSets. The builder works best with SNOMED (at the moment).</td>
<td>ValueSets are stored on the Terminology Server</td>
</tr>
</tbody>
</table>

**Current servers**

<table>
<thead>
<tr>
<th>Data Server</th>
<th>Conformance Server</th>
<th>Terminology Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public HAPI STU3 server</td>
<td>Public HAPI STU3 server</td>
<td>Public HAPI STU3 server</td>
</tr>
</tbody>
</table>

Add Server: Set all the same as the Data Server

**FHIR Links**

- STU-3 (R3) Specification
- STU-2 Specification
- FHIR wiki
- FHIR Chat
- FHIR.org
- Clinicians Workshop

**clinFHIR Videos**

- Scenario Builder
- Logical Modeller
- Logical Modeller and Scenario Builder
- RESTful query tool

*Note that some of these videos may describe earlier versions, so may not completely match the current functionality.*

**Other links**

- SNOMED browser
<table>
<thead>
<tr>
<th>Main modules (open in new tab)</th>
<th>Experimental modules (open in new tab)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Patient Viewer</strong></td>
<td>Display resources for a specific patient, using a number of different views such as a list by resource type, JSON &amp; tree views, encounters by condition, numeric Observation charting and graphical relationship views. There is also the option to add a new patient, and to create sample data for that patient.</td>
</tr>
<tr>
<td><strong>Server Query</strong></td>
<td>Supports ad hoc queries against any FHIR server. Includes a simple query builder. The response can be displayed as JSON or a Tree view, and FHIRPath is supported.</td>
</tr>
<tr>
<td><strong>Scenario Builder</strong></td>
<td>The Scenario Builder is used to join together the resources needed to represent a specific clinical scenario. It can use Core Resource types, Profiles and Logical models as it does this. The intention is to help people understand how resources can tell a clinical story, and to validate that the resource types available (including profiles) are sufficient. Note that the builder still has issues with more complex resource types - this is a work in progress</td>
</tr>
<tr>
<td><strong>Logical Modeller</strong></td>
<td>The Logical modeller allows the creation of a model that represents a particular interoperability requirement in a format that is easy to use. It uses FHIR data types, and can be based on an existing resource type or completely 'ad hoc'. It is intended to act as a 'bridge' between Modeller and User, and can act as the basis for the generation of the profiling components required by FHIR. Models are saved on the Conformance Server. Can reference ValueSets from the Terminology Server.</td>
</tr>
<tr>
<td><strong>Implementation Guide Browser</strong></td>
<td>Display the contents of an Implementation Guide, and the relationships between the contents of the Guide. The Implementation guide, profiles and Extension Definitions are on the Conformance Server, the terminology resources (e.g. ValueSet) are on the Terminology Server.</td>
</tr>
<tr>
<td><strong>Extension Definition builder</strong></td>
<td>Views and builds extension definitions. These can be defined and applied to the Logical Model, which will allow them to be included in the generated Profile Extension definitions are saved on the Conformance Server.</td>
</tr>
<tr>
<td><strong>CodeSystem builder</strong></td>
<td>The CodeSystem defines a set of Concepts from which a ValueSet provides possible values for a resource element. The actual 'binding' between CodeSystem and element is done by the ValueSet. This component allows you to build (and edit) a CodeSystem, and optionally builds the ValueSet as well. CodeSystems are saved on the Terminology Server.</td>
</tr>
<tr>
<td><strong>ValueSet explorer</strong></td>
<td>Lets you view existing ValueSets. The builder works best with SNOMED (at the moment). ValueSets are stored on the Terminology Server.</td>
</tr>
<tr>
<td><strong>Data Server</strong></td>
<td>Public HAPI STU3 server</td>
</tr>
<tr>
<td><strong>Conformance Server</strong></td>
<td>Public HAPI STU3 server</td>
</tr>
<tr>
<td><strong>Terminology Server</strong></td>
<td>Public HAPI STU3 server</td>
</tr>
<tr>
<td><strong>Add Server</strong></td>
<td>Set all the same as the Data Server</td>
</tr>
</tbody>
</table>

**FHIR Links (open in new tab)**
- STU-3 (R3) Specification
- STU-2 Specification
- FHIR wiki
- FHIR Chat
- FHIR.org
- Clinicians Workshop

**clinFHIR Videos (open in new tab)**
- Scenario Builder
- Adding structured data
- Logical Modeller
- Logical Modeller and Scenario Builder
- RESTful query tool

Note that some of these videos may describe earlier versions, so may not completely match the current functionality.

