



Testing FHIR Data Exchanges

SCORING SOFTWARE EXPLORATION REPORT
Trevor Andrus for JP Systems Data Quality

Outline

Introduction

-
- Purpose
 - FHIR
 - Mitre
 - ONC

Installation

-
- Process
 - Docker
 - WSL
 - Browser Alternative
 - Demonstration

Walkthrough

-
- Problems
 - FHIR Server
 - API compatibility
 - Results

Conclusion

-
- Possible applications
 - Data testing
 - Server testing
 - Resources

Introduction

What is Inferno?

Where did Inferno come from?

Who is responsible for Inferno's development?

What is **Inferno?**

"Inferno is an open source tool for testing data exchanges enabled by the Fast Healthcare Interoperability Resources (FHIR) standard."

- Github Documentation

In short, Inferno was created to test FHIR servers.

FHIR servers can consist of a combination of a few elements, but, generally speaking consist of two parts:

API - (Application programming interface) - a gateway that allows two applications to communicate - in this case, it gives permissions and allows access to patient data within databases.

Databases - This is where patient data is stored. Servers can contain millions of records that can be accessed through the API.

Inferno was designed to test both of these parts - it tests both how well the data is organized according to FHIR standards, but also how the FHIR server reacts to queries for data.

Where did **Inferno** Come from?

Inferno is the result of a collaboration between **ONC** (Office of the National Coordinator for Health Information Technology) and **MITRE** (A federally funded research and development organization).

MITRE was experimenting with scoring softwares in a project called Crucible. **ONC** contracted with **MITRE** to develop software to test a specific **ONC** implementation guide built upon the Crucible framework. The collaboration has resulted in two editions of Inferno:

Program Edition - An edition of Inferno aimed specifically at testing the **ONC Standardized API for Patient and Population Service**.

Community Edition - An edition of inferno that tests a number of FHIR Implementation Guides - users can either run existing Community Edition tests, or author their own tests using Inferno's open testing language.

ONC

- Office of national coordinator for health information technology
- Principal federal entity charged with coordination of health information technology and the electronic exchange of health information

MITRE

- FFRDC - Federally funded research and development center
- Sponsors both federally funded R&D centers and public-private partnerships
- Contracted with ONC to build Inferno on the model they developed in the Crucible project

Installing Inferno

Docker

WSL Containerization

Installing Inferno

Because of the open source nature of Inferno, the application is still in development and available through an online Github repository. To retrieve and use the program takes a couple more steps than simply downloading an .exe file.

Step 1: Download Docker

Through the instructions on Github, Inferno is only currently only available through Docker

Step 2: Enable WSL Compatability

Docker can only be implemented with a back-end powered by WSL: a linuxed based kernel for Windows operating systems

Step 3: Download the Inferno image, and run through Docker.

After these two preliminary steps are finished, the Inferno image can be downloaded and run.

Docker is a containerization platform designed to maximize ease of access to software in development

PROJECTS ARE ACCESSED THROUGH GITHUB, AND BUILT IN AN INDEPENDENT ENVIRONMENT TO PREVENT BUILD ERRORS

Rather than spend the time to optimize for multiple operating systems, Inferno developers use docker to allow the application to run on any operating system.

WSL

Containerization

(Noun), running (a program or application) in a discrete environment set up within an operating system specifically for that purpose and allocated only essential resources

Windows subsystem for Linux

The installation of Docker requires reconfiguration of Windows operating systems. This can be done through the Powershell. Virtualization capability must be enabled - meaning the native OS will allow a separate lightweight computer to be built on top of it.

Light and efficient

This eliminates the need for running an entire new operating system for a single application, and allows developers (such as those working on Inferno) to distribute their application in a controlled environment.

In short, **Docker** helps developers ensure their software runs as intended. In order to do so, we must enable the computer's containerization compatibility - this is often done by downloading the **Windows Subsystem for Linux (WSL)**.

WE WON'T GO OVER THE SPECIFICS OF DOCKER INSTALLATION

In order to install and have Docker working properly (with WSL compatibility) there is a bit of command line interface work that is required. Those familiar with Git will have the know-how necessary to get it working without issue.

