МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА» (БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №2

**тема: «SoapUI как инструмент эмуляции сервисов»**

по дисциплине «Тестирование программных систем»

Выполнил: ст. группы ПВ-202

Буйвало А.А.

Проверили:

Лебединская Анастасия Александровна

Бабенко Анастасия Александровна

Белгород 2023 г.

***Цель работы:*** изучить принципы работы со средством эмуляции работы веб-сервисов SoapUI.

**Задание:**

Для операции CheckAccount реализовать:

- Возвращать Close, Blocked, Open при получении определенных значений AccountNumber (значения выбрать самостоятельно).

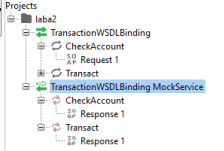
- Для остальных значений Account Number возвращать ошибочный ответ.

- В AccountNumber всегда возвращать входное значение.

- Name и Surname в запросе указывать свое.

***Выполнение:***

Примеры запросов и Mock-сервис:



В сервисе две операции: проверить счет (CheckAccount) и провести перевод (Transact).

Операция CheckAccount, входные параметры:

•AccountNumber – номер счета;

•Name – имя клиента;

•Surname – фамилия клиента;

•Passport – номер паспорта.

Параметры ответа:

•AccountNumber – номер счета;

•Status – статус (может принимать значения: Close, Open, Blocked). Операция Transact, входные параметры:

•AccountNumber

•Sum

•Target

Параметры ответа:

•TransactionId – идентификационный номер транзакции;

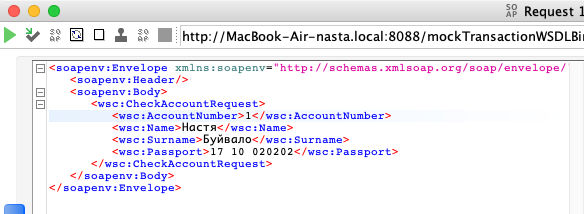
•TransactionStatus – статус (Success).

Каждый сервис может вернуть ошибку состоящую из:

•ErrorCode – код ошибки;

•ErrorString – описание ошибки.

Запрос имеет вид:



В ответ на данный запрос может прийти два вида ответа:

Success

<soapenv:Envelope

xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:wsc="http://bank/schema/WSCommonTypes">

<soapenv:Header/>

<soapenv:Body>

<wsc:CheckAccountResponse>

<wsc:AccountNumber>${AccountNumber}</wsc:AccountNumber> <wsc:Status>${Status}</wsc:Status>

</wsc:CheckAccountResponse>

</soapenv:Body>

</soapenv:Envelope>

Error

<soapenv:Envelope

xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wsc="http://bank/schema/WSCommonTypes">

<soapenv:Header/>

<soapenv:Body>

<wsc:CheckAccountResponse>

<wsc:AccountNumber>${AccountNumber}</wsc:AccountNumber> <wsc:ErrorCode>666</wsc:ErrorCode>

<wsc:ErrorString>Ошибка</wsc:ErrorString>

</wsc:CheckAccountResponse>

</soapenv:Body>

</soapenv:Envelope>

**Код программы:**

def groovyUtils = new com.eviware.soapui.support.GroovyUtils(context);

def holder = groovyUtils.getXmlHolder(mockRequest.getRequestContent());

holder.declareNamespace("wsc", "http://bank/schema/WSCommonTypes");

def accountNumber =

holder.getNodeValue("//wsc:CheckAccountRequest/wsc:AccountNumber"); context.AccountNumber = accountNumber;

if(accountNumber == "1"){

context.Status = "Open";

return "Success";

}

if (accountNumber == "2"){

context.Status = "Close";

return "Success";

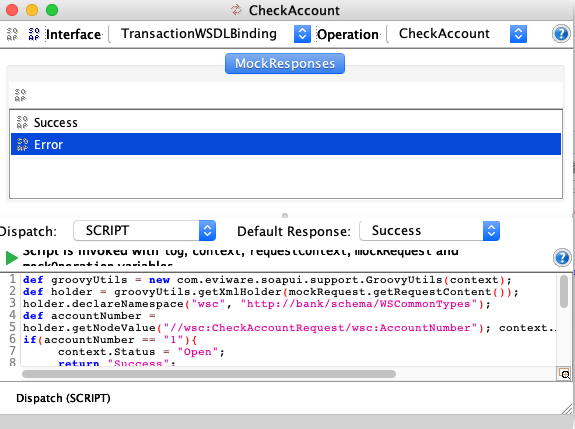
}

if (accountNumber == "3"){

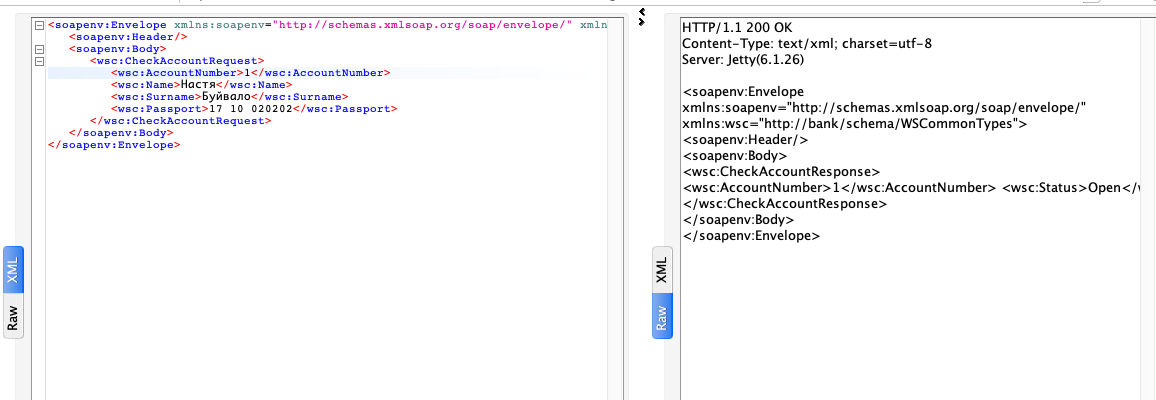
context.Status = "Blocked";

return "Success";

}

return "Error"; 

**Скриншоты работы программы**

AccountNumber = 1 -> Open 

AccountNumber = 2 -> Close 

AccountNumber = 3 -> Blocked



AccountNumber = 4 -> error



**Вывод**

В ходе выполнения данной работы были изучены принципы работы со средством эмуляции работы веб-сервисов SoapUI. Также на примере запроса checkAccount был получен опыт в написании скриптов на groovy для эмуляции работы веб-сервиса и параметризованного ответа зависящего от запроса.