**Other links**
- SNOMED browser
## Modules

### Main modules (open in new tab)
- **Patient Viewer**
  - Display resources for a specific patient, using a number of different views such as a list by resource type, json & tree views, encounters by condition, numeric Observation charting and graphical relationship views.
  - There is also the option to add a new patient, and to create sample data for that patient.
- **Server Query**
  - Supports ad hoc queries against any FHIR server. Includes a simple query builder. The response can be displayed as Json or a Tree view, and FHIRPath is supported.
- **Scenario Builder**
  - The Scenario Builder is used to join together the resources needed to represent a specific clinical scenario. It can use Core Resource types, Profiles and Logical models as it does this.
  - The intention is to help people understand how resources can tell a clinical story, and to validate that the resource types available (including profiles) are sufficient.
  - Note that the builder still has issues with more complex resource types - this is a work in progress.
- **Logical Modeller**
  - The Logical modeller allows the creation of a model that represents a particular interoperability requirement in a format that is easy to use. It uses FHIR datatypes, and can be based on an existing resource type or completely 'ad hoc'. It is intended to act as a 'bridge' between Modeller and User, and can act as the basis for the generation of the profiling components required by FHIR.
- **Implementation Guide Browser**
- **Extension Definition builder**
  - Views and builds extension definitions. These can be defined and applied to the Logical Model, which will allow them to be included in the generated Profile.
- **CodeSystem builder**
  - The CodeSystem defines a set of Concepts from which a ValueSet provides possible values for a resource element. The actual 'binding' between CodeSystem and element is done by the ValueSet. This component allows you to build (and edit) a CodeSystem, and optionally builds the ValueSet as well.
- **ValueSet explorer**
  - Lets you view existing ValueSets. The builder works best with SNOMED (at the moment).

### Experimental modules (open in new tab)
- Patient resources are stored on the Data Server. The server should support the Patient/$everything operation.
- Can access any compliant FHIR server (must expose a Capability Statement).

### Other links
- SNOMED browser
Server Selection

**Main modules (open in new tab)**

- **Patient Viewer**
  - Display resources for a specific patient, using a number of different views such as a list by resource type, json & tree views, encounters by condition, numeric Observation charting and graphical relationship views.
  - There is also the option to add a new patient, and to create sample data for that patient.

- **Server Query**
  - Supports ad hoc queries against any FHIR server. Includes a simple query builder. The response can be displayed as Json or a Tree view, and FHIRPath is supported.

- **Scenario Builder**
  - The Scenario Builder is used to join together the resources needed to represent a specific clinical scenario. It can use Core Resource types, Profiles and Logical models as it does this.
  - The intention is to help people understand how resources can tell a clinical story, and to validate that the resource types available (including profiles) are sufficient.
  - Note that the builder still has issues with more complex resource types - this is a work in progress

- **Logical Modeller**
  - The Logical modeller allows the creation of a model that represents a particular interoperability requirement in a format that is easy to use. It uses FHIR datatypes, and can be based on an existing resource type or completely 'ad hoc'. It is intended to act as a 'bridge' between Modelled and User, and can act as the basis for the generation of the profiling components required by FHIR

- **Implementation Guide Browser**

- **Extension Definition builder**
  - Views and builds extension definitions. These can be defined and applied to the Logical Model, which will allow them to be included in the generated Profile

- **CodeSystem builder**
  - The CodeSystem defines a set of Concepts from which a ValueSet provides possible values for a resource element. The actual 'binding' between CodeSystem and element is done by the ValueSet. This component allows you to build (and edit) a CodeSystem, and optionally builds the ValueSet as well.

- **ValueSet explorer**
  - Lets you view existing ValueSets. The builder works best with SNOMED (at the moment).

**Experimental modules (open in new tab)**

- **Patient resources**
  - Patient resources are stored on the Data Server. The server should support the Patient/$everything operation.

- **Can access any compliant FHIR server (must expose a Capability Statement)**

- **Patient information**

- **Models are saved on the Conformance Server. Can reference ValueSets from the Terminology Server**

**Add Server**

- **Data Server** Public HAPI STU3 server
- **Conformance Server** Public HAPI STU3 server
- **Terminology Server** Public HAPI STU3 server

**FHIR Links (open in new tab)**

- STU-3 (R5) Specification
- STU-2 Specification
- FHIR wiki
- FHIR Chat
- FHIR.org
- Clinicians Workshop

**clinFHIR Videos (open in new tab)**

- Scenario Builder
- Adding structured data
- Logical Modeller
- Logical Modeller and Scenario Builder
- RESTful query tool

**Other links**

- SNOMED browser
Useful FHIR Links

**Main modules (open in new tab)**

**Patient Viewer**
Display resources for a specific patient, using a number of different views such as a list by resource type, json & tree views, encounters by condition, numeric Observation charting and graphical relationship views.

