

Федеральное государственное автономное образовательное
учреждение высшего образования

Университет ИТМО

Дисциплина: Сервис-ориентированная архитектура

Лабораторная работа 4

Вариант 621997122

Выполнили:

Кривоносов Егор Дмитриевич

Группа: Р34111

Преподаватель:

Райла Мартин

2023 г.

Санкт-Петербург

Оглавление

Задание	3
Выполнение	3
Исходный код	3
Инструкция как настраивать Mule в Anypoint	3
Инструкция как задеплоить проект Mule на сервер	3
Решение проблем с CORS:	3
Mule	4
Настройки Listener	5
Настройки Consume	9
Примеры Transform Message (входящий и исходящий)	10
Пример Error Propagate (Transform Message)	11
Кастомные код ошибок:	11
Инструкция для развертывания:	13

Задание

Лабораторная работа #4

Введите вариант: 621997122

Внимание! У разных вариантов разный текст задания!

Переработать сервисы из лабораторной работы #2 следующим образом:

- Первый ("вызываемый") сервис переписать в соответствии с требованиями протокола SOAP.
- Развернуть переработанный сервис на сервере приложений по собственному выбору.
- Оставшийся сервис не модифицировать, не менять его API, протокол и используемый сервер приложений.
- Установить и сконфигурировать на сервере Helios программное обеспечение Mule ESB.
- Настроить интеграцию двух сервисов с использованием установленного программного обеспечения.
- Реализовать дополнительную REST-"прослойку", обеспечивающую возможность доступа к переработанному сервису клиентского приложения без необходимости его модификации. Никакой дополнительной логики, помимо вызовов SOAP-сервиса, разработанная REST-прослойка содержать не должна.

Выполнение

Исходный код

<https://github.com/RedGry/SOA/tree/lab4>

Инструкция как настраивать Mule в Anypoint

■ Tutorial_po_4_labe_SOA.pdf

Инструкция как задеплоить проект Mule на сервер

[Ссылка](#)

Решение проблем с CORS:

