Test Driven Development II - Advanced - Exercises

01 - Reliable and Repeatable Testing - 2 Users, Same Data

Track lead: Richard Ettema

During this hands-on session of the Test Driven Development II - Advanced tutorial we will examine a series of TestScripts that illustrate how to ensure reliable and repeating testing. The scenario is where 2 (or more) users execute the same TestScript against the same Test System using the same (static) data.

TestScripts for this exercise are in the* ***FHIR3-0-1-DevDays17/TDD-2-Advanced/01-ReliableTesting* *Test Definitions*

2 Users, Same Data Scenario

The use case for this test scenario involves two steps:

1. Create a new Patient resource instance using a known, static fixture
2. Search for the created Patient based on the known data from the static fixture

The success outcome is that the search operation will return a single match of the created Patient.

Now let’s examine what happens when this test scenario is executed by 2 different users against the same test system:

- ✓ User 1 executes this test and the search returns the expected single match of the created Patient.
- ✗ User 2 then executes this test but now the search returns two matched Patients – the one created by User 1 and by User 2. **The test definition and conditions did not change but, this is <u>not</u> the expected outcome.**

The following four TestScripts illustrate the progressive test definition techniques designed into the TestScript resource and a FHIR Test Engine that demonstrate this original issue and the means to resolve this issue.

TDD-2-Adv-01-Reliable-01-patient-create-search-static-[xxx]

Test creating a static Patient fixture and then searching for the created Patient using known, hard-coded values from the static fixture.

Features

- ❖ Defines the original test scenario without any modifications; *examine and become familiar with the TestScript*
- ❖ Uses a static fixture; *examine the fixture and note the static contents; i.e. the hard-coded data values in the static contents is part of the problem*
- ❖ Examine the various asserts; specifically, look at the last assert on the search operation; *this assert verifies that the search only returns a single Patient match; notice the ‘warningOnly’ setting of true that will allow the TestScript to not fail but simply report a warning; this assert really should <u>not</u> have the ‘warningOnly’ setting*

TDD-2-Adv-01-Reliable-02-patient-create-search-variables-[xxx]

Test creating a static Patient fixture and then searching for the created Patient using <u>a value stored as a variable from the static fixture</u>.

Features

- ❖ Defines the test scenario introducing the use of variables; *examine and become familiar with the TestScript*
- ❖ Continues to use a static fixture; *examine the fixture and note the static contents; i.e. the hard-coded data values in the static contents is part of the problem*
- ❖ Uses variables; *how does the use of variables begin to address the problem?*

TDD-2-Adv-01-Reliable-03-patient-create-search-placeholders-[xxx]
Test creating a static Patient fixture containing placeholder variables and then searching for the created Patient using a value stored as a variable from the static fixture.
Features

- Defines the test scenario now adding the use of placeholder variables within the static fixture; *examine and become familiar with the TestScript*
- Continues to use variables; *how does the use of variables begin to address the problem?*
- Continues to use a static fixture but now containing placeholder variables for data values; *examine the fixture and note the use of placeholders instead of hard-coded data values; i.e. this is part of the solution*

★ Key Concept: Placeholders - a special type of variable

- **Placeholders are character strings of varying lengths from 1 to 20 and are of three types: (C)haracter, (D)igit and (CD) mixed; e.g. ${CD12}**
- **Placeholders define unique data for each user so multiple users can run the same TestScript(s) concurrently against the same test system without interfering with each other's results**
- **To see your assigned placeholder values in Touchstone, go to your username menu item and select "${} My Placeholders"**

2 Users, Same Data Issue Resolved, But…
At this point the issue of 2 (or more) Users, Same Data has been addressed. However, there is still one remaining issue → **Same User, Same Data**. This issue will continue to occur for the same user because of the same placeholder values for a single user. So, one more modification is needed.

TDD-2-Adv-01-Reliable-04-patient-create-search-setup-[xxx]
Test using the setup section with a Patient conditional delete before creating a static Patient fixture containing placeholder variables and then searching for the created Patient using a value stored as a variable from the static fixture.
Features

- Defines the test scenario now adding the setup section to conditionally delete the Patient resource with matching data; *examine and become familiar with the TestScript*
- Continues to use variables; *how does the use of variables begin to address the problem?*
- Continues to use a static fixture but now containing placeholder variables for data values; *examine the fixture and note the use of placeholders instead of hard-coded data values; i.e. this is part of the solution*
- Introduces the use of the setup section to initialize the FHIR system under test; *this is the final piece of the solution*

★ Key Concept: Touchstone, TestScript setup and Conditional Deletes

- **The FHIR specification defines conditional deletes; however, not all FHIR systems/servers may support this feature.**
- **Because FHIR systems/servers are more likely to support the search operation and delete of single resources, Touchstone implements the conditional delete within the setup section only as follows:**
  - **First, perform a search operation using the conditional delete criteria/search parameters**
  - **Second, delete all found matching resources using individual delete operations**
  - **Then, perform the search operation again; if matches are found, delete them; if no matches, then proceed to the next setup operation if present**
  - **If matches were found in the 2nd search, perform the search operation a third and final time; if search matches are repeatedly found even after deletes (reported as successful by test system), report an error that the FHIR system under test may have an issue**

Have fun, and remember to ask questions if you need help!