There is also the option to add a new patient, and to create sample data for that patient.

**Server Query**
Supports ad hoc queries against any FHIR server. Includes a simple query builder. The response can be displayed as Json or a Tree view, and FHIRPath is supported.

**Scenario Builder**
The Scenario Builder is used to join together the resources needed to represent a specific clinical scenario. It can use Core Resource types, Profiles and Logical models as it does this. The intention is to help people understand how resources can tell a clinical story, and to validate that the resource types available (including profiles) are sufficient.

Note that the builder still has issues with more complex resource types - this is work in progress.

**Logical Modeller**
The Logical modeller allows the creation of a model that represents a particular interoperability requirement in a format that is easy to use. It uses FHIR datatypes, and can be based on an existing resource type or completely 'ad hoc'. It is intended to act as a 'bridge' between Modeller and User, and can act as the basis for the generation of the profiling components required by FHIR.

**Implementation Guide Browser**

**Extension Definition builder**
Views and builds extension definitions. These can be defined and applied to the Logical Model, which will allow them to be included in the generated Profile.

**CodeSystem builder**
The CodeSystem defines a set of Concepts from which a ValueSet provides possible values for a resource element. The actual 'binding' between CodeSystem and element is done by the ValueSet. This component allows you to build (and edit) a CodeSystem, and optionally builds the ValueSet as well.

**ValueSet explorer**
Lets you view existing ValueSets. The builder works best with SNOMED (at the moment).

**Experimental modules (open in new tab)**

**Patient**
Patient resources are stored on the Data Server. The server should support the Patient/$everything operation.

**Server Query**
Can access any compliant FHIR server (must expose a Capability Statement)

**Scenario Builder**

**Logical Modeller**
Models are saved on the Conformance Server. Can reference ValueSets from the Terminology server.

**Implementation Guide Browser**
The Implementation guide, profiles and Extension Definitions are on the Conformance Server, the terminology resources (eg ValueSet) are on the Terminology Server.

**Extension Definition builder**
Extension definitions are saved on the Conformance Server.

**CodeSystem builder**
CodeSystems are saved on the Terminology Server.

**ValueSet explorer**
ValueSets are stored on the Terminology Server.

**Current servers**
- Data Server: Public HAPI STU3 server
- Conformance Server: Public HAPI STU3 server
- Terminology Server: Public HAPI STU3 server

**Add Server**
Set all the same as the Data Server

**FHIR Links (open in new tab)**
- STU-3 (R5) Specification
- STU-2 Specification
- FHIR wiki
- FHIR Chat
- FHIR.org
- Clinicians Workshop

**clinFHIR Videos (open in new tab)**
- Scenario Builder
- Adding structured data
- Logical Modeller
- Logical Modeller and Scenario Builder
- RESTful query tool

Note that some of these videos may describe earlier versions, so may not completely match the current functionality.
# ClinFHIR Video Demos

<table>
<thead>
<tr>
<th>Main modules (open in new tab)</th>
<th>Experimental modules (open in new tab)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Patient Viewer</strong></td>
<td>Display resources for a specific patient, using a number of different views such as a list by resource type, JSON &amp; tree views, encounters by condition, numeric Observation charting and graphical relationship views. There is also the option to add a new patient, and to create sample data for that patient.</td>
</tr>
<tr>
<td><strong>Server Query</strong></td>
<td>Supports ad hoc queries against any FHIR server. Includes a simple query builder. The response can be displayed as JSON or a Tree view, and FHIRPath is supported.</td>
</tr>
<tr>
<td><strong>Scenario Builder</strong></td>
<td>The Scenario Builder is used to join together the resources needed to represent a specific clinical scenario. It can use Core Resource types, Profiles and Logical models as it does this. The intention is to help people understand how resources can tell a clinical story, and to validate that the resource types available (including profiles) are sufficient. Note that the builder still has issues with more complex resource types - this is a work in progress.</td>
</tr>
<tr>
<td><strong>Logical Modeller</strong></td>
<td>The Logical modeller allows the creation of a model that represents a particular interoperability requirement in a format that is easy to use. It uses FHIR datatypes, and can be based on an existing resource type or completely 'ad hoc'. It is intended to act as a 'bridge' between Modeller and User, and can act as the basis for the generation of the profiling components required by FHIR. Models are saved on the Conformance Server. Can reference ValueSets from the Terminology server.</td>
</tr>
<tr>
<td><strong>Implementation Guide Browser</strong></td>
<td>Display the contents of an Implementation Guide, and the relationships between the contents of the Guide. The Implementation guide, profiles and Extension Definitions are on the Conformance Server, the terminology resources (eg ValueSet) are on the Terminology Server.</td>
</tr>
<tr>
<td><strong>Extension Definition builder</strong></td>
<td>Views and builds extension definitions. These can be defined and applied to the Logical Model, which will allow them to be included in the generated Profile. Extension definitions are saved on the Conformance Server.</td>
</tr>
<tr>
<td><strong>CodeSystem builder</strong></td>
<td>The CodeSystem defines a set of Concepts from which a ValueSet provides possible values for a resource element. The actual 'binding' between CodeSystem and element is done by the ValueSet. This component allows you to build (and edit) a CodeSystem, and optionally builds the ValueSet as well. CodeSystems are saved on the Terminology Server.</td>
</tr>
<tr>
<td><strong>ValueSet explorer</strong></td>
<td>Lets you view existing ValueSets. The builder works best with SNOMED (at the moment). ValueSets are stored on the Terminology Server.</td>
</tr>
</tbody>
</table>

