# 2022-05 Patient Track

- **Short Description**: This track provides new participants with a friendly introduction to FHIR, using a simple scenario that can be met with limited domain knowledge and by those who have not had a lot of exposure to FHIR.

- **Long Description**: This is the Patient Track testing that is included in all FHIR Connectathons. This track gives participants the opportunity to learn about how FHIR Resources are constructed, how you can create a FHIR Resource on a FHIR Server, how you can update that FHIR Resource, how you can review the version history for that FHIR Resource, search for that FHIR Resource and delete the FHIR Resource. This tracks is used by both FHIR Clients and FHIR Servers to learn about the basic foundational concepts used by FHIR. No prior experience is necessary, and it can be accomplished with or without the use of development or programming tools.

- **Type**: Educate on the use of FHIR technology

<table>
<thead>
<tr>
<th><strong>Submitting Work Group/Project/Accelerator/Affiliate/Implementer Group</strong></th>
<th>FHIR Management Group (FMG)</th>
</tr>
</thead>
</table>

| **Track Lead(s)** | Ron Shapiro |
| **Test Support** | Richard Ettema |

| **Track Lead Email(s)** | ron@qvera.com; richard.ettema@aegis.net |

| **Related Tracks** | None |

| **FHIR Version** | Current build (primary), FHIR R4, STU3, DSTU2 |

| **Specification(s) this track uses** | http://build.fhir.org/ |

| **Artifacts of focus** | Patient - http://build.fhir.org/patient.html |

| **Expected participants** | **Level 1:**  
- Any new participants that are looking for a connectathon track to participate on.  
- Public testing platforms may be used to assist participants; e.g. AEGIS - Touchstone tool and test scripts (optional)  

**Level 2:**  
- Those Connectathon attendees interested in using a more formalized testing approach.  
- Public testing platforms are used and will host the required TestScripts; e.g. AEGIS - Touchstone tool and test scripts |

| **Zulip stream** | https://chat.fhir.org/#narrow/stream/179207-connectathon-mgmt/topic/Patient.20Track |
### Track Kick Off Call

**Track Details**

**System Roles**

**FHIR Client**

This actor initiates the processing requests that enable the creation, deletion, manipulation and retrieval of Patient resource instances. The required, supported interactions are the defined basic CRUD operations: create, read, update and delete. Additional required, supported interactions are the operations: vread, history and search.

**FHIR Server**

This actor receives, processes and responds to the requests for creation, deletion, manipulation and retrieval of Patient resource instances. The implementation of this actor would normally provide for a repository storage mechanism along with corresponding maintenance and retrieval capabilities of the Patient resource instances. The required, supported interactions are the defined basic CRUD operations: create, read, update and delete. Additional required, supported interactions are the operations: vread, history and search.

**Pre-Requisites**

For all levels of testing the required pre-requisite is the fundamental requirement that all FHIR servers SHALL support the capabilities interaction.

**Level 1 - Introduction - New Participants/Systems**

This has been and will remain the primary purpose of this track and provides a 'friendly introduction' for those new to FHIR. Attendees participate in this track using a simple scenario that can be met with limited domain knowledge and by those who have not had a lot of exposure to FHIR. It is quite feasible to complete the client side of the track within a day with only knowledge of a development environment and little to no previous FHIR knowledge. If creating a server, advanced preparation will be required, but this scenario should somewhat limit the effort involved.

Pre-connectathon testing is encouraged, but not required, where the participants can utilize the existing Public Test Servers.

Testing and test reporting at the Connectathon event will be self-attested using the Results tab of the Connectathon Management Tool and primarily involve peer-to-peer execution between known FHIR clients and/or servers.

**Level 2 - Formal Testing - Participants with FHIR experience**

(Level 1 +) This level introduces a more formalized testing approach for those participants that have been working the FHIR specification and wish to move beyond basic testing and may have systems that are in active development, deployed or soon to be deployed into a production environment. Automated testing is significantly leveraged for both automated testing (testing tool to FHIR server) and surveillance of peer-to-peer testing (external FHIR client to external FHIR server).

Pre-connectathon testing is highly encouraged in order to be better prepared for the actual Connectathon event and become familiar with the public testing platforms that will be used for the formal testing.

Testing and test reporting will be done using the public testing platforms which will provide test results via the new FHIR TestReport resource type as well as any specific reporting capabilities of those testing platforms. These reports will provide qualitative and quantitative analysis of the system under test and its conformance to the FHIR specification.

**Scenarios**

**Level 1 - Introduction - New Participants/Systems**

The following scenarios represent the basic scenarios that participants will work to implement during the Connectathon event. Execution of these scenarios is expected to be performed with the other participants of this track as well as the Public Test Servers.

