

**Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики**

Сервис-ориентированная архитектура

Лабораторная работа №4

Вариант 3002

Группа: Р33122

Студент:

Беляков Д.С

Преподаватель:

Усков И. В.

Санкт-Петербург

2021

Задание	2
Выполнение	2
Soap	2
Mule ESB	5
Спецификация	5
Схема в Anypoint Studio	9
Обработка CORS	15
Пример обработки запроса	15
Вывод	17

Задание

Переработать сервисы из лабораторной работы #3 следующим образом:

- Второй ("вызывающий") 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.

Код: https://github.com/kevinche75/soa_lab4_itmo_autumn_2021

Выполнение

Soap

```
@WebService(endpointInterface = "ru.itmo.soap.SoapDisciplineServiceI",
serviceName = "discipline_service")
```

```

public class SoapDisciplineService implements SoapDisciplineServiceI{

    private SecondServiceI service;

    public SoapDisciplineService() {
        service = RemoteBeanLookup.lookupRemoteStatelessBean();
    }

    @WebResult(partName="disciplines_result")
    @Override
    public DisciplineResult getDisciplines() throws ServiceFault {
        try {
            return service.getDisciplines();
        } catch (EJBException e){
            throw new ServiceFault(e.getMessage());
        } catch (Exception e){
            throw new ServiceFault("Unexpected error, try again!.500");
        }
    }

    @WebResult(partName="discipline")
    @Override
    public Discipline getDiscipline(String disciplineId) throws
ServiceFault {
        try {
            return service.getDiscipline(disciplineId);
        } catch (EJBException e){
            throw new ServiceFault(e.getMessage());
        } catch (Exception e){
            throw new ServiceFault("Unexpected error, try again!.500");
        }
    }

    @WebResult(partName="labworks_result")
    @Override
    public LabWorksResult getDisciplineLabWorks(String disciplineId)
throws ServiceFault {
        try {
            return service.getDisciplineLabWorks(disciplineId);
        } catch (EJBException e){
            throw new ServiceFault(e.getMessage());
        } catch (Exception e){
            throw new ServiceFault("Unexpected error, try again!.500");
        }
    }

    @WebResult(partName="serverResponse")
    @Override

```

```

    public ServerResponse createDiscipline(String name) throws
ServiceFault {
        System.out.println("strDiscipline: " + name);
        try {
            return service.createDiscipline(name);
        } catch (EJBException e){
            throw new ServiceFault(e.getMessage());
        } catch (Exception e){
            throw new ServiceFault("Unexpected error, try again!.500");
        }
    }

    @WebResult(partName="serverResponse")
    @Override
    public ServerResponse deleteLabWorkFromDiscipline(String
disciplineId, String labWorkId) throws ServiceFault {
        try {
            return service.removeLabWorkFromDiscipline(disciplineId,
labWorkId);
        } catch (EJBException e){
            throw new ServiceFault(e.getMessage());
        } catch (Exception e){
            throw new ServiceFault("Unexpected error, try again!.500");
        }
    }

    @WebResult(partName="serverResponse")
    @Override
    public ServerResponse addLabWorkToDiscipline(String disciplineId,
String labWorkId) throws ServiceFault {
        try {
            return service.addLabWorkToDiscipline(disciplineId,
labWorkId);
        } catch (EJBException e){
            throw new ServiceFault(e.getMessage());
        } catch (Exception e){
            throw new ServiceFault("Unexpected error, try again!.500");
        }
    }
}

```

```

@WebService
public interface SoapDisciplineServiceI {

```

```

    @WebMethod
    DisciplineResult getDisciplines() throws ServiceFault;

```

```

    @WebMethod

```

```

    Discipline getDiscipline(String disciplineId) throws ServiceFault;

    @WebMethod
    LabWorksResult getDisciplineLabWorks(String disciplineId) throws
ServiceFault;

    @WebMethod
    ServerResponse createDiscipline(String strDiscipline) throws
ServiceFault;

    @WebMethod
    ServerResponse deleteLabWorkFromDiscipline(String disciplineId,
String labWorkId) throws ServiceFault;

    @WebMethod
    ServerResponse addLabWorkToDiscipline(String disciplineId, String
labWorkId) throws ServiceFault;
}