## Current servers

<table>
<thead>
<tr>
<th>Data Server</th>
<th>Public HAPI STU3 server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conformance Server</td>
<td>Public HAPI STU3 server</td>
</tr>
<tr>
<td>Terminology Server</td>
<td>Public HAPI STU3 server</td>
</tr>
</tbody>
</table>

- **Edit**: Set all the same as the Data Server

## FHWR Links (open in new tab)

- STU-3 (R3) Specification
- STU-2 Specification
- FHIR wiki
- FHIR Chat
- FHIR.org
- Clinicians Workshop

## ClinFHIR Videos (open in new tab)

- Scenario Builder
- Adding structured data
- Logical Modeller
- Logical Modeller and Scenario Builder
- RESTful query tool

*Note that some of these videos may describe earlier versions, so may not completely match the current functionality.*

## Other links

- SNOMED browser
### Terminology Links

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Main modules (open in new tab)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Patient Viewer</strong></td>
<td>Display resources for a specific patient, using a number of different views such as a list by resource type, JSON &amp; tree views, encounters by condition, numeric Observation charting and graphical relationship views. There is also the option to add a new patient, and to create sample data for that patient.</td>
</tr>
<tr>
<td><strong>Server Query</strong></td>
<td>Supports ad hoc queries against any FHIR server. Includes a simple query builder. The response can be displayed as JSON or a Tree view, and FHIRPath is supported. Can access any compliant FHIR server (must expose a Capability Statement).</td>
</tr>
<tr>
<td><strong>Scenario Builder</strong></td>
<td>The Scenario Builder is used to join together the resources needed to represent a specific clinical scenario. It can use Core Resource types, Profiles and Logical models as it does this. The intention is to help people understand how resources can tell a clinical story, and to validate that the resource types available (including profiles) are sufficient. Note that the builder still has issues with more complex resource types - this is a work in progress.</td>
</tr>
<tr>
<td><strong>Logical Modeller</strong></td>
<td>The Logical Modeller allows the creation of a model that represents a particular interoperability requirement in a format that is easy to use. It uses FHIR data types, and can be based on an existing resource type or completely 'ad hoc'. It is intended to act as a 'bridge' between Modeler and User, and can act as the basis for the generation of the profiling components required by FHIR. Models are saved on the Conformance Server. Can reference ValueSets from the Terminology Server.</td>
</tr>
<tr>
<td><strong>Implementation Guide Browser</strong></td>
<td>Display the contents of an Implementation Guide, and the relationships between the contents of the Guide. The Implementation guide, profiles and Extension Definitions are on the Conformance Server; the terminology resources (e.g., ValueSet) are on the Terminology Server.</td>
</tr>
<tr>
<td><strong>Extension Definition builder</strong></td>
<td>Views and builds extension definitions. These can be defined and applied to the Logical Model, which will allow them to be included in the generated Profile.</td>
</tr>
<tr>
<td><strong>CodeSystem builder</strong></td>
<td>The CodeSystem defines a set of Concepts from which a ValueSet provides possible values for a resource element. The actual 'binding' between CodeSystem and element is done by the ValueSet. This component allows you to build (and edit) a CodeSystem, and optionally builds the ValueSet as well. CodeSystems are saved on the Terminology Server.</td>
</tr>
<tr>
<td><strong>ValueSet explorer</strong></td>
<td>Lets you view existing ValueSets. The builder works best with SNOMED (at the moment). ValueSets are stored on the Terminology Server.</td>
</tr>
<tr>
<td><strong>Conformance Server</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Terminology Server</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Current servers</strong></td>
<td></td>
</tr>
<tr>
<td>Data Server</td>
<td>Public HAPI STU3 server</td>
</tr>
<tr>
<td>Conformance Server</td>
<td>Public HAPI STU3 server</td>
</tr>
<tr>
<td>Terminology Server</td>
<td>Public HAPI STU3 server</td>
</tr>
</tbody>
</table>

