Создание и управление definition json

Integration Platform

Exported on 2025-03-07 13:31:04

**Table of Contents**

**1 Описание wf/definition 5**

1.1 Стартер rest 9

1.1.1 Примеры 9

1.2 Стартер rabbitmq 10

1.2.1 Пример 12

1.3 Стартер kafkaConsumer 13

1.3.1 Примеры 16

1.4 Стартер sapInbound 20

1.4.1 Пример 21

1.5 Стартер scheduler 22

1.5.1 Пример 23

1.6 Стартер mail\_consumer 23

1.6.1 Пример 24

1.7 Составной Activity 25

1.8 Примитив await\_for\_message 29

1.8.1 Пример 31

1.9 Примитив rest\_call 31

1.9.1 Пример 37

1.10 Примитив db\_call 42

1.10.1 Пример 45

1.11 Примитив send\_to\_rabbitmq 48

1.11.1 Пример 50

1.12 Примитив send\_to\_kafka 50

1.12.1 Пример 54

1.13 Примитив send\_to\_s3 55

1.13.1 Пример 57

1.14 Примитив send\_to\_sap 58

1.14.1 Пример 60

1.15 Примитив xslt\_transform 61

1.15.1 Пример 63

1.16 Примитив transform 64

1.16.1 Пример 66

1.17 Инъекция inject 67

1.17.1 Пример 67

1.18 Условие switch 68

1.18.1 Пример 69

1.19 Parallel 70

1.19.1 Пример 71

1.20 Timer 72

1.20.1 Пример 73

**2 JS Path 74**

2.1 Пример 77

**3 LUA 82**

3.1 Пример 84

**4 Примеры 86**

4.1 WF1 Схема содержит основные типы активити, исключая: SAP, send\_to\_sap, await\_for\_message 86

4.2 WF 2 json\_to\_xml, rest\_call, xml\_to\_json, send\_to\_sap 94

4.3 WF 3 sap\_inbound 99

4.4 WF 4 Kafka-kafka 103

4.5 WF 4 Преобразование параметров, пришедших из await\_for\_message 105

* [Описание wf/definition](#ux3syzc4hssh)
  + [Стартер rest](#pyunlekalvoy)
    - [Примеры](#zczehpcawb5b)
  + [Стартер rabbitmq](#nv4urhq2limd)
    - [Пример](#robh88usrfp4)
  + [Стартер kafkaConsumer](#ail68sqsbio)
    - [Примеры](#r7aqnqhy9rdo)
  + [Стартер sapInbound](#yb0xom6wrj44)
    - [Пример](#mmbetbso7yg1)
  + [Стартер scheduler](#4442144icu3e)
    - [Пример](#c2i6nb4yhoit)
  + [Стартер mail\_consumer](#3eom1i9z3z2g)
    - [Пример](#kn594su4oqkb)
  + [Составной Activity](#aviokoesctp3)
  + [Примитив await\_for\_message](#4hj29b8bzako)
    - [Пример](#pi05jfldwnyz)
  + [Примитив rest\_call](#kg42t4jyt0jm)
    - [Пример](#hah7lkgws03m)
  + [Примитив db\_call](#r57wxdzb0zf9)
    - [Пример](#akau1kpga5f7)
  + [Примитив send\_to\_rabbitmq](#r5egb5cltzay)
    - [Пример](#yyxzullo2mn)
  + [Примитив send\_to\_kafka](#fk96lke8jt71)
    - [Пример](#x6889opotxpj)
  + [Примитив send\_to\_s3](#cenawx90wsty)
    - [Пример](#n13pc0b9rwh7)
  + [Примитив send\_to\_sap](#xqhkyts6pf58)
    - [Пример](#kd1ilzya5zy7)
  + [Примитив xslt\_transform](#ur55n91df0vm)
    - [Пример](#3nvkxpvz5jqu)
  + [Примитив transform](#uhssk2x6g27i)
    - [Пример](#ezsj69t5eg9t)
  + [Инъекция inject](#8ka14hfocys4)
    - [Пример](#t9dtnjgx5q6j)
  + [Условие switch](#rd6oroh4ks9b)
    - [Пример](#ozry57tooly2)
  + [Parallel](#m0cf268sxlkq)
    - [Пример](#1gbw0t4a3ase)
  + [Timer](#gw9k2odme4bm)
    - [Пример](#mihr234yhfxk)
* [JS Path](#v46j1cyjhg55)
  + [Пример](#i4g16e2xve4a)
* [LUA](#hjrh7jn7qtgl)
  + [Пример](#u426pccm5ulb)
* [Примеры](#9duvajntgixs)
  + [WF1 Схема содержит основные типы активити, исключая: SAP, send\_to\_sap, await\_for\_message](#2cdxiz41yue)
  + [WF 2 json\_to\_xml, rest\_call, xml\_to\_json, send\_to\_sap](#4nrkhlfmjeip)
  + [WF 3 sap\_inbound](#6ksh8i50rbn0)
  + [WF 4 Kafka-kafka](#efvwraq7sze4)
  + [WF 4 Преобразование параметров, пришедших из await\_for\_message](#xdboeshsu89s)