1. **Register a new patient**

   **Action:** FHIR client creates a new patient and save to FHIR server. The client can assign the Id.
   **Precondition:** Patient does not exist in FHIR server prior to action
   **Success Criteria:** Patient created correctly on FHIR server (use browser to inspect Patient)
   **Bonus point:** The Patient resource has an extension
   **Note:** the resource Id can either be created by the client or the server (depending on the capability of the server). However, if the server assigns the Id, then the client will need to be able to retrieve the Id from the server response or by a patient query.

2. **Update a patient**

   **Action:** FHIR client updates the patient created in scenario #1 and updates to FHIR server. The patient is retrieved by Id.
   **Precondition:** Patient has been created
   **Success Criteria:** Patient updated on FHIR server (use browser to inspect Patient)
   **Bonus Point #1:** Update a patient that has extensions, but leaving the extension untouched.
   **Bonus Point #2:** Update a patient that has extensions, and update the extension also.

3. **Retrieve Patient history**
Action: FHIR client searches the FHIR server for the history of a Patient  
Precondition: There is a patient that has at least one update  
Success Criteria: Patient's history displayed in interface. (use browser to query FHIR server)  
Bonus point: The UI allows the user to display previous versions of the Patient

4. Search for a patient on name
Action: FHIR client searches the FHIR server for patients with a given name
Precondition: Patients with that name have been created
Success Criteria: Patients displayed in interface. (use browser query to confirm)

5. Delete a patient
Action: FHIR client deletes the patient with a given id
Precondition: a Patient with that Id has been created
Success Criteria: Subsequently querying for the patient - either searching by name or by Id - fails.

Formal Testing

Level 2 - Formal Testing - Participants with FHIR experience

The following scenarios represent the formal testing scenarios that participants have been working to implement both prior to and during the Connectathon event. Execution of these scenarios will focus on automated testing with the public testing platforms and is expected to be performed with the other participants of this track as well as the Public Test Servers. Each of the scenarios are implemented as FHIR TestScript resources that include extensive assertions to provide a more comprehensive validation/verification of the systems under test conformance to the FHIR specification.

NOTE 1: All testing scenarios are performed by choosing FHIR TestScript resources that use:
   (a) XML or JSON
   (b) client or server assigned resource IDs

NOTE 2: When testing a FHIR server, all of the test scenarios can be completed with a single TestScript--see 99. test scenario below.

NOTE 3: When testing a FHIR client, be sure to remember the following:
   (a) Use the proxy URL of the FHIR server you are sending the request to, not the proxy URL of the FHIR client. The proxy URL assigned to the FHIR client is only used if the FHIR client is also a FHIR server and can accept requests.
   (b) The warning about a missing conformance statement for the FHIR client can be ignored. If the FHIR client does publish a conformance statement, it is used by the test tool, but it is not required.
   (c) When the test tool is "Waiting for Request", click on "Waiting for Request" and check the details of what it is waiting for under the "Submit the following request:" section--specifically, the Method, URL and Header values which should all match 100% what is sent.

1. Patient Registration/Creation
   FHIR Server
   Action: Use testing tool to create a new patient on the FHIR server
   Precondition: TestScript will first delete the patient if it exists
   Success Criteria: Testing tool passes all assertions and validations which include (a) HTTP status returned is 201 (Created), (b) returned format matches sent format, (c) patient can be retrieved and HTTP status 200 (OK) is returned, (d) retrieved patient format matches sent format and (e) conforms to base FHIR Patient profile.

   FHIR Client
   Action: FHIR client creates a new patient and saves it to FHIR server. A testing tool is used as a proxy in order to validate that the transaction is processed correctly.
   Precondition: Patient does not exist in FHIR server prior to action
   Success Criteria: Testing tool passes all assertions and validations which include (a) HTTP method is PUT, (b) URL contains full URL to Patient resource, (c) HTTP Header Accept contains the correct value, (d) HTTP Header Content-Type contains the correct value, (e) requested resource type is Patient, and (f) the HTTP Response from the FHIR server is valid.

2. Patient Modification/Update
   FHIR Server
   Action: Use testing tool to update an existing patient on the FHIR server with a new birth date
   Precondition: TestScript will first delete the patient if it exists and create a new patient resource to update
   Success Criteria: Testing tool passes all assertions and validations which include (a) HTTP status returned is 200 (OK), (b) HTTP Header Content-Type is returned with correct value, (c) updated patient can be retrieved and HTTP status 200 (OK) is returned, (d) retrieved patient conforms to base FHIR Patient profile.

