

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
науки и высшего образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)

Кафедра информационных систем и программной инженерии

Лабораторная работа
по дисциплине
" Интеграция кроссплатформенных программных систем"
Тема: "Шаблоны интеграции с Apache Camel"

Выполнил:
ст. гр. ИСТм-121
Хлызова В.Г.

Принял:
Спирин И.В.

Владимир, 2022 г.

ЦЕЛЬ РАБОТЫ

Разработка интеграционного взаимодействия используя Apache Camel.

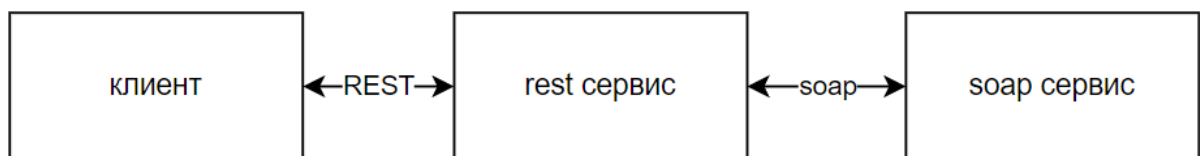
ВЫПОЛНЕНИЕ РАБОТЫ

Ссылка на документацию: <https://www.baeldung.com/camel-integration-patterns>

Разработанное приложение:

[https://github.com/arranay/Integration of cross platform software systems/tree/main/lab9/camel-soap-rest-master](https://github.com/arranay/Integration%20of%20cross%20platform%20software%20systems/tree/main/lab9/camel-soap-rest-master)

В данной работе было разработано два сервиса (soap и rest), которые взаимодействуют между собой используя фреймворк Apache Camel и используют SOAP в качестве протокола передачи данных. В качестве клиента в данной лабораторной работе будет выступать Postman.



SOAP сервис

RouteBuilder класс:

```
package com.lab9.camel.soap;

import com.lab9.camel.model.TrackService;
import org.apache.camel.builder.RouteBuilder;
import org.springframework.stereotype.Component;

@Component
public class SoapRouter extends RouteBuilder {

    private String url = "cxf:track?serviceClass=" +
TrackService.class.getName();

    @Override
    public void configure() throws Exception {
        from(url).routeId("track-soap")
            .to("log:input")
            .recipientList(simple("direct:${header.operationName}"));
    }
}
```

```

        from("direct:getTrack")
            .bean(GetTrackProcessor.class)
            .to("log:output");

        from("direct:addTrack")
            .bean(AddTrackProcessor.class)
            .to("log:output");
    }
}

```

GetTrackProcessor:

```

public class GetTrackProcessor implements Processor {

    @Autowired
    TrackServiceImplementation trackService;

    private final Logger log = LoggerFactory.getLogger(this.getClass());

    @Override
    public void process(Exchange exchange) throws Exception {
        GetTrackResponse trackResponse = new GetTrackResponse();
        trackResponse.setTracks(trackService.getTrack());
        exchange.getOut().setBody(trackResponse);
    }
}

```

AddTrackProcessor:

```

public class AddTrackProcessor implements Processor {

    @Autowired
    TrackServiceImplementation trackService;

    private final Logger log = LoggerFactory.getLogger(this.getClass());

    @Override
    public void process(Exchange exchange) throws Exception {
        Track track = exchange.getIn().getBody(Track.class);
        trackService.addTrack(track);

        exchange.getOut().setBody(track);
    }
}

```

REST сервис

TrackManager сервис:

```

import com.lab9.camel.model.GetTrackResponse;
import com.lab9.camel.model.Track;

```

```

import com.lab9.camel.model.TrackService;
import org.apache.cxf.frontend.ClientProxyFactoryBean;
import org.apache.cxf.interceptor.LoggingInInterceptor;
import org.apache.cxf.interceptor.LoggingOutInterceptor;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Service;

import javax.annotation.PostConstruct;

@Service
public class TrackManager {

    @Value("${user.soap.address}")
    private String address;

    private TrackService trackService;

    @PostConstruct
    private void setUp() {
        ClientProxyFactoryBean factoryBean = new ClientProxyFactoryBean();
        factoryBean.setAddress(address);
        factoryBean.setServiceClass(TrackService.class);
        factoryBean.getInInterceptors().add(new LoggingInInterceptor());
        factoryBean.getOutInterceptors().add(new LoggingOutInterceptor());

        trackService = factoryBean.create(TrackService.class);
    }

    public GetTrackResponse getTrack() {
        return trackService.getTrack();
    }

    public Track addTrack(Track track) {
        return trackService.addTrack(track);
    }
}