Using Inferno

[Github Instructions](#)

[Imbedded Test Cases](#)

Inferno also allows for testing through a browser-based instance

THIS LIGHT WEIGHT VERSION ALLOWS US TO VIEW SIMPLE FUNCTIONALITY

For using inferno extensively, it is suggested to download the full version using the process we previously discussed, but the Inferno Github repository is equipped with a browser-based walkthrough to illustrate its concepts.

Walkthrough Commentary

A side by side look at the openly available Inferno Documentation

Walkthrough Outline

As provided on the Inferno Github page in the README file, these are the current instructions for creating a working instance of Inferno for testing.

- Step 1: Build a SMART-on-FHIR Sandbox
- Step 2: Open Inferno
- Step 3: Select FHIR Version and Enter FHIR Endpoint
- Step 4: Perform Discovery and Registration Tests
- Step 5: Perform Standalone Patient App Tests
- Step 6: Perform EHR Launch App Tests
- Step 7: Perform Data Access Tests
- Step 8: Review Results

Instructions Provided

- **Step 1: Build a SMART-on-FHIR Sandbox**
- **Step 2: Open Inferno**
- **Step 3: Select FHIR Version and Enter FHIR Endpoint**

Deprecated Results

This step for the most part goes as described - the server host, however has changed. Follow the redirection to the new host and continue as described.

After installation (or using the browser alternative) Inferno opens correctly

Select the FHIR version as described in the instructions, then copy the endpoint from the sandbox documentation

Instructions Provided

- Step 4: Perform Discovery and Registration Tests
- Step 5: Perform Standalone Patient App Tests
- Step 6: Perform EHR Launch App Tests
- Step 7: Perform Data Access Tests
- Step 8: Review Results

Deprecated Results

It seems that the testing stage of the instructions are where things start break down

Sandboxes can be created, but when linked to Inferno for testing, almost all tests fail.

This renders viewing results and performing data access tests impossible.

Despite Inferno's deprecated Github walkthrough, the developers have included **cloud-based test cases** within the software itself.

THIS ALLOWS US TO SEE WHAT A PROPER RESULT SHOULD LOOK LIKE

While further documentation may be required to get an instance of Inferno working on consumer data, the provided test cases gives a look at what the software is designed to do.

Inferno In Action

Demonstration with
Imbedded Test Cases

Landing Page

After traveling to <https://inferno.healthit.gov/>, this is the landing page we see to access the Inferno validator. We can see the two versions previously discussed.



Inferno is a rich and rigorous testing suite for HL7® Fast Healthcare Interoperability Resources (FHIR) to help developers implement the FHIR standard consistently. We offer two options for developers wanting to use Inferno – Inferno Program Edition and Inferno Community Edition.

ONC PROGRAM EDITION

Inferno Program Edition is a streamlined testing tool for services seeking to meet the requirements of the Standardized API for Patient and Population Services criterion § 170.315(g)(10) in the 2015 Edition Cures Update.

Try It Here

COMMUNITY EDITION

The Community Edition contains a community curated set of tests and tools for select FHIR Implementation Guides. Develop tests to meet your own use cases and contribute them to the community.

Community Tests

FHIR Resource Validator

Select Program Edition

To access the imbedded test cases, select the "Try it Here" option form the ONC program edition on the left.

ONC PROGRAM EDITION

Inferno Program Edition is a streamlined testing tool for services seeking to meet the requirements of the Standardized API for Patient and Population Services criterion § 170.315(g)(10) in the 2015 Edition Cures Update.

Try It Here

Download

COMMUNITY EDITION

The Community Edition contains a community curated set of tests and tools for select FHIR Implementation Guides. Develop tests to meet your own use cases and contribute them to the community.

Community Tests

FHIR Resource Validator

Download

Select Inferno Reference Server

Under the 'Enter a FHIR Endpoint' dropdown menu, select "Inferno Reference Server". This allows us to access the prebuilt test cases developed by the Inferno team.

These reference applications are provided by ONC for demonstration only.
Do not use to access sensitive data or Protected Health Information (PHI).
Data is removed every Sunday at 12:01am ET.



INFERNO PROGRAM EDITION

Test conformance to the Standardized API for Patient and Population Services criterion § 170.315(g)(10) in the 2015 Edition Cures Update.