# Описание wf/definition

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | Тип WF.  Чаще всего **complex**. Составной тип примитива, используется для создания БП из нескольких действий. Выполняется и обрабатывается последовательность действий (activity).  так же может быть   * await\_for\_message. Ожидание вызова от внешней системы/внешнего БП с заранее заданным текстом. * rest\_call. Вызов МС (REST API). * db\_call. Вызов функции или выборки из БД. * send\_to\_rabbitmq. Отправка сообщения в очередь rabbitmq * send\_to\_sap. Отправка idoc в sap * transform. трансформация. | обязательное |
| name | String255 | Имя WF | обязательное |
| tenantId | String255 | id системы которая использует WF | необязательное, значение по умолчанию "default" |
| version | Int | Версия WF | необязательное, значение по умолчанию 1, при каждом редактировании WF увеличивается на 1 |
| description | String4000 | Описание WF | необязательное |
| compiled | Составной Compiled | Описание составного типа WF | обязательное, если примитив составной (type = "complex") |
| details | Составной DefinitionDetails | Для типа БП **complex** - описание ожидаемых параметров на вход БП  Для других типов БП - описание деталей БП | обязательное, если примитив простой (type != "complex") |
| flowEditorConfig | Составной | Данные для UI, на исполнение схемы они не влияют | необязательное |

Составной DefinitionDetails

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| inputValidateSchema | JsonObject | Стандартная json schema <https://www.jsonschemavalidator.net/>] с описанием механизма проверки входящих переменных.  С помощью этого параметра можно задать параметры, ожидаемые на старт БП. Если старт БП не будет соответствовать параметрам, то экземпляр БП упадет с ошибкой 400. | необязательное |
| outputValidateSchema | JsonObject | Стандартная json schema <https://www.jsonschemavalidator.net/>] с описанием механизма проверки исходящих переменных | необязательное |
| starters | Массив из составных starters | Описание параметров старта | обязательный |
| sendToKafkaConfig | Составной sendToKafkaConfig | Описание деталей вызова | обязательное, если FlowDefinition.type = 'send\_to\_kafka' |
| sendToS3Config | Составной sendToS3Config | Описание деталей вызова | обязательное, если FlowDefinition.type = 'send\_to\_s3' |
| restCallConfig | Составной restCallConfig | Описание вызова REST | обязательное, если FlowDefinition.type = 'rest\_call' |
| xsltTransformConfig | Составной xsltTransformConfig | Описание xslt-трансформации | обязательное, если FlowDefinition.type = 'xslt\_transform' |
| databaseCallConfig | Составной databaseCallConfig | Описание вызова БД (select или function) | обязательное, если FlowDefinition.type = 'db\_call' |
| sendToRabbitmqConfig | Составной sendToRabbitmqConfig | Описание отправки сообщения в Rabbitmq | обязательное, если FlowDefinition.type = 'send\_to\_rabbitmq' |
| awaitForMessageConfig | Составной awaitForMessageConfig | Описание ожидаемого сообщения от внешней системы | обязательное, если FlowDefinition.type = 'await\_for\_message' |
| sendToSapConfig | Составной sendToSapConfig | Описание параметров отправки idoc в sap | обязательное, если FlowDefinition.type = 'send\_to\_sap' |

Составной starters

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | Тип старта:   * kafka\_consumer - старт схемы производится при вычитывании сообщения из очереди кафка * sap\_inbound - старт схемы производится при получении idoc в указанный канал SAP * rest\_call - старт схемы производится при вызове low-code с id схемы. Если не указан стартер, то по умолчанию присваивается rest\_call * scheduler - старт схемы по расписанию * rabbitmq\_consumer - старт схемы производится при вычитывании сообщения из очереди rabbitmq * mail\_consumer - старт схемы производится при вычитывании всех непрочитанных писем почтового ящика | обязательное |
| name | String255 | Имя стартера | обязательное для всех типов старта кроме rest\_call |
| description | String255 | Описание стартера | необязательное |
| rabbitmqConsumer | Составной rabbitmqConsumer | Описание деталей стартера подключения rabbitmq | обязательное, если стартер rabbitmq |
| kafkaConsumer | Составной kafkaConsumer | Описание деталей подключения стартера kafka | обязательное, если стартер kafka |
| sapInbound | Составной sapInbound | Описание деталей подключения стартера SAP | обязательное, если стартер SAP |
| shedulerStarter | Составной shedulerStarter | Описание деталей работы стартера по расписанию | обязательное, если стартер по расписанию |