```

Mule ESB

Спецификация

```

#%RAML 1.0
title: soa_lab4
mediaType:
- application/xml
protocols:
- HTTPS
baseUri: /lab2/bars/disciplines
version: "2.0"
/disciplines/:
  get:
    displayName: Get Disciplines
    responses:
      200:
        body:
          type: object
      400:
        body:
          type: object
      500:
        body:
          type: object
    securedBy: []
  options:
    displayName: Options
    responses:

```

```
        200: {}
    securedBy: []
  post:
    displayName: Create Discipline
    responses:
      200:
        body:
          type: object
      400:
        body:
          type: object
      500:
        body:
          type: object
    securedBy: []
    body:
      type: discipline
/disciplines/{id}:
  get:
    displayName: Get Discipline by Id
    responses:
      200:
        body:
          type: object
      400:
        body:
          type: object
      500:
        body:
          type: object
    securedBy: []
  options:
    displayName: Options
    responses:
      200: {}
    securedBy: []
  uriParameters:
    id:
      example: test
      type: string
/disciplines/{id}/labworks:
  get:
    displayName: Get Discipline's LabWorks
    responses:
      200:
        body:
          type: object
      400:
```

```

        body:
            type: object
    500:
        body:
            type: object
    securedBy: []
options:
    displayName: Options
    responses:
        200: {}
    securedBy: []
uriParameters:
    id:
        example: test
        type: string
/disciplines/{discipline-id}/labwork/{labwork-id}/remove:
delete:
    displayName: Delete LabWork From Discipline