Enter a FHIR Endpoint

START TESTING

Or test Inferno against ✓ a public sandbox
Inferno Reference Server

OVERVIEW SAMPLE DATA INSTALLATION INSTRUCTIONS RELEASE NOTES

Inferno Program Edition

Inferno Program Edition is a streamlined testing tool for Health Level 7 (HL7®) Fast Healthcare Interoperability Resources (FHIR®) services seeking to meet the requirements of the Standardized API for Patient and Population Services criterion § 170.315(g)(10) in the 2015 Edition Cures Update.

Open Source Issues Version 1.7.0

Run Tests

After selecting the Inferno Reference Server and clicking "Start Testing" You should be brought to the following page. Here we can see options for what tests Inferno will run, and under each option which tests will be performed. In this instance, we will be doing the Standalone Patient App tests - this tests if a third party app is able to access patient data on the FHIR server with proper authentication.

The screenshot shows the HealthIT.gov Inferno Program Edition interface. At the top, it displays the HealthIT.gov logo and a message: "These reference applications are provided by ONC for demonstration only. Do not use to access sensitive data or Protected Health Information (PHI). Data is removed every Sunday at 12:01am ET." Below this, the Inferno logo and "PROGRAM EDITION" are shown. To the right, it says "2015 Edition Cures Update - Standardized API Testing FHIR R4". The URL "https://inferno.healthit.gov/reference-server/r4" is in the address bar, along with "State", "Save", and "Help" buttons. A navigation menu below the address bar includes "STANDALONE PATIENT APP" (which is highlighted in orange), "LIMITED APP", "EHR PRACTITIONER APP", "SINGLE PATIENT API", "MULTI-PATIENT API", and "OTHER".

The main content area is titled "STANDALONE PATIENT APP - FULL PATIENT ACCESS" and "STEP 1 OF 6". It describes a scenario where a system performs a Patient Standalone Launch to a SMART on FHIR confidential client with a patient context, refresh token, and OpenID Connect (OIDC) identity token. After launch, a simple Patient resource read is performed on the patient in context. The access token is then refreshed, and the Patient resource is read using the new access token to ensure that the refresh was successful. The authentication information provided by OpenID Connect is decoded and validated, and simple queries are performed to ensure that access is granted to all USCDI data elements.

A large blue button labeled "► Run Tests" is present. Below it, three numbered steps are listed:

- 1 **SMART On FHIR Discovery** - 5 tests - [Show Details](#)
Retrieve server's SMART on FHIR configuration
- 2 **Standalone Launch With Patient Scope** - 10 tests - [Show Details](#)
Perform Standalone SMART launch sequence and test OpenID Connect and token refresh functionality.
- 3 **OpenID Connect** - 7 tests - [Show Details](#)
Use OpenID Connect ID token provided during launch sequence to authenticate user.

At the bottom, there are links for "Open Source", "Issues", and "Version 1.7.0".

Execute

After clicking to run tests, we are presented with the following information: This pop-up window defines the FHIR endpoint for testing (the imbedded test server), and the specifications we will be testing on. All of these options should be left as default to run on the provided test server.

https://inferno.healthit.gov/inferno/oauth2/static/launch

Input Fields

FHIR Endpoint
URL of the FHIR endpoint used by standalone applications
`https://inferno.healthit.gov/reference-server/r4`

Client ID
Client ID provided during registration of Inferno as a standalone application
`SAMPLE_CONFIDENTIAL_CLIENT_ID`

Client Secret
Client Secret provided during registration of Inferno as a standalone application
`SAMPLE_CONFIDENTIAL_CLIENT_SECRET`

Standalone Patient Scope
OAuth 2.0 scope provided by system to enable all required functionality
`launch/patient openid fhirUser offline_access patient/Medication.read patient/AllergyIntolerance.read patient/CarePlan.read patient/CareTeam.read patient/Condition.read patient/Device.read patient/DiagnosticReport.read patient/DocumentReference.read patient/Encounter.read patient/Goal.read patient/Immunization.read patient/Location.read patient/MedicationRequest.read patient/Observation.read patient/Organization.read patient/Patient.read patient/Practitioner.read patient/Procedure.read patient/Provenance.read patient/PractitionerRole.read`

Predefined Test Parameters 🔒

Public Client Confidential Client

Cancel **Execute**

Continue

After pressing continue in this pop up window, we will be directed to an external website to define some final specifications before receiving test results. In editions of Inferno downloaded through Docker, You would now be directed to a browser window.

