

Pocket – Treinamento Apache Camel + Spring Boot

Ministrante: Thiago Hernandes de Souza – thiago.hernandes.souza@everis.com



M ó d u l o 01



Pocket – Treinamento Apache Camel + Spring Boot

Ministrante: Thiago Hernandes de Souza – thiago.hernandes.souza@everis.com



O que veremos:

1 – Funções de conversão	página 3
2 – Chamada de Transação no Mainframe com Qt	página 4
3 – Configurações padrão de header	página 4
4 – Bearer Token Exemplo	página 5
5 – Envio/retorno de objeto de uma transação Qt	página 5
5.1 - Envio	página 5
5.2 - Retorno	página 6
5.3 – Bloco: envio e retorno	página 6
6 – Connector IMS	página 7
6.1 Entrada DTO	página 7
6.2 Saída DTO	página 8
7 – Qt Route Exemplo	página 9
7.1 @AreaContrato – entrada	página 9
8 – Confluence – Itaú	página 9

Pocket – Treinamento Apache Camel + Spring Boot

Ministrante: Thiago Hernandes de Souza – thiago.hernandes.souza@everis.com



an NTT DATA Company

1 – Funções de conversão

```
/* Formata qualquer lista de objeto para JSON
 * @author Thiago Hernandes de Souza - Everis/Itaú
 * @since 28-05-2019
 * */
public String formatListJoJson(List<?> lista) {
    return new Gson().toJson(lista);
}

/*
 * Formata uma String de dados XML para JSON
 * @author Thiago Hernandes de Souza - Everis/Itaú
 * @since 31-05-2019
 * */
public String XMLToJSON(String dadosXML) {
    JSONObject parserReturn = XML.toJSONObject(dadosXML);
    return parserReturn.toString();
}

/*
 * Formata uma String de dados XML para JSON
 * @author Thiago Hernandes de Souza - Everis/Itaú
 * @since 31-05-2019
 * */
public String XMLToString(Object xml) throws JAXBException {
    JAXBContext jc = JAXBContext.newInstance(Object.class);
    Marshaller marshaller = jc.createMarshaller();
    marshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true);
    StringWriter sw = new StringWriter();
    marshaller.marshal(xml, sw);
    return sw.toString();
}
```

Pocket – Treinamento Apache Camel + Spring Boot

Ministrante: Thiago Hernandes de Souza – thiago.hernandes.souza@everis.com



an NTT DATA Company

2 – Chamada de Transação no Mainframe com Qt

```
public SimulacaoSaidaSE00682X obterDadosSimulacao(  
    final SimulacaoEntradaSE00682X dados,  
    final Class<SimulacaoSaidaSE00682X> responseType) throws ConnectorException {  
  
    final HttpHeaders headers = headersFactory.getHeaders();  
    headers.set("CodigoTipoTerminal", connectionProperties.getTerminal());  
  
    final ConnectorQt conn = new ConnectorQt(this.connectionProperties.getHostQt(), restTemplate);  
    final Object response = conn.sendQt(dados, responseType, headers);  
  
    return (SimulacaoSaidaSE00682X)response;  
}
```

3 – Configurações padrão de header

```
public void configure(){  
    from(TO_STS)  
        .routeId(TO_STS_ID)  
        .errorHandler(defaultErrorHandler().maximumRedeliveries(3))  
        .doTry()  
            .setHeader(Exchange.HTTP_METHOD, simple("GET"))//mudar metodo para o que for preciso  
            .setHeader(Exchange.CONTENT_TYPE, simple("application/x-www-form-urlencoded"))  
            .setHeader(Exchange.HTTP_QUERY, simple(""))  
            .setHeader("x-itaui-correlationID", simple("${xItaiCorrelationid}"))  
            .setHeader("x-itaui-flowID", simple("${xItaiFlowid}"))  
            .setHeader("x-itaui-apikey", simple("${clientId}"))  
            .log("${headers}")  
            .setBody(simple("grant_type=client_credentials&client_id=${clientId}&client_secret=${clientSecret}"))  
            .toD("https-api:${oauth.server.endpoint.url}/api/oauth/token?bridgeEndpoint=true")  
            .choice()  
                .when(new CustomPredicate().getStsPredicate())  
                    .log("INFO: Recebendo o Retorno do STS")  
                    .unmarshal().json(JsonLibrary.Jackson, OfertaSaidaSts.class)  
                    .bean(OfertaController.class, "armazenarSaidaSts")  
  
                    .log("INFO: Criando Entrada para a API de Consulta de Solicitação")  
                    .bean(OfertaController.class, "criarEntradaT900334X")  
  
                    .log("INFO: Chamando API Ofertas")  
                    .toD(OfertaT900334XRoute.TO_TRANSACAO_T900334X)  
                    .log("INFO: Chamada a API Ofertas executada com sucesso")  
                .otherwise()  
                    .log("INFO: Validação do retorno do STS - Falhou")  
                    .bean(OfertaEntradaPredicate.class, "retornarErroSts");//TODO: implementar metodo  
            .endDoTry()  
            .doCatch(Exception.class)  
                .log("INFO: Foi capturada uma exceção - STS")  
                .bean(GenericException.class, "tratarException")  
            .end();  
}
```