```

RouteBuilder класс:

```

import com.lab9.camel.model.Track;
import org.apache.camel.builder.RouteBuilder;
import org.apache.camel.model.rest.RestBindingMode;
import org.springframework.stereotype.Component;

import java.util.ArrayList;

@Component
public class RestRouter extends RouteBuilder {
    @Override
    public void configure() throws Exception {
        restConfiguration()
            .apiProperty("api.title", "Camel REST API")
            .apiProperty("api.version", "1.0")
            .apiProperty("cors", "true")
            .bindingMode(RestBindingMode.json);

        rest("/track")
            .produces("application/json")
            .consumes("application/json")
            .get("/").route().routeId("get-track-route")
            .outputType(ArrayList.class)
            .bean(GetTrackProcessor.class);
    }
}

```

```

        rest("/track")
            .produces("application/json")
            .consumes("application/json")
            .post("/").route().routeId("post-track-route")
            .inputType(Object.class)
            .outputType(Track.class)
            .bean(AddTrackProcessor.class);
    }
}

```

GetTrackProcessor:

```

import com.lab9.camel.model.GetTrackResponse;
import org.apache.camel.Exchange;
import org.apache.camel.Processor;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;

public class GetTrackProcessor implements Processor {

    private final Logger log = LoggerFactory.getLogger(this.getClass());

    @Autowired
    private TrackManager trackManager;

    @Override
    public void process(Exchange exchange) throws Exception {
        GetTrackResponse track = trackManager.getTrack();
        exchange.getOut().setBody(track);
    }
}

```

AddTrackProcessor:

```

import com.fasterxml.jackson.databind.ObjectMapper;
import com.lab9.camel.model.Track;
import org.apache.camel.Exchange;
import org.apache.camel.Processor;
import org.apache.camel.http.common.HttpMessage;
import org.springframework.beans.factory.annotation.Autowired;

public class AddTrackProcessor implements Processor {

    @Autowired
    private TrackManager trackManager;

    @Override
    public void process(Exchange exchange) throws Exception {
        Object body = exchange.getIn(HttpMessage.class).getBody();

        ObjectMapper mapper = new ObjectMapper();
        Track track = mapper.convertValue(body, Track.class);
        Track created_track = trackManager.addTrack(track);

        exchange.getOut().setBody(created_track);
    }
}