    responses:
        200:
            body:
                type: object
        400:
            body:
                type: object
        500:
            body:
                type: object
    securedBy: []
options:
    displayName: Options
    responses:
        200: {}
    securedBy: []
uriParameters:
    discipline-id:
        example: test
        type: string
    labwork-id:
        example: test
        type: string
/disciplines/{id}/{labwork-id}:
post:
    displayName: Add LabWork to Discipline
    responses:
        200:
            body:
                type: object

```

```

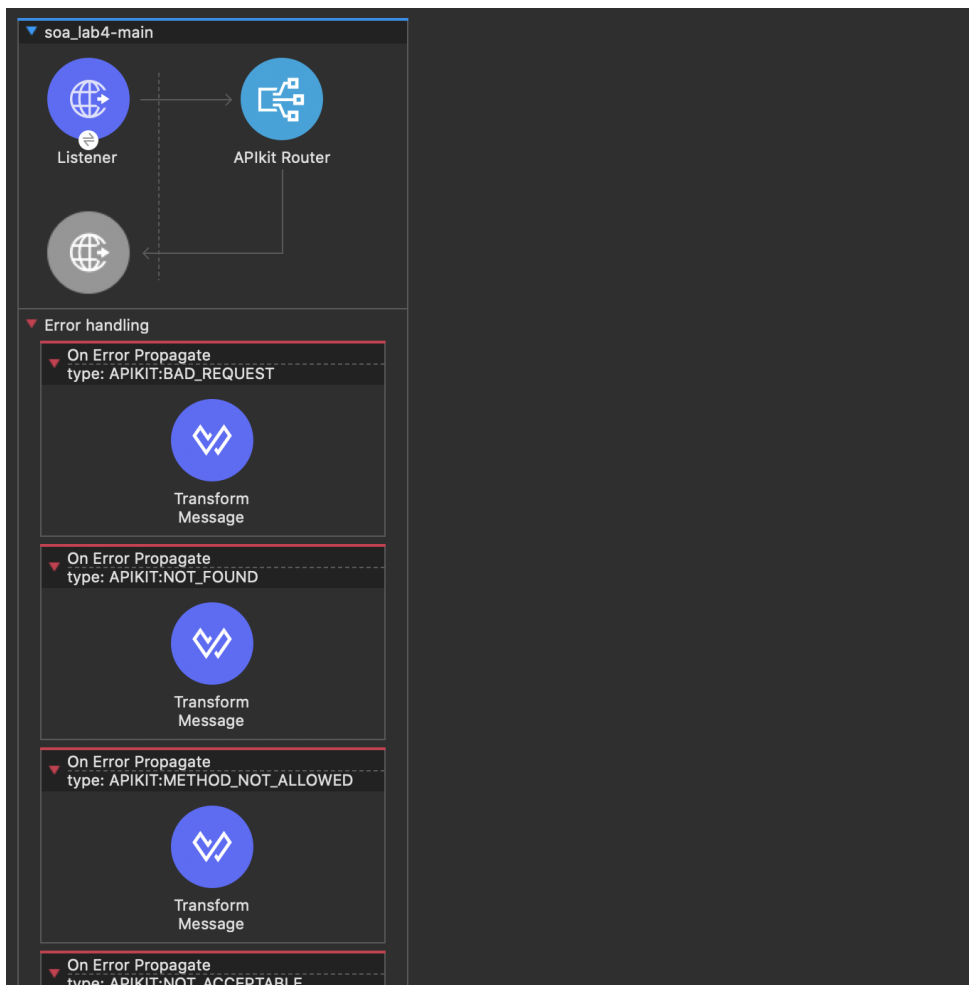
        400:
            body:
                type: object
        500:
            body:
                type: object
    securedBy: []
options:
    displayName: Options
    responses:
        200: {}
    securedBy: []
uriParameters:
    id:
        example: test
        type: string
    labwork-id:
        example: test
        type: string
/labworks/{labwork-id}/difficulty/increase/{steps-count}:
    put:
        displayName: Increase LabWork Difficulty
        responses:
            200:
                body:
                    type: object
            400:
                body:
                    type: object
            500:
                body:
                    type: object
        securedBy: []
options:
    displayName: Options
    responses:
        200: {}
    securedBy: []
uriParameters:
    labwork-id:
        example:
            test
        type: string
    steps-count:
        example:
            test
        type: string
types:

```



```
discipline:
  properties:
    name:
      example:
        test
      type:
        string
```

Схема в Anypoint Studio



▼ put:\labworks\(\labwork-id)\difficulty\increase\(\steps-count):soa_lab4-config

Source



Logger



Transform Message
Copy_of_Transform
Message



Consume



Transform
Message

▼ Error handling

▼ Copy_of_On Error Continue



Transform Message
Copy_of_Transform
Message



Set Variable
Copy_of_Set
Variable

▼ delete:\disciplines\(\discipline-id)\labwork\(\labwork-id)\remove:soa_lab4-config

Source



Logger



Transform Message
Copy_of_Transform
Message



Consume



Transform Message
Copy_of_Transform
Message

▼ Error handling

▼ Copy_of_On Error Continue



Transform Message
Copy_of_Transform
Message



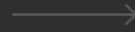
Set Variable
Copy_of_Set
Variable

▼ options:\disciplines\soa_lab4-config

Source



Logger



Set Variable
Copy_of_Copy_of_S
et Variable

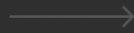
► Error handling

▼ options:\disciplines\(\discipline-id)\labwork\(\labwork-id)\remove:soa_lab4-config

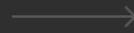
Source



Transform
Message



Logger



Set Variable
Copy_of_Copy_of_S
et Variable

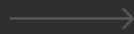
► Error handling

▼ options:\disciplines\(\id):soa_lab4-config

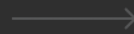
Source



Transform
Message



Logger



Set Variable
Copy_of_Copy_of_S
et Variable

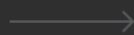
► Error handling

▼ options:\disciplines\(\id)\labworks:soa_lab4-config

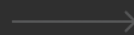
Source



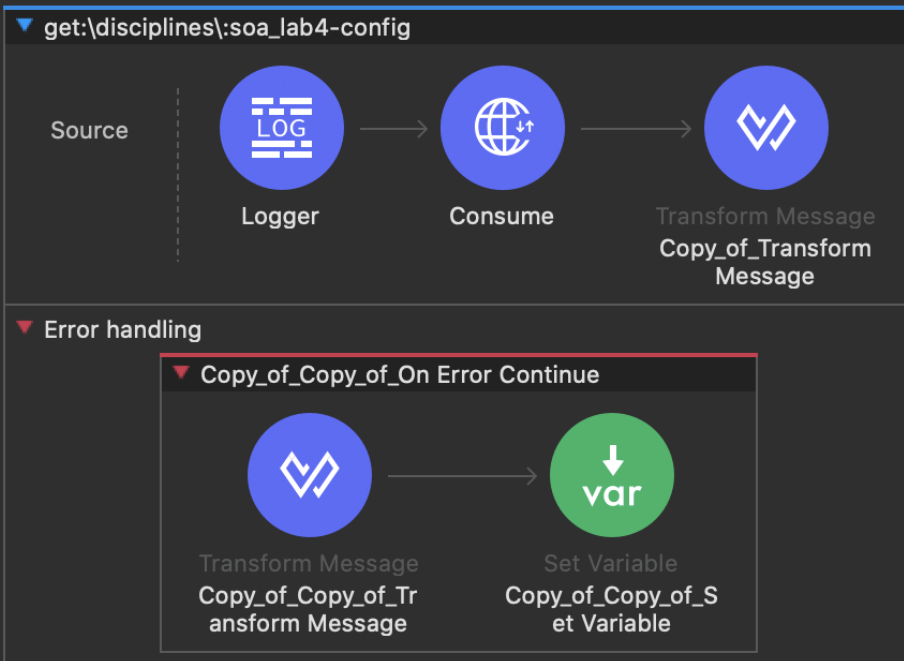
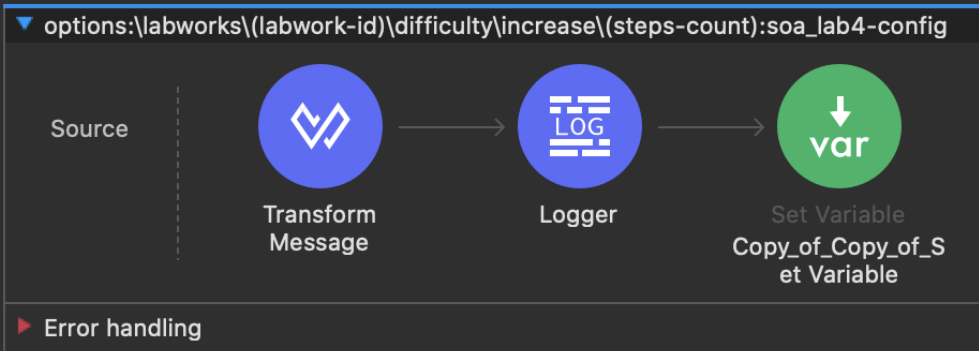
Transform
Message