   FHIR Client
Action: FHIR client updates a patient on the FHIR server. A testing tool is used as a proxy in order to validate that the transaction is processed correctly.
Precondition: Patient exists on FHIR server prior to action
Success Criteria: Testing tool passes all assertions and validations which include (a) HTTP method is PUT, (b) URL contains full URL to Patient resource, (c) HTTP Header Accept contains the correct value, (d) HTTP Header Content-Type contains the correct value, (e) requested resource type is Patient, and (f) the HTTP Response from the FHIR server is valid.

3. Patient Read

FHIR Server
Action: Use testing tool to retrieve a patient from the FHIR server
Precondition: TestScript will first delete the patient if it exists and create a new patient resource to retrieve
Success Criteria: Testing tool passes all assertions and validations which include (a) HTTP status returned is 200 (OK), (b) HTTP Header Content-Type is returned with correct value, (c) retrieved patient conforms to base FHIR Patient profile

FHIR Client
Action: FHIR client retrieves a patient from the FHIR server. A testing tool is used as a proxy in order to validate that the transaction is processed correctly.
Precondition: Patient exists on FHIR server prior to action
Success Criteria: Testing tool passes all assertions and validations which include (a) HTTP method is GET, (b) URL contains full URL to Patient resource, (c) HTTP Header Accept contains the correct value, (d) HTTP Header Content-Type is absent and (e) the HTTP Response from the FHIR server is valid.

4. Patient History

FHIR Server
Action: Use testing tool to retrieve patient history from the FHIR server
Precondition: TestScript will first delete the patient if it exists and create a new patient resource and update it with a 2nd revision to be retrieved
Success Criteria: Testing tool passes all assertions and validations which include (a) HTTP status returned is 200 (OK), (b) HTTP Header Content-Type is returned with correct value, (c) returned resource type is Bundle, (d) the returned Bundle conforms to the base FHIR Bundle profile, (e) the Bundle type is history, (f) the Bundle contains at least 2 entries and (g) the Bundle total value matches the number of entries.

FHIR Client
Action: FHIR client retrieves the patient history from the FHIR server. A testing tool is used as a proxy in order to validate that the transaction is processed correctly.
Precondition: Patient exists on FHIR server prior to action
Success Criteria: Testing tool passes all assertions and validations which include (a) HTTP method is GET, (b) URL contains full URL to Patient resource history, (c) HTTP Header Accept contains the correct value, (d) HTTP Header Content-Type is absent and (e) the HTTP Response from the FHIR server is valid.

5. Patient Version Read

FHIR Server
Action: Use testing tool to retrieve specific patient versions from the FHIR server
Precondition: TestScript will first delete the patient if it exists and create a new patient resource and update it with a 2nd revision to be retrieved
Success Criteria: Testing tool passes all assertions and validations for both versions of the patient which include (a) HTTP status returned is 200 (OK), (b) HTTP Header Content-Type is returned with correct value, (c) returned resource conforms to base FHIR Patient profile.

FHIR Client
Action: FHIR client retrieves a specific patient version from the FHIR server. A testing tool is used as a proxy in order to validate that the transaction is processed correctly.
Precondition: Patient exists on FHIR server prior to action
Success Criteria: Testing tool passes all assertions and validations which include (a) HTTP method is GET, (b) URL contains full URL to specific Patient version, (c) HTTP Header Accept contains the correct value, (d) HTTP Header Content-Type is absent and (e) the HTTP Response from the FHIR server is valid.

6. Patient Searching via Multiple Criteria

FHIR Server
Action: Use testing tool to search for patient on the FHIR server
Precondition: TestScript will first delete the patient if it exists and create a new patient resource to search for by identifier, given and family name
Success Criteria: Testing tool passes all assertions and validations which include (a) HTTP status returned is 200 (OK), (b) HTTP Header Content-Type is returned with correct value, (c) returned resource type is Bundle, (d) the returned Bundle conforms to the base FHIR Bundle profile, (e) the Bundle type is searchset, and (f) the Bundle total value matches the number of entries.
FHIR Client

Action: FHIR client searches for the patient on the FHIR server. A testing tool is used as a proxy in order to validate that the transaction is processed correctly.

Precondition: Patient exists on FHIR server prior to action

Success Criteria: Testing tool passes all assertions and validations which include (a) HTTP method is GET, (b) URL contains search parameters for identifier, family and given, (c) HTTP Header Accept contains the correct value, (d) HTTP Header Content-Type is absent and (e) the HTTP Response from the FHIR server is valid.

7. Patient Deletion/Removal

FHIR Server

Action: Use testing tool to delete patient on the FHIR server

Precondition: TestScript will first delete the patient if it exists and create a new patient resource to be deleted

Success Criteria: Testing tool passes all assertions and validations which include (a) HTTP status returned is 204 (No Content), (b) after attempting to read patient an HTTP status 410 (Gone) is returned, and (c) after attempting to search patient an HTTP status 200 (OK) is returned, the HTTP Content-Type is returned with the correct value, returned resource type is Bundle, and the returned Bundle contains no entries.

