ClinFHIR Tutorial
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
# ClinFHIR.com Main Page

<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>Patient resources are stored on the Data Server. The server should support the Patient/$everything operation.</strong></td>
</tr>
<tr>
<td>Displays 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></td>
</tr>
<tr>
<td><strong>Server Query</strong></td>
<td><strong>Can access any compliant FHIR server (must expose a Capability Statement)</strong></td>
</tr>
<tr>
<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>
<td></td>
</tr>
<tr>
<td><strong>Scenario Builder</strong></td>
<td><strong>Patient information is on the Data Server. Profiles on the Conformance server. ValueSets on the Terminology 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></td>
</tr>
<tr>
<td>Models are saved on the Conformance Server. Can reference ValueSets from the Terminology server.</td>
<td></td>
</tr>
<tr>
<td><strong>Logical Modeller</strong></td>
<td><strong>Create a simple scenario</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></td>
</tr>
<tr>
<td><strong>Implementation Guide Browser</strong></td>
<td><strong>Create a Document</strong></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></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></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></td>
</tr>
</tbody>
</table>

**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 (R3) Specification
- STU-2 Specification
- 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
<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>Patient resources are stored on the Data Server. The server should support the Patient/$everything operation.</strong></td>
</tr>
<tr>
<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>
<td></td>
</tr>
<tr>
<td><strong>Scenario Builder</strong></td>
<td><strong>Patient information is on the Data server. Profiles on the Conformance server. Vetsets on the Terminology 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>Models are saved on the Conformance Server. Can reference ValueSets from the 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></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>

**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 (R3) Specification
- STU-2 Specification
- FHIR wiki
- Hay on FHIR
- 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.*
## Modules

### Main modules

<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, 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. Patient information is on the Data Server. Profiles on the Conformance server. ValueSets on the Terminology server. 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>
</tbody>
</table>

### Experimental modules

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>Server</th>
<th>Server Type</th>
</tr>
</thead>
<tbody>
<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**

- STU-3 (R3) Specification
- STU-2 Specification
- 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.*
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.
  - Patient resources are stored on the Data Server. The server should support the Patient/$everything operation.

- **Server Query**
  - Support 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.
  - Models are saved on the Conformance Server. Can reference ValueSets from the Termination server.

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

- **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.
  - Extension definitions are saved on the Conformance Server.

- **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.
  - CodeSystems are saved on the Terminology Server.

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

Experimental modules (open in new tab)

- **Conformance Server**
  - Public HAPI STU3 server
  - Can access any compliant FHIR server (must expose a Capability Statement).

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

Add Server

- Set all the same as the Data Server

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
# 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 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.

## Experimental modules (open in new tab)

- **Patient resources**
  - 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**

- **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.

## Current servers

- **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 (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.*

### Other links

- **SNOMED browser**
### 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><strong>Patient resources are stored on the Data Server. The server should support the Patient/Everything operation.</strong></td>
</tr>
<tr>
<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>
<td>Can access any compliant FHIR server (must expose a Capability Statement).</td>
</tr>
<tr>
<td><strong>Scenario Builder</strong></td>
<td><strong>Patient information is on the Data Server. Profiles on the Conformance server. ValueSets on the Terminology 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>Create a simple scenario Using structured data to a scenario Creating a Document</td>
</tr>
<tr>
<td><strong>Logical Modeller</strong></td>
<td><strong>Models are saved on the Conformance Server. Can reference ValueSets from the 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>Create an Information Model Create a Resources Model</td>
</tr>
<tr>
<td><strong>Implementation Guide Browser</strong></td>
<td><strong>The Implementation guide, profiles and Extension Definitions are on the Conformance Server. The terminology resources (eg ValueSet) are on the Terminology Server</strong></td>
</tr>
<tr>
<td>Display the contents of an Implementation Guide, and the relationships between the contents of the Guide.</td>
<td>Create an Implementation Guide</td>
</tr>
<tr>
<td><strong>Extension Definition builder</strong></td>
<td><strong>Extension definitions are saved on the Conformance Server</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></td>
</tr>
<tr>
<td><strong>CodeSystem builder</strong></td>
<td><strong>CodeSystems are saved on the Terminology Server.</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></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></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

**clnFHIR 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

### 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 Modeler
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 Modeler and User, and can act as the basis for the generation of the prototyping components required by FHIR.

### Experimental modules (open in new tab)
#### Patient Viewer
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 Modeler
Models are saved on the Conformance Server. Can reference ValueSets from the Terminology server.

### 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).

### Current servers
- **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 (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 Modeler
- Logical Modeler 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:
- SNOEMED browser
Create User Account
Create User Account

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

Main modules (open in new tab) | Experimental 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.


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).
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 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.

**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.

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

**Scenario Builder**

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

**Implementation Guide Browser**
The Implementation guide, profiles and Extension Definitions are on the Conformance Server; the terminology resources (e.g. 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.

**Data Server**
- Public HAPI STU3 server
- Grahames STU2 server
- Grahames STU3 server
- Public HAPI STU2 server

**Conformance Server**
- HealthConnex STU2 server
- HealthConnex STU3 server
- Local HAPI STU2 server
- Local HAPI STU3 server
- HL7 New Zealand STU2 server
- fhir.org
- Ontoserver STU3
- MHIIN STU2
- Simplifier R3
- Aegirs WildFHIR STU3
- clinFHIR R2
- clinFHIR R3
- GoFHIR
- HSPC-14
- HSPC Careplan
- cfProxy

**Terminology Server**
- Public HAPI STU3 server
- HealthConnex STU2 server
- HealthConnex STU3 server
- Local HAPI STU2 server
- Local HAPI STU3 server
- HL7 New Zealand STU2 server
- fhir.org
- Ontoserver STU3
- MHIIN STU2
- Simplifier R3
- Aegirs WildFHIR STU3
- clinFHIR R2
- clinFHIR R3
- GoFHIR
- HSPC-14
- HSPC Careplan
- cfProxy

**Other links**

**SNOMED browser**
Task

- Open clinfhir.com
- Set up an account – DO NOT REUSE AN OLD PW
- Set up your servers –
  - Data Server – HSPC CarePlan
  - Conformance – Public HAPI STU3
  - Terminology – Ontoserver (Terminology)
- Lower your laptop lid when you’re done
### Modules

#### 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, 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.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Extender 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). ValueSets are stored on the Terminology Server.</td>
</tr>
</tbody>
</table>

#### Current servers

- **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 (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.*
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.
Search for Patient

Entering name:
- Bob Anyman male 1954-09-17
- Robert Anyman male 1956-05-15
- Bob Anyman male 1954-01-01
- Larry Anyman male 1956-03-01

Selected patient: Bob Anyman male 1954-09-17

Outward references:
- Goal.subject -> Patient/CarePlan-Patient-1
- Bob Anyman

Inward references:
- Goal.addresses 0 -> Condition/CarePlan-Condition-1
- Type 2 diabetes mellitus
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 “Betsy” in Patient Search
- Select “Betsy Johnson” patient
- Explore patient’s FHIR resources
- Explore other servers and try out names like
  - David Sullivan
  - Meghan Anderson
- Create a sample patient and view the data
Patient Viewer

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.
hypertension
Ok – Hang on!
Here we go!!!!
Creating FHIR Resource INSTANCES

https://fhirblog.com/creating-a-simple-scenario/

Time Check – set working time....
Adding Structured Data

https://fhirblog.com/adding-structured-data-to-a-scenario/

Time Check – set working time....
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