Составной Compiled

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| start | String255 | Указывается activity id, с которого начинается WF (точка входа start) | обязательное, если примитив составной (type = "complex") |
| activities | Составной Activity | Массив с описанием всех Activity данного WF | обязательное, если примитив составной (type = "complex") |
| outputTemplate | JsonObject | Фильтр всех данных запуска по завершению процесса  Если старт схемы будет производится с помощью синхронного REST-запроса и в ответ необходимо получить только определенные параметры, для более удобной их обработки, то необходимо указать необходимые для фильтрации параметры. | необязательный |

## Стартер rest

Составной starters

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | rest\_call  **Важно!** Если далее в схеме планируется использовать отдельные данные, которые пришли на старте, то обязательно необходимо указать details.inputValidateSchema, в которой они будут объявлены. Если планируется просто передать данные старта без изменений, можно использовать стандартную переменную jp{wf.initVariables} | обязательное |

### Примеры

На страте приходит xml и далее в схеме используется преобразование данных - необходимо в явном виде указать формат {"type": "string","stringFormat": "xml"}

    "details": {  
        "inputValidateSchema": {  
            "$schema": "[http://json-schema.org/draft-04/schema#](http://json-schema.org/draft-04/schema)",  
            "type": "object",  
            "properties": {  
                "mis\_xml": {  
                    "type": "string",  
                    "stringFormat": "xml"  
                },  
                "sap\_xml": {  
                    "type": "string",  
                    "stringFormat": "xml"  
                },  
                "sap\_idoc": {  
                    "type": "string",  
                    "stringFormat": "xml"  
                }  
            },  
            "required": [  
                "sap\_xml"  
            ]  
        },  
        "starters": [  
            {  
                "type": "rest\_call"  
            }  
        ]  
    }

На старте приходят параметры, которые далее используются в схеме

    "details": {  
        "inputValidateSchema": {  
            "$schema": "[http://json-schema.org/draft-04/schema#](http://json-schema.org/draft-04/schema)",  
            "properties": {  
                "apiId": {  
                    "type": "string"  
                },  
                "revisionId": {  
                    "type": "string"  
                }  
            },  
            "required": [  
                "apiId",  
                "revisionId"  
            ],  
            "type": "object"  
        }  
    }

## Стартер rabbitmq

Составной starters

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | rabbitmq\_consumer | обязательное |
| name | String255 | Имя стартера | обязательное для всех типов старта кроме rest\_call |
| description | String255 | Описание стартера | необязательное |
| rabbitmqConsumer | Составной rabbitmqConsumer | Описание деталей стартера подключения rabbitmq | обязательное, если стартер rabbitmq |

Составной rabbitmqConsumer

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| queue | String255 | Наименование очереди | обязательное, если стартер rabbitmq |
| connectionDef | Составной connectionDef | Параметры подключения | обязательное, если стартер rabbitmq |
| payloadValidateSchema | JsonObject | Схема валидации сообщения. Стандартная json schema <https://www.jsonschemavalidator.net/>] с описанием механизма проверки входящих переменных.  Важно! Если какие-то параметры не будут соответствовать обязательности, указанной в payloadValidateSchema, то по вычитанному сообщению старта схемы не будет. | необязательный |
| outputTemplate | JsonObject | Объявление переменных на основе полученных в сообщении данных.  Например,  {"properties":{"transactionID":{"type":"string"},"messageType":{"type":"string"},"version":{"type":"string"},"data":{"properties":{"eventType":{"type":"string"},"requestType":{"type":"string"},"serviceID":{"type":"string"}},"required":["eventType","requestType","serviceID"],"type":"object"}},"required":["transactionID","messageType","version","data"],"type":"object"}  Важно! Если объявлены переменные в outputTemplate, то необходимо так же описать details.inputValidateSchema, но уже во вложенном объекте payload:  {"type":"object","required":["payload"],"properties":{"payload":{"properties":{"transactionID":{"type":"string"},"messageType":{"type":"string"},"version":{"type":"string"},"data":{"properties":{"eventType":{"type":"string"},"requestType":{"type":"string"},"serviceID":{"type":"string"}},"required":["eventType","requestType","serviceID"],"type":"object"}},"required":["transactionID","messageType","version","data"],"type":"object"}}}  Т.е. добавляется объект  {     "type": "object",     "required": [         "payload"     ],     "properties": {         "payload": {  **<здесь описывается outputTemplate>**}     } } | необязательный |