**FHIR Links (open in new tab)**
- STU-3 (R3) Specification
- STU-2 Specification
- FHIR wiki
- FHIR Chat
- FHIR.org
- Clinicians Workshop

**clinFHIR Videos (open in new tab)**
- Scenario Builder
- Logical Modeller
- Logical Modeller and Scenario Builder
- RESTful query tool

Note that some of these videos may describe earlier versions, so may not completely match the current functionality.

**Other links**
- SNOMED browser
Create User Account

This is an UNSECURED server! Use a dummy password!
### Server Selection

#### Main modules (open in new tab)

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Patient Viewer</strong></td>
<td>Display resources for a specific patient, using a number of different views such as a list by resource type, JSON &amp; tree views, encouters by condition, numeric Observation charting and graphical relationship views. There is also the option to add a new patient, and to create sample data for that patient.</td>
</tr>
<tr>
<td><strong>Server Query</strong></td>
<td>Supports ad hoc queries against any FHIR server. Includes a simple query builder. The response can be displayed as JSON or a tree view, and FHIRPath is supported.</td>
</tr>
<tr>
<td><strong>Scenario Builder</strong></td>
<td>The Scenario Builder is used to join together the resources needed to represent a specific clinical scenario. It can use Core Resource types, Profiles and Logical models as it does this. The intention is to help people understand how resources can tell a clinical story, and to validate that the resource types available (including profiles) are sufficient. Note that the builder still has issues with more complex resource types - this is a work in progress.</td>
</tr>
<tr>
<td><strong>Logical Modeller</strong></td>
<td>The Logical modeller allows the creation of a model that represents a particular interoperability requirement in a format that is easy to use. It uses FHIR datatypes, and can be based on an existing resource type or completely 'ad hoc'. It is intended to act as a 'bridge' between Modeller and User, and can act as the basis for the generation of the profiling components required by FHIR.</td>
</tr>
<tr>
<td><strong>Extension Definition builder</strong></td>
<td>Views and builds extension definitions. These can be defined and applied to the Logical Model, which will allow them to be included in the generated Profile.</td>
</tr>
<tr>
<td><strong>CodeSystem builder</strong></td>
<td>The CodeSystem defines a set of Concepts from which a ValueSet provides possible values for a resource element. The actual 'binding' between CodeSystem and element is done by the ValueSet. This component allows you to build (and edit) a CodeSystem, and optionally builds the ValueSet as well.</td>
</tr>
<tr>
<td><strong>ValueSet explorer</strong></td>
<td>Lets you view existing ValueSets. The builder works best with SNOMED (at the moment).</td>
</tr>
</tbody>
</table>

#### Experimental modules (open in new tab)

- Patient resources are stored on the Data Server. The server should support the Patient/$everything operation.
- Can access any compliant FHIR server (must expose a Capability Statement).
- The implementation guide, profiles and Extension Definitions are on the Conformance Server, the terminology resources (e.g., ValueSet) are on the Terminology Server.

#### Add Server
- Set all the same as the Data Server

#### Current servers
- **Data Server**
  - Public HAPI STU3 server
- **Conformance Server**
  - Public HAPI STU3 server
- **Terminology Server**
  - Public HAPI STU3 server

#### FHIRE Links (open in new tab)
- STU-3 (R3) Specification
- STU-2 Specification
- FHIR Chat
- FHIR Wiki
- Clinicians Workshop

#### clinFHIRE Videos (open in new tab)
- Scenario Builder
- Adding structured data
- Logical Modeller
- Logical Modeller and Scenario Builder
- RESTful query tool

*Note that some of these videos may describe earlier versions, so may not completely match the current functionality.*
# Server Selection