Pocket – Treinamento Apache Camel + Spring Boot

Ministrante: Thiago Hernandes de Souza – thiago.hernandes.souza@everis.com



an NTT DATA Company

4 – Bearer Token Exemplo

```
@Override
public void configure() throws Exception {
    from(TO_TRANSACAO_T900334X)
        .routeId(TO_TRANSACAO_T900334X_ID)
        .doTry()
        .setHeader(Exchange.HTTP_METHOD, simple("GET"))
        .setHeader(Exchange.CONTENT_TYPE, simple("application/json"))
        .setHeader("x-itaui-apikey", simple("${clientId}"))
        .setHeader("x-itaui-flowID", simple("${xItaiFlowid}"))
        .setHeader("x-itaui-correlationID", simple("${xItaiCorrelationid}"))
        .setHeader("CodigoTipoTerminal", simple("71DS1"))
        .process(ex->{
            OfertaSaidaSts sts =(OfertaSaidaSts) ex.getProperty("SaidaSts");
            ex.getIn().setHeader("Authorization", "Bearer "+sts.getAccessToken());
        })
}
```

5 – Envio/retorno de objeto de uma transação Qt

5.1 - Envio

```
/**
 * Classe realiza o encapsulamento dos dados para envio ao IMS
 * @author samirad
 * @since 2019-03-29
 */
@JsonPropertyOrder({ "transacao", "trancode", "usuario", "token" })
public class DataConnectorQT {

    @JsonProperty("transacao")
    private String transacao;

    @JsonProperty("trancode")
    private String trancode;

    public String getTransacao() {
        return transacao;
    }

    public void setTransacao(String transacao) {
        this.transacao = transacao;
    }

    public String getTrancode() {
        return trancode;
    }

    public void setTrancode(String trancode) {
        this.trancode = trancode;
    }
}
```

Pocket – Treinamento Apache Camel + Spring Boot

Ministrante: Thiago Hernandes de Souza – thiago.hernandes.souza@everis.com



an NTT DATA Company

5.2 - Retorno

```
public class RetornoT900334X {  
  
    @JsonProperty(value = "Result")  
    private String result;  
  
    @JsonProperty(value = "Server")  
    private String server;  
  
    @JsonProperty(value = "Terminal")  
    private String terminal;  
  
    public String getResult() {  
        return result;  
    }  
  
    public void setResult(String result) {  
        this.result = result;  
    }  
  
    public String getServer() {  
        return server;  
    }  
  
    public void setServer(String server) {  
        this.server = server;  
    }  
  
    public String getTerminal() {  
        return terminal;  
    }  
}
```

5.3 – Bloco: envio e retorno

```
.marshal().json(JsonLibrary.Jackson, DataConnectorQT.class)  
.log("Body de envio ${in.body}")  
.toD("https-api:{{api.connector.endpoint.url.qt}}?bridgeEndpoint=true")  
.log("Status da volta ${header.CamelHttpResponseCode}")  
.unmarshal().json(JsonLibrary.Jackson, RetornoT900334X.class)
```

Pocket – Treinamento Apache Camel + Spring Boot

Ministrante: Thiago Hernandes de Souza – thiago.hernandes.souza@everis.com



an NTT DATA Company

6 – Connector IMS

```
@Component
public class AppIMSRoute extends RouteBuilder {

    public static final String ROUTE_CONECTOR_ID = "rota_exemplo_conector";
    public static final String ROUTE_ENDPOINT_CONECTOR = "direct:rota_exemplo_conector";

    @Override
    public void configure() throws Exception {
        from(ROUTE_ENDPOINT_CONECTOR).routeId(ROUTE_CONECTOR_ID)
            .convertBodyTo(String.class)
            .unmarshal()
                .json(JsonLibrary.Jackson, ImsEntradaDTO.class)
                .to("ims://UD00132?{{opcionais}}")
                .id("ims").marshal().json(JsonLibrary.Jackson, ImsSaidaDTO.class)
            .endRest();
    }
}
```

6.1 Entrada DTO

```
public class ImsEntradaDTO {

    @JsonProperty(value="id_operacao")
    private String idOperacao;

    public String getIdOperacao() {
        return idOperacao;
    }

    public void setIdOperacao(String idOperacao) {
        this.idOperacao = idOperacao;
    }
}
```

