**SoapUI:**

SoapUI – Простой протокол доступа к объектам. SoapUI **работает только с xml**, а так же, в отличие от REST, который работает по протоколу HTTP, SoapUI может работать с разными протоколами.

Еще про отличие REST от SoapUI: REST работает с ресурсами, SoapUI работает с операциями, которые выполняют какую-то логику с помощью интерфейсов. REST быстрее.

WSDL документ – описывает операции и то как они должны выглядеть.

XSD – схемы на основании которых стоится WSDL документ:

Как тестировать с помощью SoapUI:

Мы должны загрузить WSDL, долее выполняем запрос. Как правило, у нас должна быть какая-то документация, которая говорит какие значения мы может вставить в WSDL.

**Описание запроса:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="http://www.oorsprong.org/websamples.countryinfo">

<soapenv:Header/>

<soapenv:Body>

<web:CapitalCity>

<web:sCountryISOCode>RUS</web:sCountryISOCode>

</web:CapitalCity>

</soapenv:Body>

</soapenv:Envelope>

Envelope- корневой тег, который определяет пространство имен.

Header – в soap этот тег не обязательный.

Body – обязательный тег, в этом теги написан запрос

Web – в нем мы пишем название операции.

**Подытожим:**

Основная цель тестирование web сервисов заключается в том, чтобы мы убедились, что мы получаем правильные ответы на отправленные запросы. Когда мы работаем с web-service у нас нет графического интерфейса и нам приходится использовать дополнительный софт (Postman, SoapUI, JMeter).

**Что нужно для тестирования Soap service:**

1. Мы должны знать операции, которые предоставляет сервис
2. Мы должны знать, какие параметры используются для этих операций

План моих действий, как я должен тестировать:

Предположим, мы загрузили WSDL документ, SoapUI сгенерировал нам разные операции и в одной из операций будет два запроса, один будет обязательный, другой необязательный. Так вот

При тестировании этих запросов мы должны:

1. Протестировать операцию с одним обязательным запрос и убедится, что все в порядке
2. Протестировать операцию с двумя обязательным запрос и убедится, что все в порядке
3. Протестировать операцию с одним необязательным запрос и убедиться, тест не работает
4. Убрать и обязательный и необязательный запрос и убедиться, что запрос не работает
5. Проверить типы данных, если запрос предполагает ввести string, то мы должны ввести string и убедиться, что все в порядке
6. Проверить типы данных, если запрос предполагает ввести string, а мы вводим number то нам нужно убедиться, что тест не проходит