Client Fhir Testing Manual:  
CS6440

Team 37

# Setup

The client fhir testing tool is in the form of a proxy application that will record transactions between a fhir client and a fhir server. These HTTP transactions are recorded in a database where validation tests can later be run against them. The requests can also be replayed to mimic client or server endpoints.

For development purposes we will use the Inferno tool to act as a FHIR client and a public endpoint will be used as a FHIR server.

## Install & Run Inferno (Client)

1. Download & install inferno using Docker directions:
   1. <https://github.com/onc-healthit/inferno#installation-and-deployment>
2. Make sure docker desktop app is running
3. Run

docker-compose up

1. Open <http://localhost:4567/>

## Run Proxy

1. Download this github repo

git clone <https://github.com/inferno-community/client-fhir-testing.git>

cd client-fhir-testing

1. Run proxy

The following will read options from filename. If filename does not exist, one with default options will be created for you. If filename is left unspecified, proxy.yml will be used by default.

It is important to set the backend config option as this is the destination the proxy forwards to.

ruby start-proxy.rb [filename]

Alternatively, you can start the proxy via the rackup process and specify the backend as an environment variable.

FHIR\_PROXY\_BACKEND="https://r4.smarthealthit.org" rackup config.ru -p 9292 -o 0.0.0.0

## Run Inferno Tests

We use inferno as our client but you can use any client/server interactions in this step. Note that the docker URL listed below resolves to the docker host machine on which the proxy is running. Using localhost would refer to the docker instance and not the host itself.

* 1. On the Inferno homepage, under "Start Testing", select "US Core v3.1.0", and put in the address of the proxy service http://host.docker.internal:9292
  2. Run tests, check the database for logged HTTP transactions.

## Run Validator in Command Line

The validator is developed based on the [US Core Client CapabilityStatement](https://www.hl7.org/fhir/us/core/CapabilityStatement-us-core-client.htm). The [client CapabilityStatement JSON file](https://github.com/inferno-community/client-fhir-testing/blob/main/resources/CapabilityStatement-us-core-client.json) was parsed into three tables, [interaction](https://github.com/inferno-community/client-fhir-testing/blob/main/resources/CapabilityStatement_interaction.csv), [searchParam](https://github.com/inferno-community/client-fhir-testing/blob/main/resources/CapabilitySatement_searchParam.csv), and [search\_criteria](https://github.com/inferno-community/client-fhir-testing/blob/main/resources/CapabilitySatement_search_criteria.csv). Capabilities rules from the three tables were used to validate the client requests.

We created a [collection of Postman requests](https://github.com/inferno-community/client-fhir-testing/blob/main/test/fhir-client-test.postman_collection.json) to simulate a client test. The tool [newman](https://www.npmjs.com/package/newman) can be used to send the collection of requests to the proxy server.