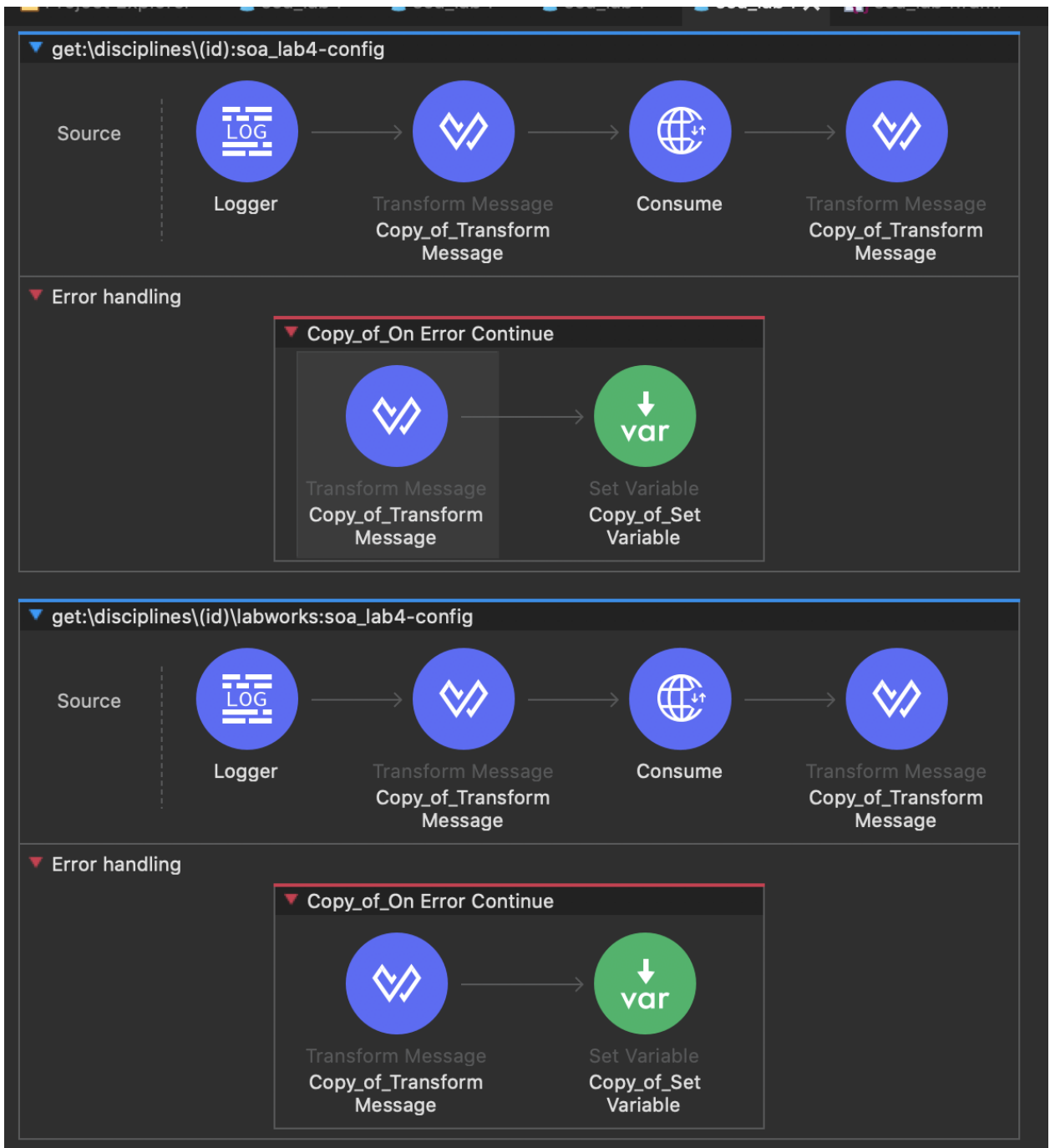


Logger

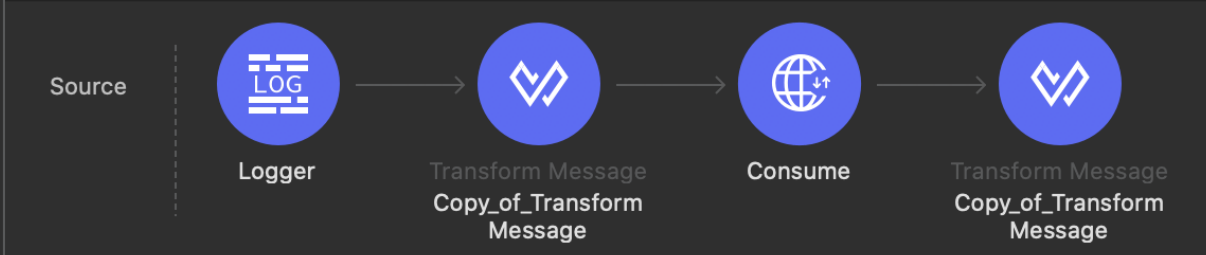


Set Variable
Copy_of_Copy_of_S



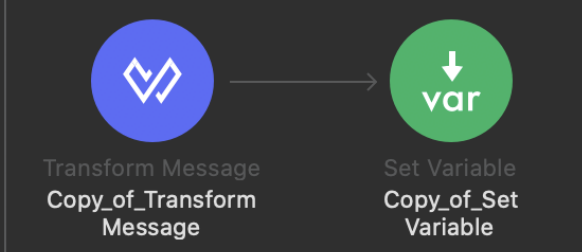


▼ post:\disciplines\application\xml:soa_lab4-config

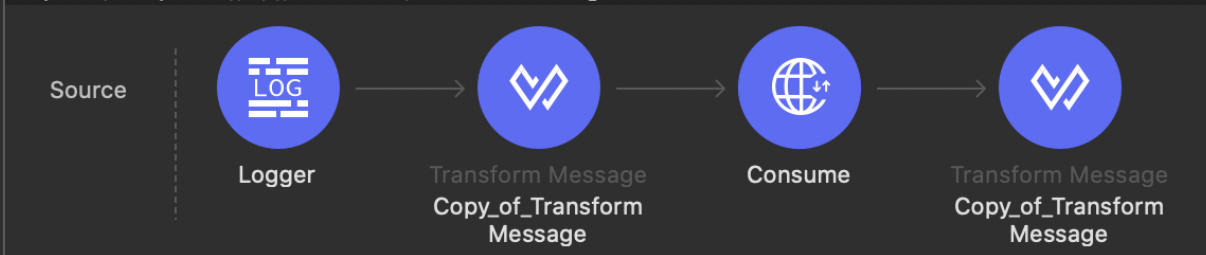


▼ Error handling

▼ Copy_of_On Error Continue

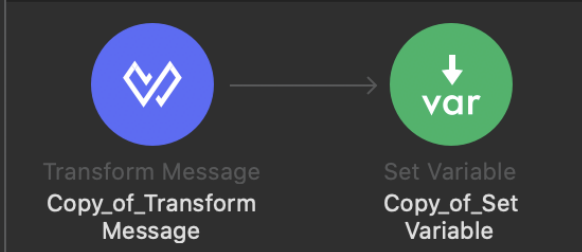


▼ post:\disciplines\id\labwork-id:soa_lab4-config



▼ Error handling

▼ Copy_of_On Error Continue



Обработка CORS

soa_lab4-main

Listener → APIKit Router

Message Flow | Global Elements | Configuration XML

Console | Problems | APIKit Consoles | **Listener** X

General | MIME Type | Redelivery | **Responses** | Advanced | Metadata | Notes | Help

There are no errors.

Response

Body: `1 payload`

Headers: `4 "Access-Control-Allow-Origin" : "*",
5 "Access-Control-Allow-Methods" : "GET, PUT, POST",
6 "Access-Control-Allow-Headers" : "Content-Type",
7 "Access-Control-Allow-Credentials" : "true",
8`

Status code: `#[vars.httpStatus default 200]`

Reason phrase:

Пример обработки запроса

delete:(discipline-id)\labwork:(labwork-id)\remove:soa_lab4-config

Source → Logger → Transform Message Copy of Transform Message → Consume → Transform Message Copy of Transform Message

Error handling: Copy of On Error Continue → Transform Message Copy of Transform Message → Set Variable Copy of Set Variable

Message Flow | Global Elements | Configuration XML

Console | Problems | APIKit Consoles | **Copy of Transform Message** X

Input

Payload: Any Define metadata

Variables

Attributes: Object

clientCertificate: Object?

type: String?

encoded: Binary?