Pocket – Treinamento Apache Camel + Spring Boot

Ministrante: Thiago Hernandes de Souza – thiago.hernandes.souza@everis.com



an NTT DATA Company

6.2 Saída DTO

```
public class ImsSaidaDTO {

    @JsonProperty(value = "status_operacao")
    @TrancodeField(length = 26, typeConvert = TypeConvert.SPACE_TO_RIGHT)
    private String statusOperacao;

    @JsonProperty(value = "nome_responsavel")
    @TrancodeField(length = 40, typeConvert = TypeConvert.SPACE_TO_RIGHT)
    private String nomeResponsavel;

    @JsonProperty(value = "email_responsavel")
    @TrancodeField(length = 46, typeConvert = TypeConvert.SPACE_TO_RIGHT)
    private String emailResponsavel;

    @JsonProperty(value = "setor_responsavel")
    @TrancodeField(length = 21, typeConvert = TypeConvert.SPACE_TO_RIGHT)
    private String setorResponsavel;

    @JsonProperty(value = "id_operacao")
    @TrancodeField(length = 50, typeConvert = TypeConvert.SPACE_TO_RIGHT)
    private String idOperacao;

    @JsonProperty(value = "nome_cliente")
    @TrancodeField(length = 40, typeConvert = TypeConvert.SPACE_TO_RIGHT)
    private String nomeCliente;

    @JsonProperty(value = "telefoneResponsavel")
    @MaskField(regex = "[0-9]", replaceWith = "#")
    @TrancodeField(length = 20, typeConvert = TypeConvert.SPACE_TO_RIGHT)
    private String telefoneResponsavel;
```


Pocket – Treinamento Apache Camel + Spring Boot

Ministrante: Thiago Hernandes de Souza – thiago.hernandes.souza@everis.com



an NTT DATA Company

7 – Qt Route Exemplo

```
public class AppQTRoute extends RouteBuilder {

    public static final String ROUTE_CONECTOR_ID = "rota_exemplo_conector";
    public static final String ROUTE_ENDPOINT_CONECTOR = "direct:rota_exemplo_conector";

    @Override
    public void configure() throws Exception {
        from(ROUTE_ENDPOINT_CONECTOR).routeId(ROUTE_CONECTOR_ID)
            .convertBodyTo(String.class)
            .unmarshal().json(JsonLibrary.Jackson, QtEntradaDTO.class) // request
            .bean(MontaEntrada.class, "trataEntradaPost").id("montaEntrada")
            .toD("qt://QT0T004X?mensagemMontadaQt=false&codigoCanal=75&classeRetorno=br.com.itaú.sigla.fuse.model.QtSaidaDTO")
            .id("qt")
            .log("Body received from QT: ${body}")
            .marshal().json(JsonLibrary.Jackson, QtHandleSaidaDTO.class) // response
            .bean(MontaEntrada.class, "teste") // SAÍDA pode ser dentro da mesma classe

            .toD("qt://QT0T004X?mensagemMontadaQt=false&codigoCanal=75&classeRetorno=br.com.itaú.sigla.fuse.model.QtSaidaDTO")
            .id("qt")
            .log("Body received from QT: ${body}")
            .marshal().json(JsonLibrary.Jackson, QtHandleSaidaDTO.class) // response
            .bean(MontaEntrada.class, "teste")

            .marshal().json(JsonLibrary.Jackson) // obrigatório
        .end();
    }
}
```

7.1 @AreaContrato – entrada

```
@AreaContrato
public class QtEntradaDTO {

    @Campo(Nome = "IND-STAT", Area = TipoArea.DADOS)
    private String indStat;

    @Campo(Nome = "TEMP-TIMEOUT", Area = TipoArea.DADOS)
    private Integer tempTimeout;

    @Campo(Nome = "CAMP01", Area = TipoArea.DADOS)
    private String campo1;

    @Campo(Nome = "CAMP02", Area = TipoArea.DADOS)
    private String campo2;

    @Campo(Nome = "CAMP03", Area = TipoArea.DADOS)
    private String campo3;

    @Campo(Nome = "RETORNO", Area = TipoArea.DADOS)
    private String retorno;

    @Campo(Nome = "IND-STAT2", Area = TipoArea.DADOS)
    private String indStat2;

    @Campo(Nome = "CAMP07", Area = TipoArea.DADOS)
```

Pocket – Treinamento Apache Camel + Spring Boot

Ministrante: Thiago Hernandes de Souza – thiago.hernandes.souza@everis.com



an NTT DATA Company

7.2 @AreaContrato – saída

```
@AreaContrato
public class QtSaidaDTO {

    @Campo(Nome="CAMP01", Area=TipoArea.DADOS)
    private String campo1;

    @Campo(Nome="CAMP02", Area=TipoArea.DADOS)
    //@MaskField
    private String campo2;

    @Campo(Nome="CAMP03", Area=TipoArea.DADOS)
    private String campo3;

    @Campo(Nome="CAMP04", Area=TipoArea.DADOS)
    private String campo4;

    @Campo(Nome="CAMP05", Area=TipoArea.DADOS)
    private String campo5;

    @Campo(Nome="CAMP06", Area=TipoArea.DADOS)
    private String campo6;

    @Campo(Nome="CAMP07", Area=TipoArea.DADOS)
    private String campo7;
```

8 – Confluence – Itaú

→ Documentação – Confluence;