1. To start the proxy server locally with the port 9292.

ruby start-proxy.rb

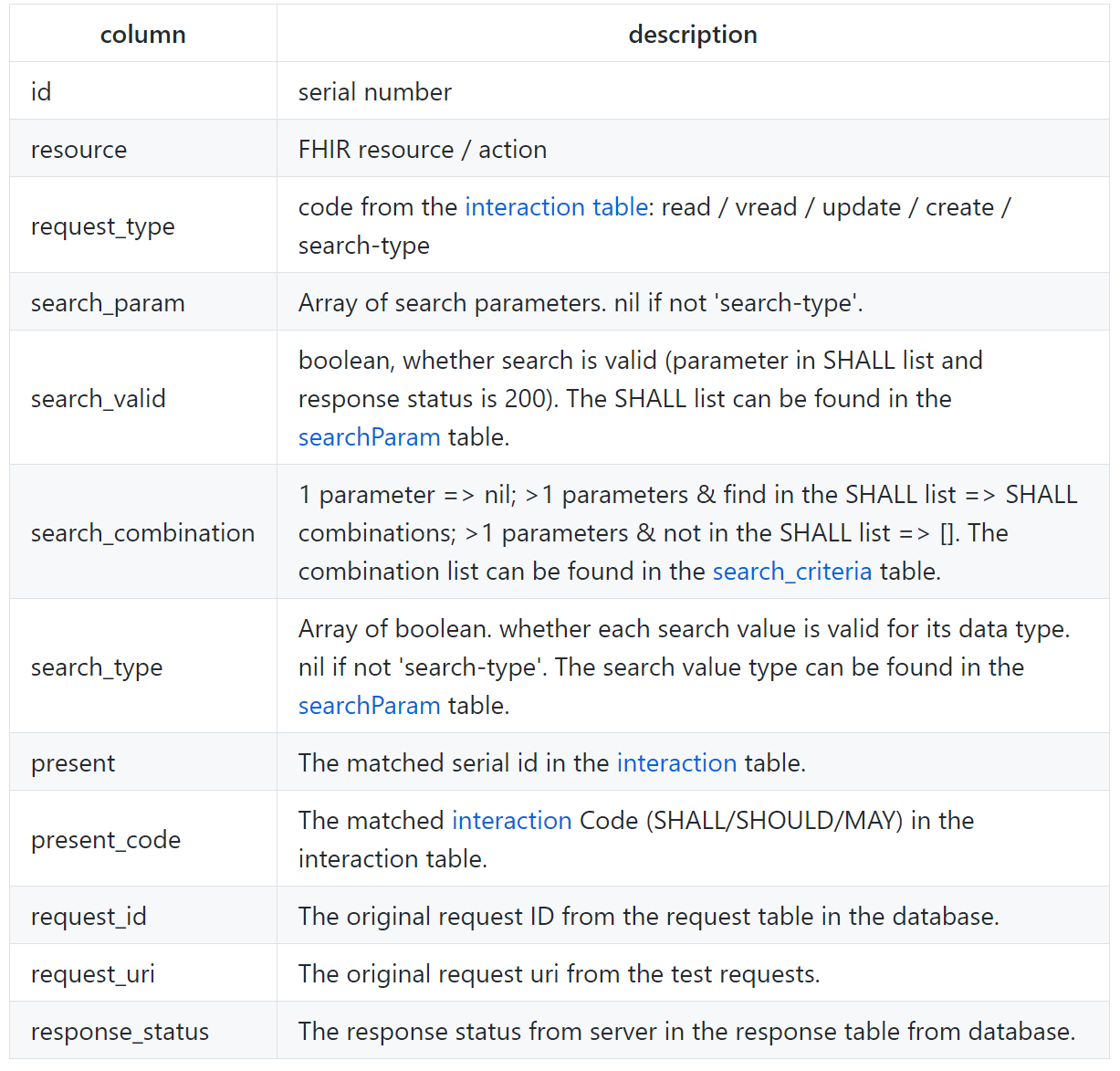
1. To send the requests with [newman](https://www.npmjs.com/package/newman) under the test directory.

cd test

newman run fhir-client-test.postman\_collection.json

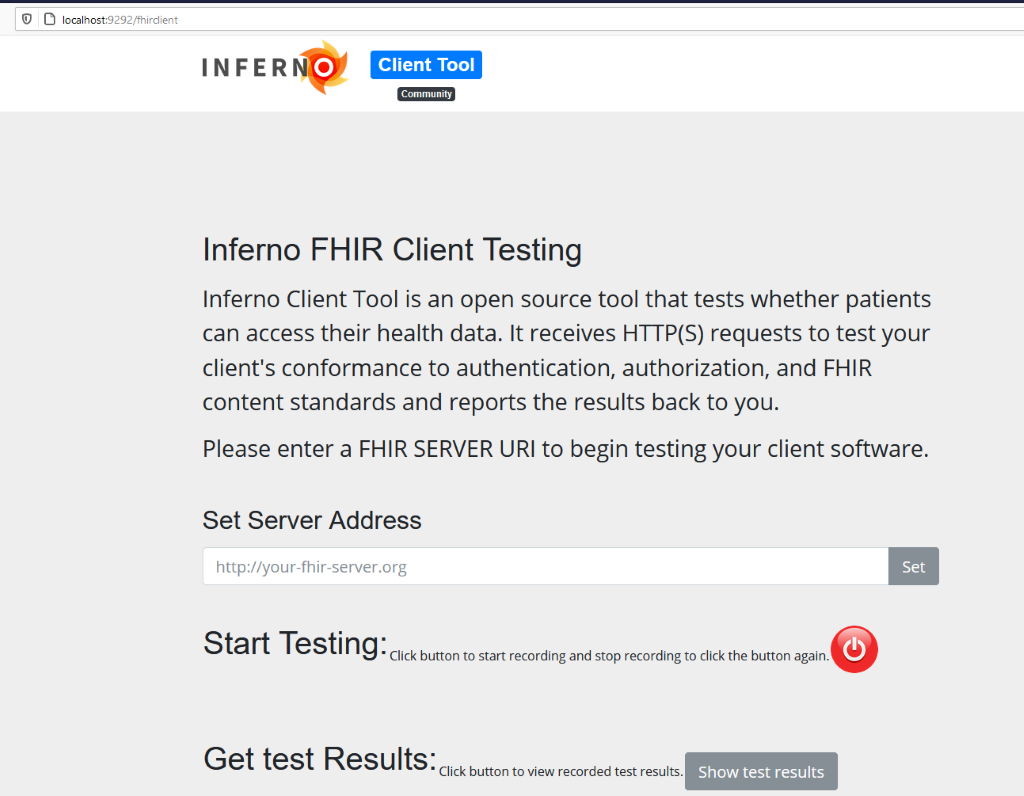
1. To run validator for the collection of requests.