```

Выполнение

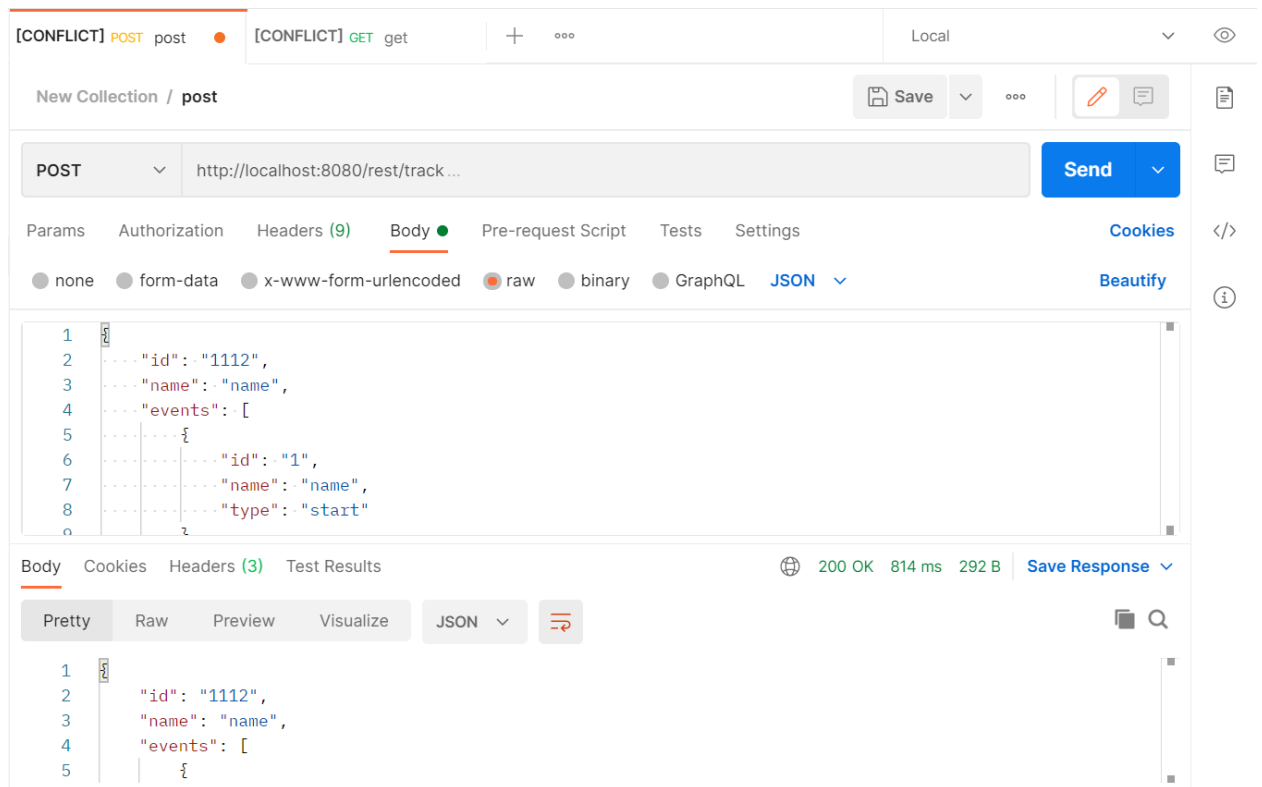


Рисунок 1 - Выполнение POST запроса.

Логирование:

```

ID: 1
Address: http://localhost:8094/webservices/track?wsdl
Encoding: UTF-8
Http-Method: POST
Content-Type: text/xml
Headers: {Accept=[*/*], SOAPAction=[""]}
Payload: <soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"><soap:Body><ns1:addTrack
xmlns:ns1="http://model.camel.lab9.com/"><ns2:arg0
xmlns:ns2="http://model.camel.lab9.com/"><actions><id>2</id><operation>start<
/operation><time>12</time></actions><events><id>1</id><name>name</name><type>
start</type></events><forks><condition>condition</condition><id>3</id></forks
><id>1112</id><name>name</name></ns2:arg0></ns1:addTrack></soap:Body></soap:E
nvelope>
-----
2023-01-09 07:38:44 [http-nio-8080-exec-1] INFO
o.a.c.s.T.T.TrackServicePortType line[274] - Inbound Message
-----
ID: 1
Response-Code: 200
Encoding: UTF-8
Content-Type: text/xml; charset=UTF-8
Headers: {Content-Length=[482], content-type=[text/xml; charset=UTF-8],
Date=[Mon, 09 Jan 2023 04:38:44 GMT]}
Payload: <soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"><soap:Body><ns1:addTrackResponse
xmlns:ns1="http://model.camel.lab9.com/"><ns2:return
xmlns:ns2="http://model.camel.lab9.com/"><actions><id>2</id><operation>start<
/operation><time>12</time></actions><events><id>1</id><name>name</name><type>
start</type></events><forks><condition>condition</condition><id>3</id></forks

```

```
><id>1112</id><name>name</name></ns2:return></ns1:addTrackResponse></soap:Body></soap:Envelope>
```