## clinFHIR Launcher

### Main modules (open in new tab)

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient Viewer</td>
<td>Display resources for a specific patient, using a number of different views such as a list by resource type, JSON &amp; Tree views, encounters by condition, numeric Observation charting and graphical relationship views. There is also the option to add a new patient, and to create sample data for that patient.</td>
</tr>
<tr>
<td>Server Query</td>
<td>Supports ad hoc queries against any FHIR server. Includes a simple query builder. The response can be displayed as JSON or a Tree view, and FHIRPath is supported.</td>
</tr>
<tr>
<td>Scenario Builder</td>
<td>The Scenario Builder is used to join together the resources needed to represent a specific clinical scenario. It can use Core Resource types, Profiles and Logical models as it does this. The intention is to help people understand how resources can tell a clinical story, and to validate that the resource types available (including profiles) are sufficient. Note that the builder still has issues with more complex resource types - this is a work in progress.</td>
</tr>
<tr>
<td>Logical Modeler</td>
<td>The Logical modeler allows the creation of a model that represents a particular interoperability requirement in a format that is easy to use. It uses FHIR datatypes, and can be based on an existing resource type or completely 'ad hoc'. It is intended to act as a 'bridge' between Modelled and User, and can act as the basis for the generation of the profiling components required by FHIR.</td>
</tr>
<tr>
<td>Extension Definition</td>
<td>Views and builds extension definitions. These can be defined and applied to the Logical Model, which will allow them to be included in the generated Profile.</td>
</tr>
<tr>
<td>CodeSystem builder</td>
<td>The CodeSystem defines a set of Concepts from which a ValueSet provides possible values for a resource element. The actual 'binding' between CodeSystem and element is done by the ValueSet. This component allows you to build (and edit) a CodeSystem, and optionally builds the ValueSet as well.</td>
</tr>
<tr>
<td>ValueSet explorer</td>
<td>Lets you view existing ValueSets. The builder works best with SNOMED (at the moment).</td>
</tr>
</tbody>
</table>

### Experimental modules (open in new tab)

- Patient resources are stored on the Data Server. The server should support the Patient/$everything operation.

### Data Server
- Public FHIR STU3 server
- Grahame's STU2 server
- Grahame's STU3 server

### Conformance Server
- HealthConx STU2 server
- HealthConx STU3 server

### Terminology Server
- Public FHIR STU3 server
- HealthConx STU2 server
- HealthConx STU3 server
- Local FHIR STU2 server
- Local FHIR STU3 server
- HL7 New Zealand STU2 server
- fhirl.org
- Ontoserver STU3
- MIHIN STU2
- Simplifier R3
- Aegis WildFHIR STU3
- clinFHIR R2
- clinFHIR R3
- GoFHIR
- HSPC-14
- HSIC Careplan
- cfProxy

### Other links
- SNOMED browser
Task

- Open clinfhir.com
- Set up an account – DO NOT REUSE AN OLD PW
- Set up your 3 servers – HAPI3
- Lower your laptop lid when you’re done
Please select a patient using the 'Select Patient' button at the upper right.

If you want to add a new patient, then click the 'Select Patient' button, and in the modal dialog that appears, there's a link to add a new patient.
Patient Viewer – Resource Explorer

1. Resource explorer
2. Condition resources
3. Type 2 diabetes mellitus
4. Outward references
5. Inward references
Patient Viewer – Resource Reference Graph

Scroll to zoom graph
Click and drag to move
Task – Select and View Patient

• Go to Patient Viewer Module
• Enter “Anyman” in Patient Search
• Select Patient
• Other interesting patients
  • Joseph Framingham male 1953-02-14
    • Id: cf-1497294721585
  • Aric734 Hoppe202 male 1928-02-28 on GoFHIR server
    • Id: 58b366413425def0f0f71357
• Explore other servers and try out names like “Smith”
• Create a sample patient and view the data
Please select a patient using the 'Select Patient' button at the upper right.

If you want to add a new patient, then click the 'Select Patient' button, and in the modal dialog that appears, there's a link to add a new patient.
Create Basic Set of Resources
Add new Patient

Progress...
Adding Lucas Thomas
Added patient with the id : 304311
Checking that the required reference resources exist
adding Conditions...
Added Conditions List
adding Encounters...
added encounters Added 10 Encounters
Added 25 Observations
Added Medications List
Added 2 Appointments

All resources have been created. Click the close button to return to the front page
You can review the resource instances that were created using the 'Details' link at the upper left on the screen.
Condition resources:

- GERD
- onychomycosis
- high cholesterol
- asthma
- angina

**hypertension**
- diabetes
- neuropathic pain
- depression
- rheumatoid arthritis - left elbow
- rheumatoid arthritis - both hands

Outward references:
Condition subject === Patient/304311
Lucas Thomas

Inward references:
List/304312, item
Lunch Break
Creating FHIR Resource INSTANCES
### Scenario Builder Module

