ClinFHIR Tutorial

Laura Heermann Langford, PhD, RN
Russ Leftwich, MD
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>Patient resources are stored on the Data Server. The server should support the Patient/$everything operation.</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>Server Query</strong></td>
<td>Can access any compliant FHIR server (must expose a Capability Statement)</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 - a work in progress.</td>
<td></td>
</tr>
<tr>
<td><strong>Scenario Builder</strong></td>
<td>Patient information is on the Data Server. Profiles on the Conformance server. ValueSets on the Terminology server.</td>
</tr>
<tr>
<td>The Logical model 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 profiling components required by FHIR.</td>
<td></td>
</tr>
<tr>
<td><strong>Logical Modeller</strong></td>
<td>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>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>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>Extension Definition builder</strong></td>
<td>Extension definitions are saved on the Conformance Server</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>CodeSystems are saved on the Terminology Server.</td>
<td></td>
</tr>
<tr>
<td><strong>CodeSystem builder</strong></td>
<td>ValueSets are stored on the Terminology Server</td>
</tr>
<tr>
<td>Lets you view existing ValueSets. The builder works best with SNOMED (at the moment).</td>
<td></td>
</tr>
<tr>
<td><strong>ValueSet explorer</strong></td>
<td></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 (open in new tab)**

- STU-3 (R3) Specification
- STU-2 Specification
- FHIR wiki
- 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
## User Settings

<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>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></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><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>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><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>
<tr>
<td>Module</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<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.</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 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>Implementation Guide</strong></td>
<td>Display the contents of an Implementation Guide, and the relationships between the contents of the Guide.</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>
<tr>
<td><strong>Experimental modules (open in new tab)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>FHIR Links (open in new tab)</strong></td>
<td></td>
</tr>
<tr>
<td>STU-3 (R3) Specification</td>
<td>Hay on FHIR</td>
</tr>
<tr>
<td>STU-2 Specification</td>
<td>FHIR Chat</td>
</tr>
<tr>
<td>FHIR wiki</td>
<td>FHIR.org</td>
</tr>
<tr>
<td>Other links</td>
<td>Clinicians Workshop</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

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

clinFHIR Videos

- 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

**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>Displays resources for a specific patient, using a variety 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 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 <code>ad hoc</code> itself. It is intended to act as a <code>bridge</code> 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 <code>binding</code> 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)**

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Patient resources</strong></td>
<td>Patient resources are stored on the Data Server. The server should support the Patient/Everything operation.</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
- 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: Some of these videos may describe earlier versions, so may not completely match the current functionality.*

**Other links**

- SNOMED browser
## Useful FHIR Links

<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>Patient resources are stored on the Data Server. The server should support the Patient/$everything operation.</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></td>
</tr>
<tr>
<td><strong>Server Query</strong></td>
<td>Can access any compliant FHIR server (must expose a Capability Statement)</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>Patient information is on the Data Server. Profiles on the Conformance server. ValueSets on the Terminology server.</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><strong>Logical Modeller</strong></td>
<td>Models are saved on the Conformance Server. Can reference ValueSets from the Terminology server.</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 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.</td>
<td></td>
</tr>
<tr>
<td><strong>Implementation Guide Browser</strong></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><strong>Extension Definition builder</strong></td>
<td>Extension definitions are saved on the Conformance Server</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>CodeSystems are saved on the Terminology Server.</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>ValueSets are stored on the Terminology Server</td>
</tr>
<tr>
<td>Lets you view existing ValueSets. The builder works best with SNOMED (at the moment).</td>
<td></td>
</tr>
</tbody>
</table>
Terminology Links

<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>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>Can access any compliant FHIR server (must expose a Capability Statement)</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 (e.g 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>Can access any compliant FHIR server (must expose a Capability Statement)</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>Can access any compliant FHIR server (must expose a Capability Statement)</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>Can access any compliant FHIR server (must expose a Capability Statement)</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>Can access any compliant FHIR server (must expose a Capability Statement)</td>
</tr>
</tbody>
</table>

Other links
- SNOMED browser
### 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. Can access any compliant FHIR server (must expose a Capability Statement)</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. 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>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</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 stored on the Terminology Server.</td>
</tr>
<tr>
<td>ValueSet explorer</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>

### Experimental modules (open in new tab)

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
</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

### Other links

- **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.*
Create User Account

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

<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. Patient resources are stored on the Data Server. The server should support the Patient/Everything operation.</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. Patient information is on the Data Server. Profiles on the Conformance server. ValueSets on the Terminology server. Create a simple scenario. Adding structured data to a scenario. Create a Document.</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. Create an Information Model. Create a Resources Model.</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. Create an Information Model. Create a Resources Model.</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>
### Server Selection