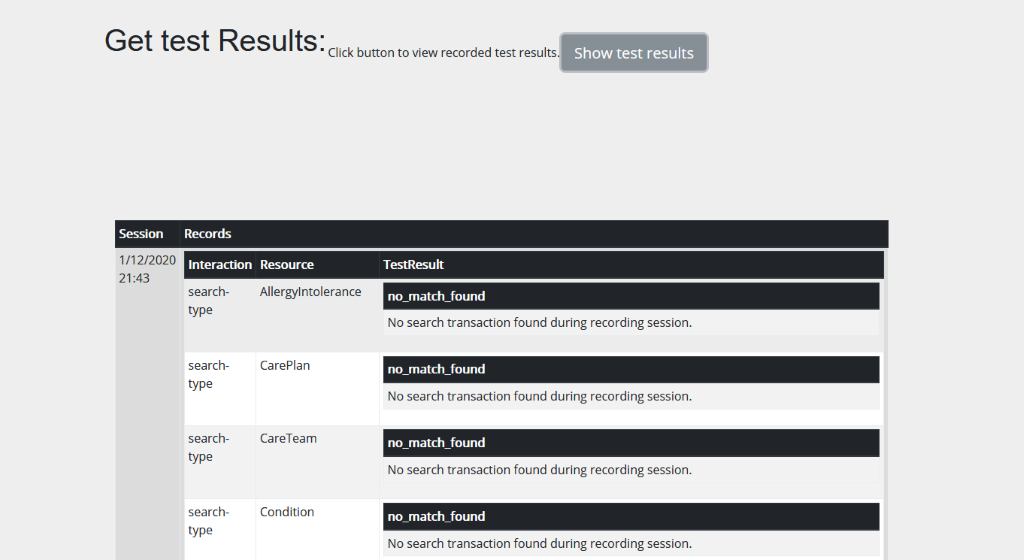
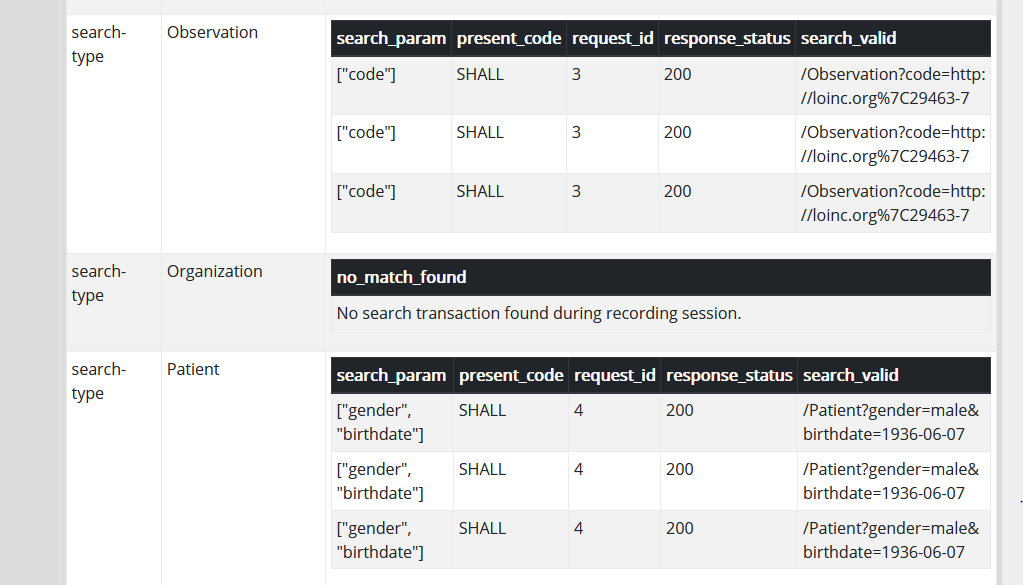
ruby ../test-validator.rb

A checklist.csv report will be generated and a check\_list table created in the database. Here is the description of the report.

## Client Tool UI

1. Once you have the tool running, navigate to <http://localhost/fhirclient> in your web browser
2. What should be displayed is below:



1. Backend Server Setting
   * The default backend server is set to <https://r4.smarthealthit.org>
   * If you want to change backend server, type URL, for example <http://hapi.fhir.org/baseR4>, in the text box and click set button.
2. Start Testing
   * Click Red button, it will change from “power” button to “rec” button. During rec is shown, run client application send all transaction to localhost:9292. The request will be redirected to [https://r4.smarthealthit.org](https://r4.smarthealthit.org/) or URL you set in above.
   * After all testing is done, click “rec” button and change button to “power”.
3. View Test Results
   * Click “Show Test Results button and show repot as below.
   * If no test data is found table record shows NO indication in the table.
   * When data is passed, the table indicates the result as below