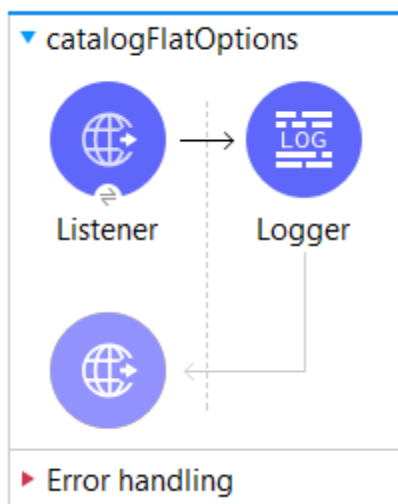
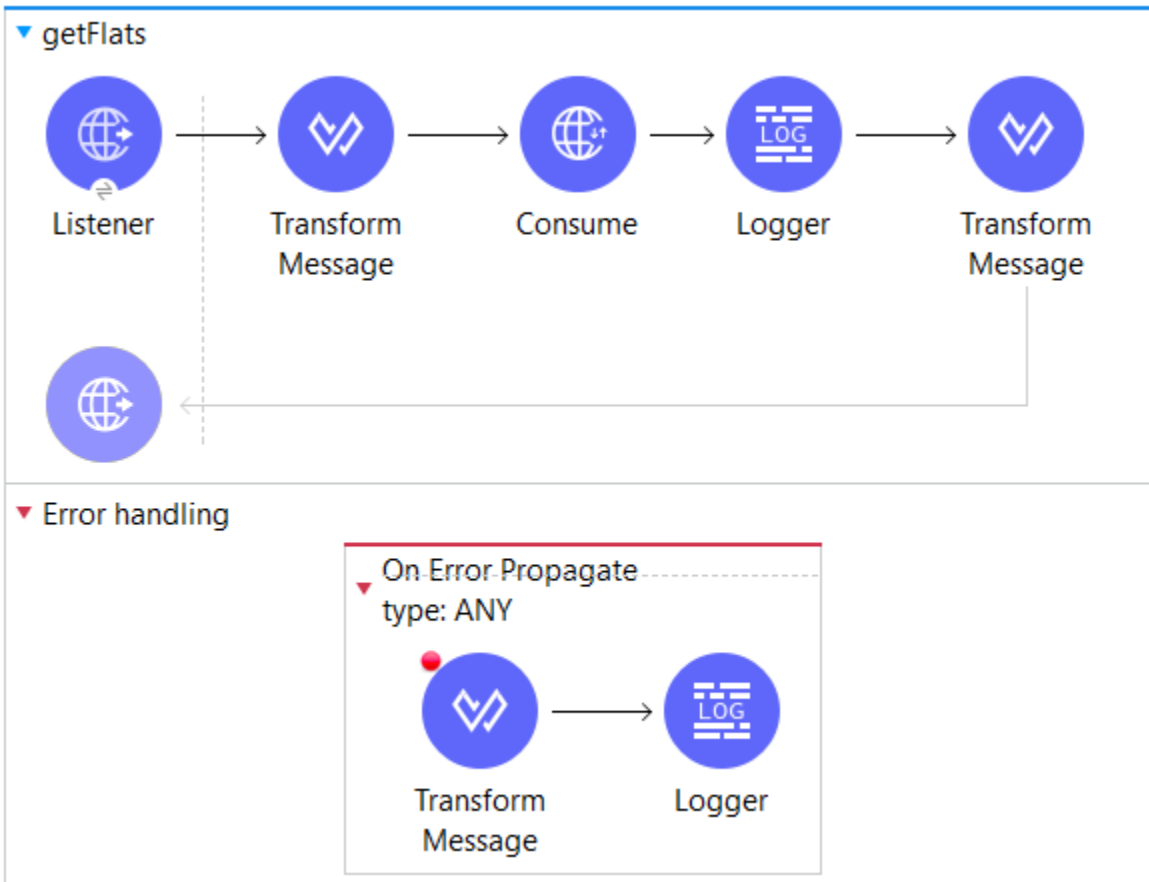
Добавить Flow + Listener на метод OPTIONS

Добавить во все PUT, POST, DELETE, OPTIONS Listener'ы заголовки:

```
output application/java
---
{
    "Access-Control-Allow-Origin" : "*",
    "Access-Control-Allow-Methods" : "GET, PUT, POST, DELETE, HEAD, OPTIONS",
    "Access-Control-Allow-Headers" : "*",
    "Access-Control-Allow-Credentials" : "true"
}
```

Mule

На каждый метод настраиваем свой Listener и указываем URI



Настройки Listener

The image shows two screenshots of the Mule IDE's 'Listener' configuration window. The top screenshot shows the 'General' tab with the following settings:

- Display Name: Listener
- Basic Settings: Connector configuration: HTTP_Listener_config
- General: Path: /catalog/flats

The bottom screenshot shows the 'Responses' tab, displaying a response with the following details:

- Response: Body: 1 payload
- Headers: A table with 2 columns: Name and Value.

Name	Value
"Access-Control-Allow-Origin"	"*"
"Access-Control-Allow-Methods"	"GET, PUT, POST, DELETE, HEAD, OPTIONS"
"Access-Control-Allow-Headers"	"*"
"Access-Control-Allow-Credentials"	"true"

Below the headers table, the Status code and Reason phrase fields are visible, both with a function icon (fx) next to them.

Problems Mule Debugger Console **Listener** x

There are no errors.

General

MIME Type

Redelivery

Responses

Advanced

Metadata

Notes

Help

Error Response

Body: `1 payload`

Headers: `fx`

Name	Value
"Access-Control-Allow-Origin"	"*"
"Access-Control-Allow-Methods"	"GET, PUT, POST, DELETE, HEAD, OPTIONS"
"Access-Control-Allow-Headers"	"*"
"Access-Control-Allow-Credentials"	"true"

Status code: `fx`

Reason phrase: `fx`

Problems Mule Debugger Console **Listener** x

There are no errors.