<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 Modeler</strong></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 Modeler 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>

### 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
- MIHIN STU2
- Simplifier R3
- Aegirs WildFHIR STU3
- clinFHIR R2
- clinFHIR R3
- GoFHIR
- HSPC-14
- HSPC Careplan
- cfProxy

### Terminology Server
- Should all be the same FHIR
- Create a simple scenario
- Adding structured data to a scenario
- Create a Document
- Create an Information Model
- Create a Resources Model

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

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

- **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
- 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
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.
Search for Patient

Search

Enter ID of patient on this server

Enter identifier of patient on this server

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

Add new patient

Outward references

Goal subject --> Patient/CarePlan-Patient-1
Bob Anyman

Goal. addresses 0 --> Condition/CarePlan-Condition-1
Type 2 diabetes mellitus

Inward references
Patient Viewer – Resource Explorer
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
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.
Patient Viewer

Search for Patient

- Robert Anyman male 1956-05-15
- Bob Anyman male 1954-01-01
- Larry Anyman male 1956-03-01

Add new patient

Outward references
- Goal subject => Patient/CarePlan-Patient-1 Bob Anyman
- Goal.addresses 0 => Condition/CarePlan-Condition-1 Type 2 diabetes mellitus

Inward references
- text high
- text Improve and maintenance of optimal foot health: aim at early detection of peripheral vascular problems and neuropathy
- text Improve and maintenance of optimal foot health: aim at early detection of peripheral vascular problems and neuropathy
- text Type 2 diabetes mellitus
Create Basic Set of Resources

Add new Patient

Identifier
First Name: Lucas
Last Name: Thomas
Date of Birth: 1984-05-13  Age: 33 years
Gender: Male

Generate samples
Add patient

Find existing patient
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
Pause
Creating FHIR Resource INSTANCES
Scenario Builder Module
Create New Scenario

To edit a scenario, you can either:
- 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!
Add new Resource

Core resource

Profile
Logical Model

Resource Type:

- NamingSystem
- NutritionOrder
- Observation
- OperationDefinition
- OperationOutcome
- Organization
- Parameters
- Patient
- PaymentNotice
- PaymentReconciliation
- Person
- PlanDefinition
- Practitioner
- PractitionerRole
- Procedure
- ProcedureRequest
- ProcessRequest
- ProcessResponse
- Provenance
- Questionnaire

Text

Type

Valid

List

Report

Type

Text

Valid

List

Report
<table>
<thead>
<tr>
<th>Local Scenarios</th>
<th>New Scenario</th>
<th>List</th>
<th>Description</th>
<th>Graph</th>
<th>FHIRPath</th>
<th>Mark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add new Resource</td>
<td>Core resource</td>
<td>Profile</td>
<td>Logical Model</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource Type:</td>
<td>Patient</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text</td>
<td>Joseph AMI001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Patient

name *

- telecom *
- gender
- birthDate
- deceased[x]
- address *
- maritalStatus
- multipleBirth[x]
- photo *
- contact *
- relationship *
- name
- telecom *
- address
- gender
- organization
- period

animal

- species
- breed
- genderStatus

communication *

- language
- preferred
- generalPractitioner[x]
- managingOrganization
- link *
Add HumanName property to Patient.name

Use: Usual

Pb: Joseph
Middle: AMIA

Joseph AMIA
Joseph AMIA001
Patient id-1509554949496

Structure & Reference

- Patient
  - identifier *
    - active
  - name *
    - telecom *
    - gender
    - birthDate
    - deceased[x]
    - address *
    - maritalStatus
    - multipleBirth[x]
    - photo *
  - contact *
    - relationship *
    - name
    - telecom *
    - address
    - gender
    - of organization
    - period
  - animal
    - species
    - breed
    - genderStatus
  - communication *
    - language
    - preferred
    - of generalPractitioner "[x]
    - of managingOrganization

A name associated with the individual.

```json
{
  "resourceType": "Patient",
  "text": {
    "status": "generated",
    "div": "<div xmlns="http://www.w3.org/1999/xhtml">Joseph AMIA001"}
}
```

```
{ "id": "c-1509554949496",
  "name": [
    { "use": "usual",
      "given": [ "Joseph"
    ],
    "family": "AMIA",
    "text": "Joseph AMIA"
  }
}
```
1. Select data element
2. Look at value set
3. Delete element
4. 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