headers: Object

Context

Output

Xml: <deleteLabWorkFromDiscipl...>

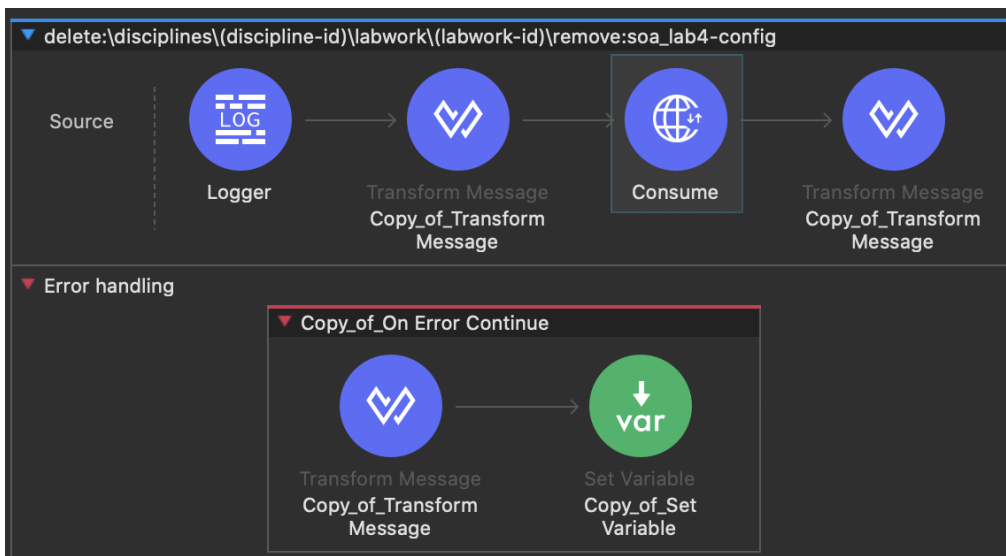
deleteLabWorkFromDiscipl

arg0: String?

arg1: String?

Output Payload

```
1 <?xml version="1.0"?>
2 ns ns0 http://soap.itmo.ru/
3 output application/xml
4 <ns0:deleteLabWorkFromDiscipline>
5 {
6   ns0:deleteLabWorkFromDiscipline: {
7     arg0: attributes.uriParams.'discipline-id',
8     arg1: attributes.uriParams.'labwork-id'
9   }
10 }
```



Message Flow Global Elements Configuration XML

Console Problems APIKit Consoles **Consume X**

General

Advanced

Error Mapping

Metadata

Notes

Help

There are no errors.

Basic Settings

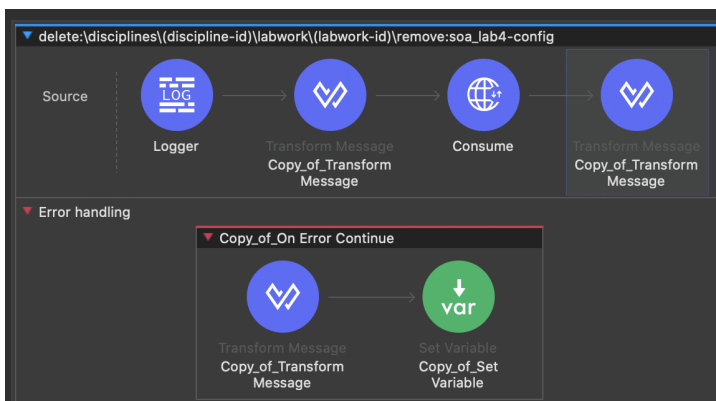
Connector configuration: Web_Service_Consumer_Config

General

Operation: `fx` deleteLabWorkFromDiscipline

Message

Body: `fx` 1 payload



Message Flow Global Elements Configuration XML

Console Problems APIKit Consoles **Copy_of_Transform Message X**

Q Input

Q Output

Output Payload Preview

```
1 %dw 2.0
2 ns ns0 http://soap.itmo.ru/
3 output application/xml
4 ---
5 serverResponse:
6   payload.body.ns0#deleteLabWorkFromDisciplineResponse.ret
```

Payload : Object

body : Object?

deleteLabWorkFromDis

Variables

discipline-id : String

labwork-id : String

Attributes : Object

Вывод

В данной лабораторной работе я реализовал soap сервис, познакомился с программой Mule ESB, с помощью неё реализовал взаимодействие 2 сервисов, при этом столкнулся с большими сложностями:

1. Если неправильно указать метод в спецификации, то приходится создавать проект заново, однако можно сделать синхронизацию, при этом необходимо выполнить достаточно много действий
2. Обработка ошибок довольно неочевидна, при этом пришлось добавить код ошибки в сообщение и обрабатывать отдельно
3. Mule ESB присутствует достаточно много багов: периодически не открывается определенный блок (например, любой transform message у меня перестал открываться в какой-то момент), drag-and-drop иногда не работает, поэтому приходилось писать код вручную, ну и самое замечательное - пустая строка передаётся в аргументы функции, хотя при этом payload заполнен правильно