General

MIME Type

Redelivery

Responses

Advanced

Metadata

Notes

Help

General

Allowed methods: `POST`

Response streaming mode: `AUTO (Default)`

Streaming strategy `Repeatable file store stream (Default)`

In memory size: `512`

Buffer unit: `KB (Default)`

☐ Primary node only

Connection

Reconnection strategy `None`

Global Element Properties

×

HTTP Listener config

Configuration element for a HttpListener.

General

Notes

Help

Name: HTTP_Listener_config

Connection

General

TLS

Advanced

Connection

Protocol:

HTTPS

▼

Host:

All Interfaces [0.0.0.0] (default)

▼

Port:

45382

Read timeout:

30000

?

Test Connection...

OK

Cancel

Global Element Properties

×

HTTP Listener config
Configuration element for a HttpListener.

GeneralNotesHelp

Key Store Configuration

Type:JKS

Path:server.keystore

Alias:localhost

Key Password:•••••☐ Show password

Password:•••••☐ Show password

Algorithm:

Advanced

Enabled Protocols:

Enabled Cipher Suites:

?

Test Connection...

OK

Cancel

General

Base path:/api/v1

Listener interceptorsNone

☐ Reject invalid transfer encoding

Global Element Properties

×

Web Service Consumer Config

Default configuration

General

Advanced

Notes

Help

Name: Web_Service_Consumer_Config

Connection

General

Security

Transport

Advanced

Soap version:

fx

SOAP11 (Default)

▼

Mtom enabled:

False (Default)

▼

Encoding:

Connection

WSDL location:

fx

http://localhost:45011/api/v1/ws/schema.wsdl

...

Service:

fx

SchemaPortService

▼

🔗

Port:

fx

SchemaPortSoap11

▼

Address:

fx

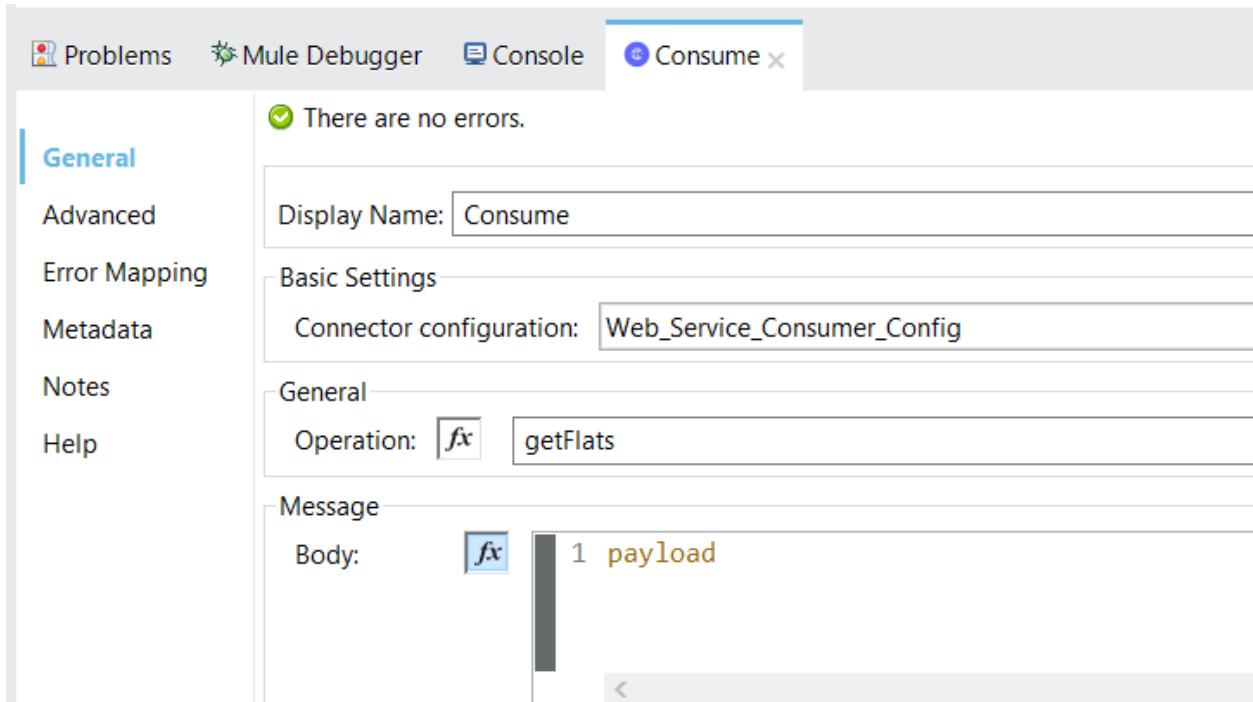
http://localhost:45011/api/v1/ws

▼

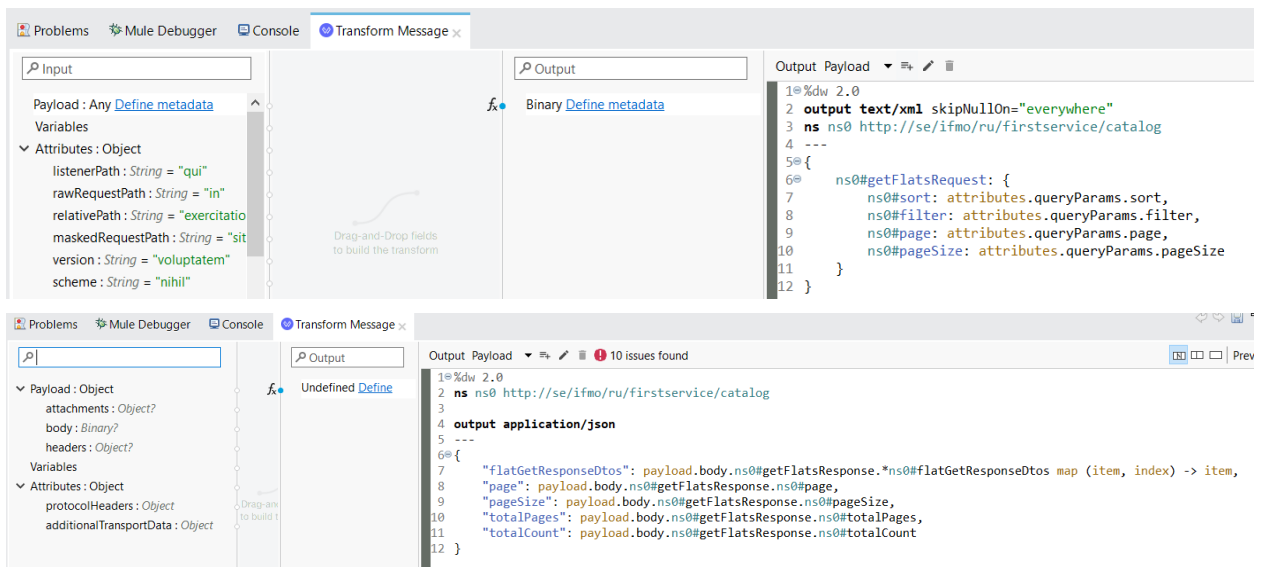
?

OK

Cancel



Примеры Ttransform Message (входящий и исходящий)



Пример Error Propagate (Transform Message)

The screenshot shows the MuleSoft Studio interface. At the top, an 'Error handling' section displays an 'On Error Propagate' configuration with 'type: ANY'. Below this, a 'Transform Message' processor is connected to a 'Logger' processor. The main workspace shows the 'Transform Message' configuration with an 'Input' field containing 'Payload: Unknown Define metadata', 'Variables', and 'Attributes: Unknown Define metadata'. The 'Output' field is set to 'String'. On the right, the 'Output Payload' tab shows a JSON response with error details:

```
1 %dw 2.0
2 output application/json
3
4 var detail = read(error.exception.cause.detail, "application/xml")
5 ---
6 {
7   "code": detail.detail.code,
8   "message": detail.detail.description,
9   "time": now()
10 }
```

Кастомные код ошибок:

The screenshot shows the Postman REST client interface. The request is a GET request to `https://localhost:45382/api/v1/catalog/flats/222`. The 'Body' tab is selected, showing a custom JSON error response:

```
1 {
2   "code": "404",
3   "message": "Flat with id 222 not found",
4   "time": "2023-01-24T19:42:34.282+03:00"
5 }
```

The status bar at the bottom indicates 'Status: 404 Not Found' and 'Time: 3.23 s'.