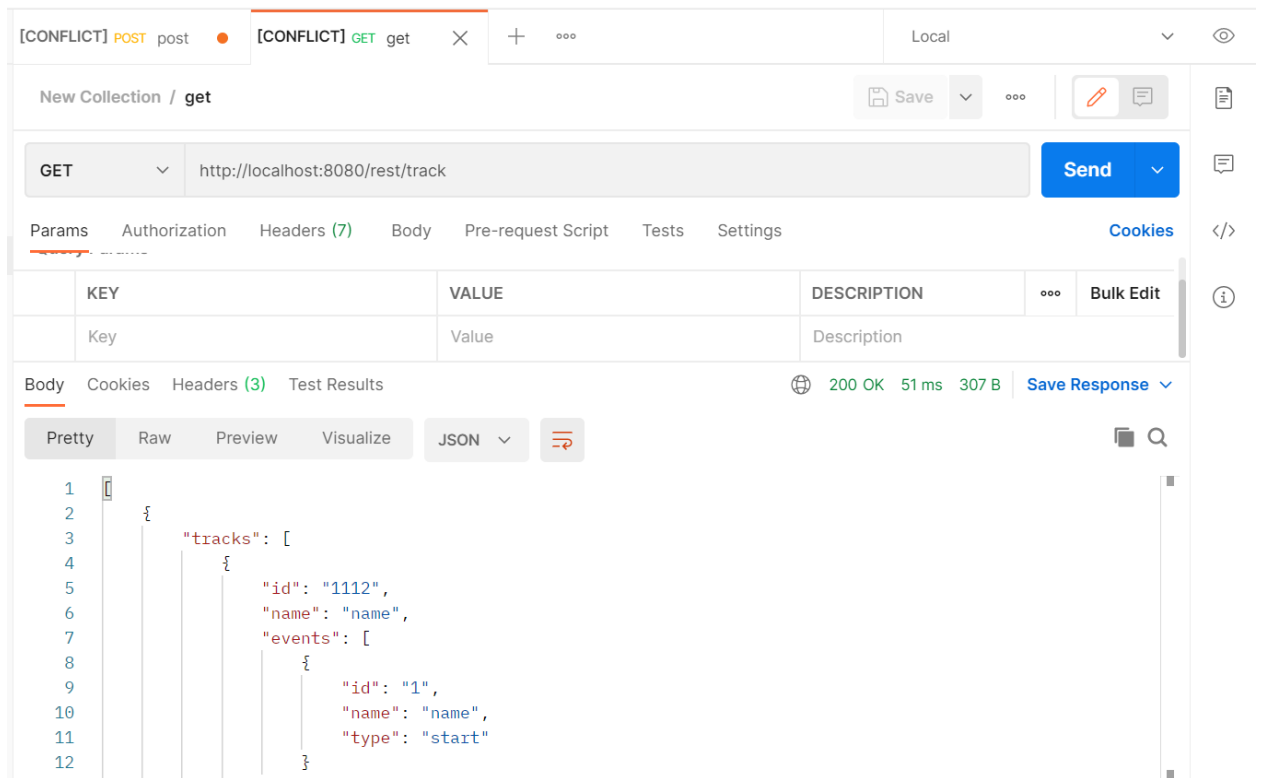


Рисунок 2 - Выполнение GET запроса.

Логирование:

```
ID: 2
Address: http://localhost:8094/webservices/track?wsdl
Encoding: UTF-8
Http-Method: POST
Content-Type: text/xml
Headers: {Accept=[*/*], SOAPAction=[""]}
Payload: <soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"><soap:Body><ns1:getTrack
xmlns:ns1="http://model.camel.lab9.com/"></soap:Body></soap:Envelope>
-----
2023-01-09 07:39:09 [http-nio-8080-exec-3] INFO
o.a.c.s.T.T.TrackServicePortType line[274] - Inbound Message
-----
ID: 2
Response-Code: 200
Encoding: UTF-8
Content-Type: text/xml; charset=UTF-8
Headers: {Content-Length=[499], content-type=[text/xml; charset=UTF-8],
Date=[Mon, 09 Jan 2023 04:39:09 GMT]}
Payload: <soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"><soap:Body><ns1:getTrackResponse
xmlns:ns1="http://model.camel.lab9.com/"><ns2:return
xmlns:ns2="http://model.camel.lab9.com/"><tracks><actions><id>2</id><operation>start</operation><time>12</time></actions><events><id>1</id><name>name</name><type>start</type></events><forks><condition><id>3</id></forks><id>1112</id><name>name</name></tracks></ns2:return></ns1:getTrackResponse></soap:Body></soap:Envelope>
```

ВЫВОДЫ

В процессе выполнения работы было разработано интеграционное взаимодействие используя Apache Camel.