**Main modules (open in new tab)**

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient Viewer</td>
<td>Display resources for a specific patient. A list by resource type, json &amp; tree views, encounters by condition, numeric observation charting and graphical relationship views. There is also the option to add a new patient and to create sample data for that patient.</td>
</tr>
<tr>
<td>Server Query</td>
<td>Supports ad-hoc queries against any FHIR server. Includes a simple query builder. The response can be displayed as Json or a Tree view, and FHIRPath is supported. The Scenario Builder is used to join together the resources needed to represent a specific clinical scenario. It uses Core Resource types, Profiles and Logical models as it does this. The intention is to help people understand how resources can tell a clinical story, and to validate that the resource types available (including profiles) are sufficient. Note that the builder still has issues with more complex resource types - this is a work in progress.</td>
</tr>
<tr>
<td>Logical Modeller</td>
<td>The Logical Modeller allows the creation of a model that represents a particular interoperability requirement in a format that is easy to use. It uses FHIR datatypes, and can be based on an existing resource type or completely 'ad hoc'. It is intended to act as a bridge between Modeller and User, and can act as the basis for the generation of the profiling components required by FHIR. Models are saved on the Conformance Server. Can reference ValueSets from the Terminology server.</td>
</tr>
<tr>
<td>Implementation Guide Browser</td>
<td>Display the contents of an Implementation Guide, and the relationships between the contents of the Guide. The implementation guide, profiles and extension definitions are on the Conformance Server, the terminology resources (eg ValueSet) are on the Terminology Server.</td>
</tr>
<tr>
<td>Extension Definition builder</td>
<td>Views and builds extension definitions. These can be defined and applied to the Logical Model, which will allow them to be included in the generated profile. Extension definitions are saved on the Conformance Server.</td>
</tr>
<tr>
<td>CodeSystem builder</td>
<td>The CodeSystem defines a set of Concepts from which a ValueSet provides possible values for a resource element. The actual 'binding' between CodeSystem and element is done by the ValueSet. This component allows you to build (and edit) a CodeSystem, and optionally builds the ValueSet as well. CodeSystems are saved on the Terminology Server.</td>
</tr>
<tr>
<td>ValueSet explorer</td>
<td>Lets you viewing existing ValueSets. The builder works best with SNOMED (at the moment). ValueSets are stored on the Terminology Server.</td>
</tr>
</tbody>
</table>

**Experimental modules (open in new tab)**

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient Viewer</td>
<td>Patient resources are stored on the Data Server. The server should support the Patient/$everything operation.</td>
</tr>
<tr>
<td>Logical Modeller</td>
<td>The Scenario Builder is used to join together the resources needed to represent a specific clinical scenario. It uses Core Resource types, Profiles and Logical models as it does this. The intention is to help people understand how resources can tell a clinical story, and to validate that the resource types available (including profiles) are sufficient. Note that the builder still has issues with more complex resource types - this is a work in progress.</td>
</tr>
<tr>
<td>Implementation Guide Browser</td>
<td>The implementation guide, profiles and extension definitions are on the Conformance Server, the terminology resources (eg ValueSet) are on the Terminology Server.</td>
</tr>
<tr>
<td>Extension Definition builder</td>
<td>Views and builds extension definitions. These can be defined and applied to the Logical Model, which will allow them to be included in the generated profile. Extension definitions are saved on the Conformance Server.</td>
</tr>
<tr>
<td>CodeSystem builder</td>
<td>The CodeSystem defines a set of Concepts from which a ValueSet provides possible values for a resource element. The actual 'binding' between CodeSystem and element is done by the ValueSet. This component allows you to build (and edit) a CodeSystem, and optionally builds the ValueSet as well. CodeSystems are saved on the Terminology Server.</td>
</tr>
<tr>
<td>ValueSet explorer</td>
<td>Lets you viewing existing ValueSets. The builder works best with SNOMED (at the moment). ValueSets are stored on the Terminology Server.</td>
</tr>
</tbody>
</table>

**Other links**

- SNOMED browser
- FHIR Links (open in new tab)
  - STU-3 (R3) Specification: Hay on FHIR
  - STU-2 Specification: FHIR Chat
  - FHIR wiki: FHIR.org
  - Clinicians Workshop

**clinFHIR Videos (open in new tab)**

- Scenario Builder
- Adding structured data
- Logical Modeller
- Logical Modeller and Scenario Builder
- RESTful query tool

*Note that some of these videos may describe earlier versions, so may not completely match the current functionality.*
Create New Scenario

