



Test Driven Development I - Intro - Exercises

99 - Touchstone TestScript Execution - Connectathon Track

Track lead: Richard Ettema

During this hands-on session of the Test Driven Development I - Intro tutorial we will examine the proposed HL7 FHIR Connectathon 17 Test Tracks and begin the analysis and design of potential TestScripts based on the track's test scenario(s). The goal is to choose a track where:

- There are currently no TestScript(s) defined, or
- The current TestScript(s) for the track may require additional test definition, or
- Additional TestScript(s) may be needed for the track

TestScript Definition

TestScript definition begins with defining the use case that will be tested. Once defined, the use case can then be analyzed and used to define corresponding test case(s) or TestScript(s). A useful process to follow is to simply follow the TestScript structure and “fill in the blanks”. Here are some guidelines for filling them in:

Fixtures

A set of resource instances that will be used during TestScript execution.

- ❖ Represent the expected, known and valid test data
- ❖ Will typically require use case domain knowledge to define

Profiles

A set of FHIR profiles (StructureDefinitions) that will be used during TestScript execution of profile asserts.

- ❖ Each profile defines a StructureDefinition instance by URI where the URI SHALL reference a known FHIR profile
- ❖ Use the FHIR Resource base profiles for validation against the base FHIR specification
- ❖ If external profiles are needed, you will need to coordinate with Touchstone Support

Variables

A set of expressions whose evaluations will be used in substitutions during TestScript execution.

- ❖ Extracts data values from static fixtures (defined in the fixtures section/element)
- ❖ Extracts data values from dynamic fixtures (operation request and/or response payloads)
- ❖ Used in operations and asserts to provide context relevant data values; e.g. operation parameters, assert comparison values

Setup

The setup section is executed to establish the testing environment.

- ❖ The purpose of the setup section is typically to pre-load required data or delete data on the FHIR system(s) under test
- ❖ The setup operations are executed once before all the tests are run
- ❖ All operations in a setup section (including assertions) must complete successfully for the subsequent tests to be executed

Test

Each test section is a set of actions – operation and assert – that are executed and evaluated.

- ❖ Operations are always executed regardless of any asserts that may follow
- ❖ When an operation or assertion fails, subsequent operations and assertions are not executed
- ❖ When a test fails, subsequent tests are still executed
- ❖ Because tests are typically interdependent, it is best practice to fix the cause of the first test failure instead of analyzing all test failures for a single test execution

Teardown

The purpose of the teardown section is to revert the FHIR server under test to a pre-test clean state.



Test Driven Development I - Intro - Exercises

99 - Touchstone TestScript Execution - Connectathon Track

Track lead: Richard Ettema

FHIR Subscriptions (First Analysis Example)

http://wiki.hl7.org/index.php?title=201801_FHIR_Subscriptions

Let's examine the FHIR Subscriptions Test Track and use the first test scenario as an analysis example.

First Test Scenario - Create FHIR Subscription

Action: Subscribing client creates a Subscription instance on FHIR Server [snipped].

Precondition: n/a

Success Criteria: Subscribing client POSTs new subscription resource to FHIR server. FHIR Server persists active subscription resource. Subscribing client can read newly created Subscription.

Bonus point: [Ignored for this example]

Analysis / Re-definition - Create and Validate FHIR Subscription

Actors

- ❖ Touchstone (T) - Test Platform, default FHIR Client
- ❖ Subscriptions System (SS) - FHIR Server supporting Subscription resource create, read, update, delete and search

Test Data

- ❖ Subscription static fixture - used to create the Subscription resource instance on (SS)

Setup

- ❖ Delete any existing Subscription resources that match the new Subscription to be created on (SS)

Test - Create Subscription

Create a new Subscription resource instance on the target Subscriptions System

- ❖ (T) sends Subscription resource to (SS)
 - Operation 'create'
 - HTTP POST [SS Base URL]/Subscription
 - Sent Subscription status is 'requested'
- ❖ (SS) responds with conformant FHIR API success

Test - Search Subscription

Search for the created Subscription resource instance on the target Subscriptions System to validate the Subscription resource has been created and processed correctly

- ❖ (T) sends search request for Subscription to (SS)
 - Operation 'search'
 - HTTP GET [SS Base URL]/Subscription?type=\${subscriptionType}&uri=\${subscriptionUri}
- ❖ (SS) responds with conformant FHIR API success and payload of a searchset Bundle containing the created Subscription
 - Validate returned Subscription status is now 'active'
 - Evaluate Subscription resource id and save in TestScript variable

Test - Read Subscription

Read the created Subscription resource instance on the target Subscriptions System to validate the Subscription resource has been created and processed correctly

- ❖ (T) sends search request for Subscription to (SS)
 - Operation 'read'
 - HTTP GET [SS Base URL]/Subscription/\${subscriptionResourceId}
- ❖ (SS) responds with conformant FHIR API success and payload of the created Subscription
 - Validate returned Subscription status is now 'active'