Test Driven Development II - Advanced - Exercises

02 - Complex Asserts - 02 - Rules

Track lead: Richard Ettema

During this hands-on session of the Test Driven Development II - Advanced tutorial we will examine TestScripts that show the use of Rules and the Touchstone Rules Engine.

TestScripts for this exercise are in the **FHIR3-0-1-DevDays17/TDD-2-Advanced/02-ComplexAsserts Test Definitions*

Test Scenario

The use case for this test scenario involves two steps:

1. Create a new Patient resource instance using a known, static fixture
2. Read the created Patient using the _summary=true option

The success outcome is that the returned Patient instance only contains the FHIR indicated summary elements.

TDD-2-Adv-02-Complex-02-rules-patient-read-summary-[xxx]

Create a new Patient resource fixture in the setup section and then read the created Patient using the _summary=true option.

Features

- Uses the setup section to initialize the FHIR system under test by creating the Patient resource instance needed for the subsequent read test; *examine and become familiar with the TestScript*
- Defines the referenced rule; *observe that there are no rule parameters - why?*
- Examine the various read operation asserts that test the returned Patient contents; specifically, the assert that calls the defined rule

The focus of this test is to illustrate the use of Rules and the Touchstone Rules Engine to simplify the TestScript assert definitions; i.e. reduce and consolidate the number of individual asserts.

Key Concept: Touchstone Rules Engine and Rule Language Support

- **Touchstone Rules Engine has access to the complete executed operation context**
- **Current scripting languages supported are Groovy, Schematron and XSLT**
  - **Groovy is the recommended language**
  - **Schematron and XSLT are only valid for XML formatted content**
- **The Touchstone Test Definitions UI provides a filter mechanism that allows the user access to uploaded rule definitions**

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