- Click the 'New Scenario' link to the left to create a new Set.
- Select the Library of scenarios ('View Library' link to the upper right) and download one to edit or view.
Note: Scenarios are saved locally and need to be posted to a server in order to be shared. Stay tuned!
<table>
<thead>
<tr>
<th>Type</th>
<th>Text</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>List</td>
<td>Report</td>
<td></td>
</tr>
</tbody>
</table>

**Add new Resource**

- **Core resource**: Patient
- **Text**: Joseph AMIA001
Add HumanName property to Patient.name

Use: Usual

First Name: Joseph
Middle: 
Surname: AMIA
Suffix: 

Combined Name: Joseph AMIA
1. Select data element
2. Look at value set
3. Delete element
4. Validate resource
Validate Resource

1. Confirm valid resource
2. Validate all resources & Post resource to server
3. Resource id of resource on the server
Task – Create a Patient (15 min)

• Use Scenario Builder module to create your own patient
  • Include a name, gender and birthdate
  • Record patient name and id on your note to find it later
• Validate your resource instance
• Update (POST) the resource to the data server
• Confirm that your patient is on the data server using the Patient Viewer module and the patient id
Create FHIR Condition Instance
Enter search term
Explore Value Set

Display ValueSet: condition-code

Includes

http://snomed.info/sct

Filters

<table>
<thead>
<tr>
<th>Property</th>
<th>Operation</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>concept</td>
<td>is-a</td>
<td>404684003</td>
</tr>
</tbody>
</table>

http://snomed.info/sct

Concepts

<table>
<thead>
<tr>
<th>Code</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>160245001</td>
<td>No current problems or disability</td>
</tr>
</tbody>
</table>

http://fhirtest.uhn.ca/baseDstu3/

Limitations

Define concepts are not recursive
SNOMED Coded Condition Concept
Hypertension
Condition of:1509562317256

<table>
<thead>
<tr>
<th>Type</th>
<th>Text</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition</td>
<td>Hypertension</td>
<td>✔</td>
</tr>
<tr>
<td>Patient</td>
<td>Joseph AMIA001</td>
<td>✔</td>
</tr>
</tbody>
</table>

- **Condition.code**
  - CodeableConcept
  - ValueSet Binding (example)
    - http://hl7.org/fhir/ValueSet/condition-code

Identification of the condition, problem or diagnosis.
1. Select subject[x] data element
2. Appropriate data types
3. Available datatypes in this scenario
Note: ClinFHIR automatically links new resources to existing patient resource when appropriate
Errors in Validation

Note: Use the validation warnings/errors to correct your resource, then update the server.
View Resource Graph of Scenario
Task – Create a Condition (15 min)

• Use Scenario Builder module to create your own condition
  • SNOMED Code and link to your patient
  • E.g. Diabetes mellitus type 2 (SCTID 44054006)
• Validate your resource instance
• Update (POST) the resource to the data server
• Confirm that your patient is on the data server using the Patient Viewer module and the patient id
• Add additional conditions if you are done early
Add HgbA1c Observation
Use Value Set Expand to find code
HgbA1c
Observation cf-1509554489602

Describes what was observed. Sometimes this is called the observation "name".
Add Quantity Value to Observation

Add a value to the observation
Add Quantity property to Observation.valueQuantity

Value
6.8
System
%

The information determined as a result of making the observation, if the information has a simple value.
Validate and POST Observation

<table>
<thead>
<tr>
<th>Type</th>
<th>Text</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition</td>
<td>Hypertension</td>
<td>✓</td>
</tr>
<tr>
<td>Observation</td>
<td>HgbA1c</td>
<td>✓</td>
</tr>
<tr>
<td>Patient</td>
<td>Joseph AMIA001</td>
<td>✓</td>
</tr>
</tbody>
</table>

The information determined as a result of making the observation, if the information has a simple value.
Task – Create an Observation (15min)

• Use Scenario Builder module to create your own Observation
  • Try Fasting Blood Glucose, Height, Weight and BMI
• Use FHIR Spec, Google or LOINC Browser to find LOINC concept code
• Validate your resource instance
• Update (POST) the resource to the data server
• Confirm that your patient is on the data server using the Patient Viewer module and the patient id
• Additional task – add additional observations
How FHIR supports Care Plan Elements

Explore Bob Anyman
Care Plan Elements

- Care Plan (Complex Resource)
  - Care Team Resource
  - Conditions
    - Goals
      - Objectives
        - Activities (MedicationRequest, ProcedureRequest, ReferralRequest, etc.)