FHIR Bulk Data Test Suite Design

SMART Health IT Team

Boston Children’s Hospital

Computational Health Informatics Program
Introduction

SMART on FHIR is an open, standards-based technology platform that enables innovators to create apps that seamlessly and securely run across the healthcare system. Using an electronic health record (EHR) system or data warehouse that supports the SMART and FHIR standards, patients, doctors, and healthcare practitioners can draw on this library of apps to improve clinical care, research, and public health. The SMART Health IT team builds and maintains several tools for use by the SMART on FHIR developer community (https://dev.smarthealthit.org/), such as the SMART sandbox and SMART App Gallery, as well as a FHIR Bulk Data Server which includes a web client and a command-line tool. The Bulk Data Server and tools are open to the community and can be used to test third-party tools and applications against a large simulated FHIR bulk dataset.

Inferno (https://inferno.healthit.gov/inferno/) is an open-source tool designed by the MITRE Corporation to test SMART on FHIR servers, and verify that these servers meet the API standards required by the Office of the National Coordinator for Health Information Technology (ONC). The tools are intended to be a one-stop-shop for easy testing of various FHIR servers.

The ONC has tasked the SMART Health IT team with producing an extendable prototype test suite to verify compliance with the flat FHIR bulk data specification.

Test Suite

The Bulk Data Tester (BDT) is a universal toolbox for testing bulk data servers. It is made out of multiple independent modules that can be used separately. There is an online version, but those who don’t want to enter their credentials online can also download and run it locally.

Below is the current draft high-level diagram of the Bulk Data Tester, followed by a short description of the most important modules.
A large collection of tests will be designed to run against bulk-data servers. The tests will be stored in a separate repository to allow for community contributions.

The Server is the backend of the online Bulk Data Tester instance. It exposes a Restful API to register and manage new bulk data servers and generate comparison reports, as well as small Socket RPC API to execute one or more tests on demand and receive test results. It has a database to store the results of the last test run for each registered server, as well as the server configurations.

The Client is the front-end of the online Bulk Data Tester instance. It provides browser UI for the test runner and server comparisons and reports. It will also include the forms and pages need for registering and configuring a target
server. The configuration of the tested server can be stored in the online database by authenticated users (site managers), or it can remain local, persisted into the browser’s local storage.

The client connects to the server via WebSocket so that we can receive multiple responses from a single test run request in real-time. Below is a screenshot of the BDT UI in its current state:
The Test Runner is the application that executes the bulk-data tests. It is a standalone application that can be executed in the terminal. It can also be used as a JavaScript module and embedded into other JS systems. At its core the runner is just a test loader and an event emitter, meaning that it knows how to walk a directory recursively and collect and execute test files. While executing tests the runner generates various events which makes it very easy to extend. Different clients can “listen” to those events and react appropriately. For example, in this diagram, there is a “stdout reporter” that displays the test results in the terminal (on the screenshot below) or the “Database reporter” that records statistics in a database.
 Integration With Third-Party Systems

While there are many similarities between the SMART Bulk Data Tester and Inferno, and the SMART team has considered the benefits of using Inferno instead of developing BDT, there would be a significant development timeline to make such upgrades. One of the biggest challenges to this approach is that Inferno is written in Ruby, while the SMART team is using JavaScript, and does not currently have the Ruby experience to make the necessary updates.
and changes to Inferno. Our team has created a stack primarily using JavaScript, which should have better support for asynchronous tasks.

To reduce the complexity of integration with third-party systems like Inferno, we have created a dedicated public API that allows any system to use the SMART software as a testing service. This covers a broad spectrum of use cases:

- Can be used online to test public servers
- Can be used online to test local servers anonymously
- Can be installed and used locally
- Can be used from the command line (CLI)
- Can be used as a service
  - From another site
  - From CLI
  - From any script or test runner (regardless of the programming language)

The diagram below shows how the Bulk Data Tester API (BDT API) can be called from an external browser-based client (such as Inferno) to run tests and pass results back for display to the end user of the external third-party system.
External Client UI

on Load

The client uses the JSON structure to render an empty list of tests

Execute one test or group of tests

Update UI to render test results

External Client

The client requests the list of available tests

Execute tests and reply with results

BDT API

The server responds with nested JSON structure describing the available tests