Tests Running

Inferno will now redirect you to an external website for user authorization. For this test sequence to complete successfully, you will need to select a patient and authorize the Inferno client to access their data. Once you authorize the Inferno client to access patient data, you should be redirected back to Inferno. If something goes wrong, you can always return to Inferno at <https://inferno.healthit.gov/inferno/5AtXPnYmp1S>.

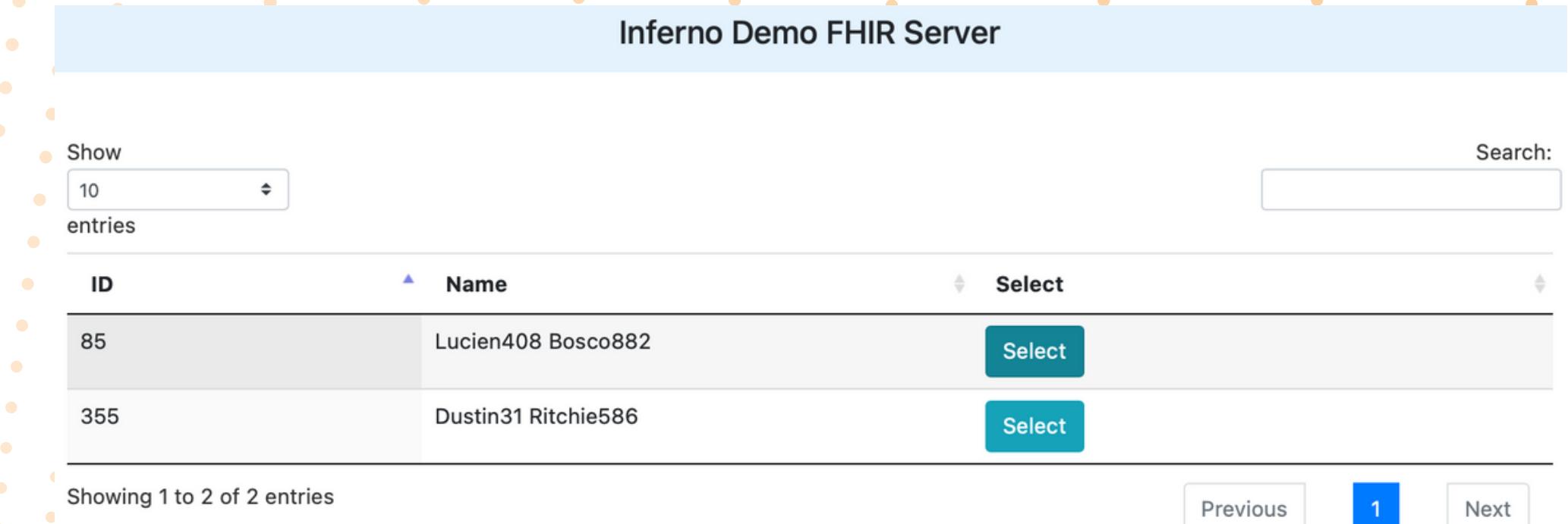
```
https://inferno.healthit.gov/reference-server/oauth/authorization?  
response_type=code&client_id=SAMPLE_CONFIDENTIAL_CLIENT_ID&redirect_uri=https%3A%2F%2Finfern  
o.healthit.gov%2Finferno%2Foauth2%2Fstatic%2Fredirect&scope=launch%2Fpatient+openid+fhirUser+offli
```

[Cancel Sequence](#)

[Continue](#)

Select a Patient

After selecting continue, we are brought to the following page. here we pick a test patient to access. You may be presented with a variety of options, but all should work the same. I picked the first name for simplicity.



The screenshot shows a web-based interface for the Inferno Demo FHIR Server. At the top, there is a blue header bar with the text "Inferno Demo FHIR Server". Below the header, there is a search bar labeled "Search:" with a placeholder text area. On the left side, there is a "Show" dropdown menu set to "10 entries". The main content area displays a table with two rows of patient data. The table has three columns: "ID", "Name", and "Select". The first row contains the ID "85", the name "Lucien408 Bosco882", and a "Select" button. The second row contains the ID "355", the name "Dustin31 Ritchie586", and a "Select" button. At the bottom of the table, it says "Showing 1 to 2 of 2 entries". To the right of the table, there are navigation buttons for "Previous", a page number "1" in a blue box, and "Next".

