

Mandatory Assignment 2

User Driven Acceptance Testing

Andreas Dan

Benedict Marien

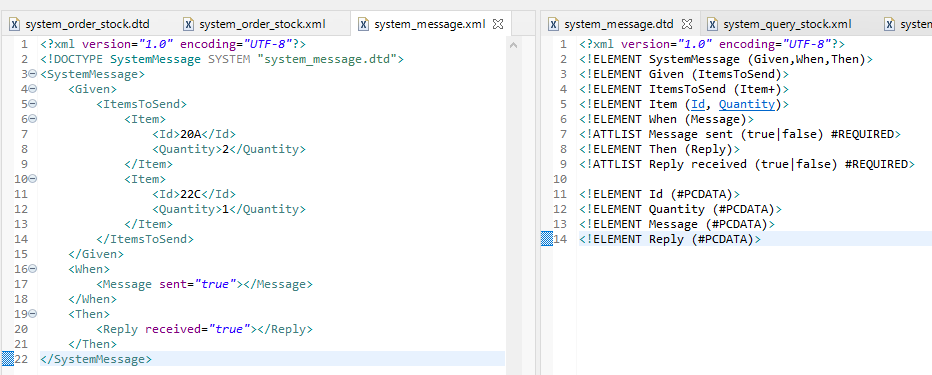
04/05/2021

General XML Structure

As the XML will serve users to write Acceptance Tests, the layout ideally reflects the following widely accepted format for Acceptance Tests – Given, When, Then:

* Given: A set of initial circumstances or state.
* When: An event happens.
* Then: The expected result.

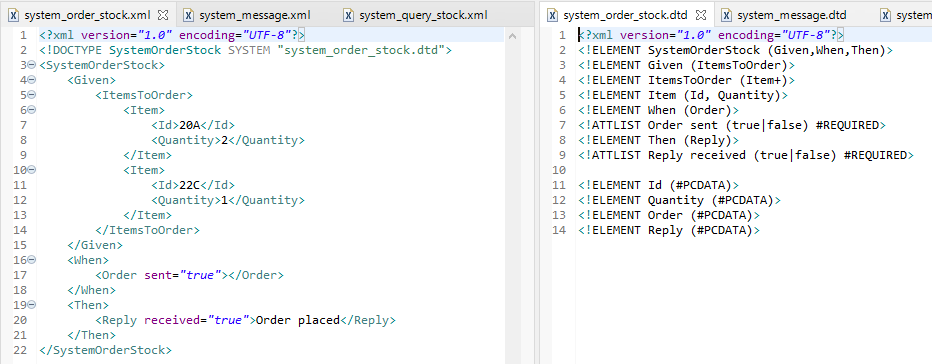
System 1 – Message for sending materials



Applying the aforementioned format and translating it to the XML file we arrive at the following structure where the contents withing the *<Given>* element are the materials that need to be sent. The *<ItemsToSend>* element encapsulates these individual materials which in turn are represented in the *<Item>* elements with the necessary data.

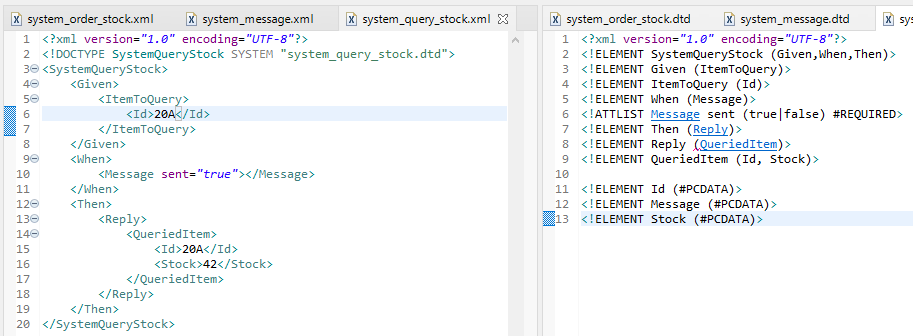
Once the structure was done the Document Type Definition (DTD) was defined to restrict the elements and attributes that can be present in the XML file, and then linked.