FHIR Client

Action: FHIR client deletes a patient from the FHIR server. A testing tool is used as a proxy in order to validate that the transaction is processed correctly.

Precondition: Patient exists on FHIR server prior to action

Success Criteria: Testing tool passes all assertions and validations which include (a) HTTP method is DELETE, (b) URL contains full URL to Patient resource, (c) HTTP Header Accept contains the correct value, (d) HTTP Header Content-Type is absent and (e) the HTTP Response from the FHIR server is valid.

98. All Non-Versioning Patient Operations Defined Above

FHIR Server (use this TestScript to perform all non-versioning test scenarios listed above with a single TestScript execution)

Action: Use testing tool to perform all of the test scenarios listed above on the FHIR server which include (1) create a patient, (2) update the patients birth date, (3) retrieve the current patient, (4) search for the patient by identifier, given and family name and (5) delete the patient.

Precondition: TestScript will first delete the patient if it exists

Success Criteria: Testing tool passes all assertions and validations which include those for each of the test scenarios listed above

99. All Patient Operations Defined Above

FHIR Server (use this TestScript to perform all test scenarios listed above with a single TestScript execution)

Action: Use testing tool to perform all of the test scenarios listed above on the FHIR server which include (1) create a patient, (2) update the patients birth date, (3) retrieve the current patient, (4) retrieve the patient history, (5) retrieve each version of the patient, (6) search for the patient by identifier, given and family name and (7) delete the patient.

Precondition: TestScript will first delete the patient if it exists

Success Criteria: Testing tool passes all assertions and validations which include those for each of the test scenarios listed above

FHIR Client

NOTE: FHIR clients must execute the 7 test scenarios one at a time so, there is no TestScript artifact for this test scenario

<table>
<thead>
<tr>
<th>Helpful Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>Here are some links to assist implementers:</td>
</tr>
<tr>
<td>• REST API in the Specification.</td>
</tr>
<tr>
<td>• Patient resource in the Specification.</td>
</tr>
<tr>
<td>• Open_Source_FHIR_implementations.</td>
</tr>
<tr>
<td>• Public Test Servers.</td>
</tr>
<tr>
<td>• Step by step tutorial and sample projects.</td>
</tr>
<tr>
<td>TextScript(s)</td>
</tr>
<tr>
<td>--------------</td>
</tr>
</tbody>
</table>
| **FHIR R4 (v4.0.0) TestScripts** | - https://github.com/FHIR/documents/tree/master/connectathons/VirualMay2020/FHIR4-0-1-Connectathon24/Patient-01-Intro  
- https://github.com/FHIR/documents/tree/master/connectathons/VirualMay2020/FHIR4-0-1-Connectathon24/Patient-02-Formal |
| **FHIR STU3 (v3.0.1) TestScripts** | - https://github.com/FHIR/documents/tree/master/connectathons/AtlantaSep2019/FHIR3-0-1-Connectathon22/Patient-01-Intro  
- https://github.com/FHIR/documents/tree/master/connectathons/AtlantaSep2019/FHIR3-0-1-Connectathon22/Patient-02-Formal |

The AEGIS Touchstone testing tool has test scripts available for tracks to test their implementations. See [www.touchstone.com](http://www.touchstone.com) to sign in our register if you are a new user. Below, you will find a link to the tests specific to this HL7 track. Please send questions or issues to touchstone_support@aegis.net and a team member will be glad to assist you.

Patient Tests can be found here:

- 4-0-1 Patient Basic Tests
- 4-0-1 Patient Advanced Tests
- r4 Patient Basic Tests
- r4 Patient Advanced Tests
- 3-5-0 Patient Basic Tests
- 3-5-0 Patient Advanced Tests
- 3-3-0 Patient Basic Tests
- 3-3-0 Patient Advanced Tests
- 3-2-0 Patient Basic Tests
- 3-2-0 Patient Advanced Tests
- 3-0-1 Patient Basic Tests
- 3-0-1 Patient Advanced Tests
- 1-8-0 Patient Basic Tests
- 1-8-0 Patient Advanced Tests
- 1-6-0 Patient Basic Tests
- 1-6-0 Patient Advanced Tests
- 1-4-0 Patient Basic Tests
- 1-2-0 Patient Basic Tests
- 1-0-2 Patient Basic Tests

Connectathon 25 Patient Intro and Formal Use Case Tests can be found [HERE](http://www.touchstone.com).

Please feel free to reach out to Touchstone_Support@aegis.net for any questions on testing or using the Touchstone FHIR Testing tool.