ID	Name	Select
85	Lucien408 Bosco882	Select
355	Dustin31 Ritchie586	Select

Select Authorizations

After selecting a patient, we are now asked to pick which authorizations to allow. (This simulates different 3rd party application authorizations). For example, if we unselect the "Medication Request" box, Inferno will give us an error and tell us we are unauthorized to access the medications for the patient.

Please select which scopes you would like to authorize:

- launch/patient
- openid
- fhirUser
- offline_access
- patient/Medication.read
- patient/AllergyIntolerance.read
- patient/CarePlan.read
- patient/CareTeam.read
- patient/Condition.read
- patient/Device.read
- patient/DiagnosticReport.read
- patient/DocumentReference.read
- patient/Encounter.read
- patient/Goal.read
- patient/Immunization.read
- patient/Location.read
- patient/MedicationRequest.read
- patient/Observation.read
- patient/Organization.read
- patient/Patient.read
- patient/Practitioner.read
- patient/Procedure.read
- patient/Provenance.read
- patient/PractitionerRole.read

Authorize

Access Results

After granting authorizations, Inferno will run tests to determine if the server abides by (In this case the ONC) implementation guides. It tests both the reactivity of the server, and the quality of the data contained within the database. From the tests shown, you can select a drop down menu to see the return codes for each individual test run.

The screenshot shows the Inferno HealthIT Reference Server interface. At the top, there's a navigation bar with a logo, the URL <https://inferno.healthit.gov/reference-server/r4>, and buttons for State, Save, and Help. Below the navigation is a breadcrumb trail: STANDALONE PATIENT APP > LIMITED APP > EHR PRACTITIONER APP > SINGLE PATIENT API > MULTI-PATIENT API > OTHER. A green checkmark icon indicates the current step is "STANDALONE PATIENT APP". To the right, it says "STEP 1 OF 6".
STANDALONE PATIENT APP - FULL PATIENT ACCESS
Result: Pass. Tests have successfully passed.
This scenario demonstrates the ability of a system to perform a Patient Standalone Launch to a SMART on FHIR confidential client with a patient context, refresh token, and OpenID Connect (OIDC) identity token. After launch, a simple Patient resource read is performed on the patient in context. The access token is then refreshed, and the Patient resource is read using the new access token to ensure that the refresh was successful. The authentication information provided by OpenID Connect is decoded and validated, and simple queries are performed to ensure that access is granted to all USCDI data elements.
[Rerun Tests](#) [Next ►](#)
SMART On FHIR Discovery - 5/5 Required Tests Passed - [Show Details](#)
Retrieve server's SMART on FHIR configuration
Standalone Launch With Patient Scope - 10/10 Required Tests Passed - [Show Details](#)
Perform Standalone SMART launch sequence and test OpenID Connect and token refresh functionality.
OpenID Connect - 7/7 Required Tests Passed - [Show Details](#)
Use OpenID Connect ID token provided during launch sequence to authenticate user.
Token Refresh - 5/5 Required Tests Passed - [Show Details](#)
Use refresh token to get new access token and verify it can access resources.
Unrestricted Resource Type Access - 14/14 Required Tests Passed - [Show Details](#)
Verify that patients can grant access to all necessary resource types.

Interpreting Results

Admittedly, the return codes and responses for individual tests are pretty dense for someone (like myself) unfamiliar with building and testing servers.

SMART On FHIR Discovery - 5/5 Required Tests Passed - [Show Details](#)
Retrieve server's SMART on FHIR configuration

Standalone Launch With Patient Scope - 10/10 Required Tests Passed - [Hide Details](#)
Perform Standalone SMART launch sequence and test OpenID Connect and token refresh functionality.