Problems Mule Debugger Console Listener x

General

MIME Type

Redelivery

Responses

Advanced

Metadata

Notes

Help

There are no errors.

Error Response

Body: `fx` 1 payload

Headers: `fx`

Name	Value
"Access-Control-Allow-Origin"	"*"
"Access-Control-Allow-Methods"	"GET, PUT, POST, DELETE, HEAD, OPTIONS"
"Access-Control-Allow-Headers"	"*"
"Access-Control-Allow-Credentials"	"true"

Status code: `fx` `#[read(error.exception.cause.detail, "application/xml").detail.code]`

Reason phrase: `fx`

Инструкция для развертывания:

1. Сначала нужно скачать себе Java 11 и установить, скачать все содержимое с гитхаба по ссылке - [ссылка](#)
2. Прошить Java 11 этими сертификатами при помощи команд:
 - a. `keytool -import -noprompt -trustcacerts -alias localhost -file server.crt -keystore $JAVA_HOME/lib/security/cacerts -storepass changeit`
 - b. `keytool -import -noprompt -trustcacerts -alias localhost2 -file localhost.crt -keystore $JAVA_HOME/lib/security/cacerts -storepass changeit`
3. Скачать Anypoint Studio.
 - a. Включаем VPN (без него не получится)
 - b. Регистрируемся и скачиваем Anypoint Studio (почту нужно будет указать при скачивании свою т.к. туда придет ссылка на скачивание)
 - c. Достаем из архива и запускаем.
 - i. Если у вас Windows, то советую проверить путь к Java или поставить на диск, на котором стоит у вас jdk. Иначе может кидаться ошибками.
 - d. Создаем проект (настраиваем workspace по собственному желанию).
 - e. Скачиваем с гитхаба - [ссылка](#) - soa4.xml и все содержимое из него переносит в ваш файл проекта {name}.xml
 - f. После чего пробуем запустить и Mule ESB должен уже заработать.
 - g. Если нужен будет Mule Standalone, тогда инструкцию ищите на гитхабе: [ссылка](#)
4. Устанавливаем WildFly 21.0.0.Final
 - a. Настраиваем конфиги у двух standalone (т.к. 2 сервиса, нужно 2 standalone)
 - i. У первого порт для http и https должен быть 45011 и 45012
 - ii. У второго порт для http и https должен быть 45391 и 45392
 - iii. Пример:

```
<socket-binding name="http" port="${jboss.http.port:45391}"/>
<socket-binding name="https" port="${jboss.https.port:45392}"/>
```
 - iv. В каждый standalone конфиг также закидываем .keystore, .truststore и .crt, которые мы скачали в пункте 1.
5. С помощью любой IDEA (советую IntelliJ IDEA от JetBrains) открыть проект с гитхаба: [ссылка](#)
 - a. Настраиваем JBOSS там на свои wildfly standalone и выбираем для деплоя у первого first-service.war, а для второго second-service.war
 - b. И на Java 11, которую мы настроили выше.
 - c. Запускаем их и все должно заработать.
6. После чего переходим на <https://se.ifmo.ru/~s284261/soa-front/#/catalog> и тестируем.

- а. P.S. в браузере придется ещё по ссылкам <https://localhost:45392>
<https://localhost:45012> и <https://localhost:45382> подтвердить, что мы согласны использовать не подтвержденные сертификаты.
7. Для деплоя на свой сервер инструкция для mule вот тут: [ссылка](#), а для .war файлов настроить точно также wildfly и закинуть в папки standalone wildfly'я в /deployments свои .war файлы. Ну после чего запустить все вместе. (не забыть главное пробросить порты 😊)