System 2 – Order stock

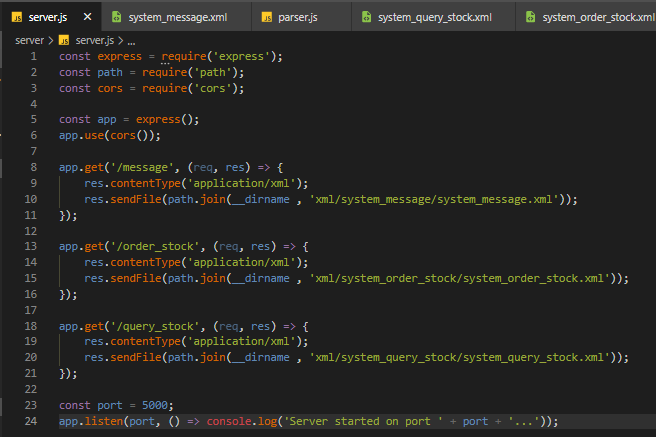


This XML and definition file are almost identical as the last with the exception of the *<ItemsToSend>* element which was renamed to *<ItemsToOrder>*.

System 3 – Query stock



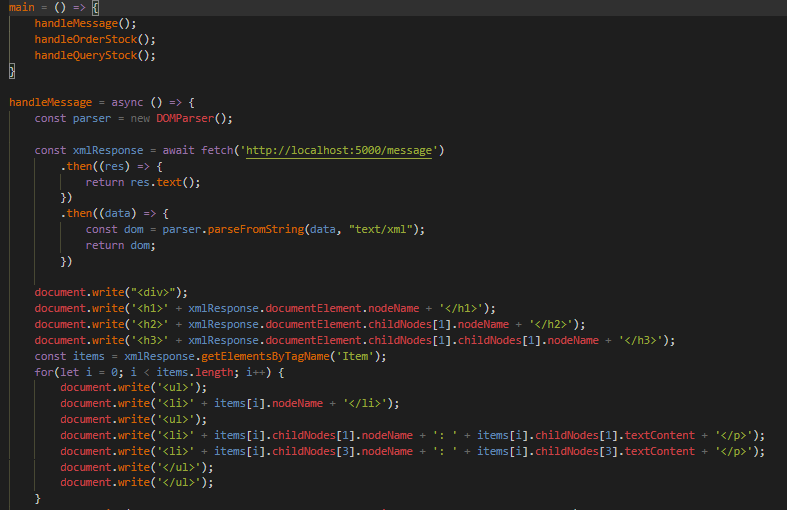
For this system the structure was changed to include *<ItemToQuery>* which takes data to identify the stock and the expected result in the *<Then>* element comes in the form of a reply from the Depot in the *<Reply>* element which includes the queried item and the information about it.

Parser

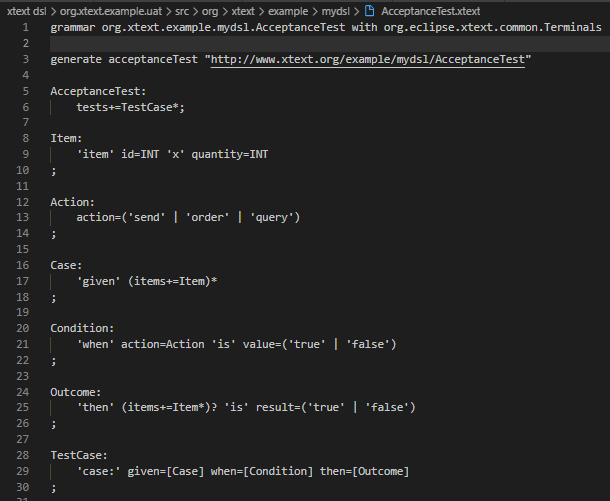
The XML files and their “.dtd” counterparts are hosted on a local Node.JS server which the client, running JavaScript, then fetches from one of three defined endpoints.

For this reason, some NPM packages were installed such as the Express framework and CORS.

When the XML file is fetched, the frontend takes care of extracting the data and writing to the DOM, the result of the parsing can then be viewed by opening the *index.html* file in the browser.



Non-XML Domain Specific Language

The following DSL was defined using *xText*, the Eclipse plugin. The DSL is set up to mimic the behaviour of the XML Domain Specific Language mentioned earlier to provide consistency and adhere to the *Given-When-Then* format.

Starting with the *TestCase*; the *given* clause takes one or more *Item*(s), which in turn carry the information about the item in question such as the item ID and the quantity.

Next, the *when* clause takes an action verb which can only be *send*, *order* or *query* and a value; *true* or *false*.

Finally, the *then* clause provides the outcome of the test and the expected result. The *result* attribute is either *true* or *false* and the clause can optionally hold *items* which is needed in the Query system.

GitHub Repository

<https://github.com/Aviater/user_acceptance_tests>