[Test Results](#) [Inputs](#) [HTTP Requests](#) [Outputs](#) [About](#)

- ✓ SPB-OSLS-01: OAuth 2.0 authorize endpoint secured by transport layer security [results...](#) ←
- ✓ SPB-OSLS-02: OAuth server redirects client browser to app redirect URI [results...](#)
- ✓ SPB-OSLS-03: Inferno client app receives code parameter and correct state parameter from OAuth server at redirect URI [results...](#)
- ✓ SPB-OSLS-04: OAuth token exchange endpoint secured by transport layer security [results...](#)
- ✓ SPB-OSLS-07: OAuth token exchange request succeeds when supplied correct information [results...](#) →
- ✓ SPB-OSLS-08: OAuth token exchange response body contains required information encoded in JSON [results...](#)
- ✓ SPB-OSLS-09: OAuth token exchange response includes correct HTTP Cache-Control and Pragma headers [results...](#)
- ✓ SPB-OSLS-10: Patient-level access with OpenID Connect and Refresh Token scopes used. [results...](#) →
- ✓ SPB-OSLS-11: Server rejects unauthorized access [results...](#) →
- ✓ SPB-OSLS-12: OAuth token exchange response body contains patient context and patient resource can be retrieved [results...](#) →

Additional Testing

After performing tests for Standalone patient apps, additional tests can be run for different use cases - API access, practitioner access, and other tests can be run to assess how the FHIR server conforms to the implementation guides.

The screenshot shows the HealthIT.gov INFERNO PROGRAM EDITION interface for the 2015 Edition Cures Update - Standardized API Testing (FHIR R4). The URL is https://inferno.healthit.gov/reference-server/r4. The navigation bar includes links for STANDALONE PATIENT APP, LIMITED APP, EHR PRACTITIONER APP, SINGLE PATIENT API (which is selected and highlighted in orange), MULTI-PATIENT API, and OTHER. There are also buttons for State, Save, and Help.

SINGLE PATIENT API

Result: Pass. Tests have successfully passed.

For each of the relevant USCDI data elements provided in the conformance statement, this test executes the required supported searches as defined by the US Core Implementation Guide v3.1.1. The test begins by searching by one or more patients, with the expectation that the Bearer token provided to the test grants access to all USCDI resources. It uses results returned from that query to generate other queries and checks that the results are consistent with the provided search parameters. It then performs a read on each Resource returned and validates the response against the relevant profile as currently defined in the US Core Implementation Guide. All MUST SUPPORT elements must be seen before the test can pass, as well as Data Absent Reason to demonstrate that the server can properly handle missing data. Note that Encounter, Organization and Practitioner resources must be accessible as references in some US Core profiles to satisfy must support requirements, and those references will be validated to their US Core profile. These resources will not be tested for FHIR search support.

Capability Statement - 5/5 Required Tests Passed - [Show Details](#)
Retrieve information about supported server functionality in the Capability Statement.

Conclusion

Impressions

Applications

Resources

Inferno Impressions

While still in development, Inferno seems a promising tool in testing the production of FHIR servers. However, the extent to which JP Systems may have use of it is questionable.

Inferno is mainly aimed at testing API and Server compatibility

Server Development - As opposed to programs like Diameter health that JP Systems currently uses, Inferno seems to be more aimed towards those developing new FHIR servers. More attention (and tests) are devoted to testing how servers respond to requests than the quality of data contained in the database.

Data Quality - While there are some tests devoted to the quality of data stored within the FHIR servers, this does not seem to be the main purpose of Inferno. While further development may include increased attention to data quality, it currently takes the back seat to server compatibility testing.

Inferno Applications

For JP Systems data quality, the current applications of Inferno are limited. It is incredibly difficult to get a working instance of the software outside of the built in test cases, and the analysis of data quality is not currently a priority.

Further development is needed before Inferno can be used effectively

Server Testing - Even in its applications as a server testing software, Inferno has a long way to come before it can be used extensively. The lack of documentation (and deprecated instructions on Github) make running the software difficult. Once running, the test results are also difficult to decode, and provide little help to those not trained technically in server architecture.

Data Quality - As was previously stated, data quality is not currently the priority for Inferno, and until (or if) improvements to this are made, Inferno may not be of much use for JP Systems data quality.

Sources

ONC is FHIR'd up: Unwrapping the new Inferno Testing Suite - Health IT Buzz

<https://www.healthit.gov/buzz-blog/interoperability/onc-is-fhird-up-unwrapping-the-new-inferno-testing-suite>

Inferno Program Edition Documentation

<https://github.com/onc-healthit/inferno-program>

Inferno Community Edition Documentation

<https://github.com/onc-healthit/inferno>

Github Walkthrough

<https://github.com/onc-healthit/inferno/wiki/Walkthrough>

Inferno

<https://inferno.healthit.gov>