Составной connectionDef для rabbitmqConsumer

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| userName | Srting255 | Имя пользователя | обязательный |
| userPass | Srting255 | Пароль пользователя | обязательный |
| addresses | Массив адресов | Адрес подключения и порт | обязательный |
| virtualHost | Srting255 | Виртуальный хост | обязательный |

### Пример

{  
    "details": {  
        "inputValidateSchema": {  
            "type": "object",  
            "required": [  
                "payload"  
            ],  
            "properties": {  
                "payload": {  
                    "type": "object"  
                }  
            }  
        },  
        "starters": [  
            {  
                "name": "february\_03\_starter",  
                "type": "rabbitmq\_consumer",  
                "description": "",  
                "rabbitmqConsumer": {  
                    "queue": "qName",  
                    "connectionDef": {  
                        "userName": "guest",  
                        "userPass": "guest",  
                        "addresses": [  
                            "localhost:5672"  
                        ],  
                        "virtualHost": "/"  
                    },  
                    "payloadValidateSchema": {},  
                    "outputTemplate": {  
                        "order": "jp{payload}"  
                    }  
                }  
            }  
        ]  
    }  
}

## Стартер kafkaConsumer

Составной starters

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | kafka\_consumer | обязательное |
| name | String255 | Имя стартера | обязательное для всех типов старта кроме rest\_call |
| description | String255 | Описание стартера | необязательное |
| kafka\_consumer | Составной kafka\_consumer | Описание деталей стартера подключения rabbitmq | обязательное, если стартер kafka\_consumer |

Составной kafka\_consumer

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| topic | String255 | Наименование топика | обязательное, если стартер kafka\_consumer |
| connectionDef | Составной connectionDef | Параметры подключения | обязательное, если стартер kafka\_consumer |
| payloadValidateSchema | JsonObject | Схема валидации сообщения. Стандартная json schema <https://www.jsonschemavalidator.net/>] с описанием механизма проверки входящих переменных.  Важно! Если какие-то параметры не будут соответствовать обязательности, указанной в payloadValidateSchema, то по вычитанному сообщению старта схемы не будет. | необязательный |
| keyValidateSchema | JsonObject | Схема валидации ключа. Стандартная json schema <https://www.jsonschemavalidator.net/>] с описанием механизма проверки входящих переменных.  Важно! Если какие-то параметры не будут соответствовать обязательности, указанной в keyValidateSchema, то по вычитанному сообщению старта схемы не будет. | необязательный |
| headersValidateSchema | JsonObject | Схема валидации заголовков. Стандартная json schema <https://www.jsonschemavalidator.net/>] с описанием механизма проверки входящих переменных.  Важно! Если какие-то параметры не будут соответствовать обязательности, указанной в headersValidateSchema, то по вычитанному сообщению старта схемы не будет. | необязательный |
| outputTemplate | JsonObject | Объявление переменных на основе полученных в сообщении данных.  Например,  {"properties":{"transactionID":{"type":"string"},"messageType":{"type":"string"},"version":{"type":"string"},"data":{"properties":{"eventType":{"type":"string"},"requestType":{"type":"string"},"serviceID":{"type":"string"}},"required":["eventType","requestType","serviceID"],"type":"object"}},"required":["transactionID","messageType","version","data"],"type":"object"}  Важно! Если объявлены переменные в outputTemplate, то необходимо так же описать details.inputValidateSchema, но уже во вложенном объекте payload:  {"type":"object","required":["payload"],"properties":{"payload":{"properties":{"transactionID":{"type":"string"},"messageType":{"type":"string"},"version":{"type":"string"},"data":{"properties":{"eventType":{"type":"string"},"requestType":{"type":"string"},"serviceID":{"type":"string"}},"required":["eventType","requestType","serviceID"],"type":"object"}},"required":["transactionID","messageType","version","data"],"type":"object"}}}  Т.е. добавляется объект  {     "type": "object",     "required": [         "payload"     ],     "properties": {         "payload": {  **<здесь описывается outputTemplate>**}     } } | необязательный |

Составной connectionDef  для kafka\_consumer

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| bootstrapServers | Srting255 | Адрес подключения | обязательный |
| authDef | Составной authDef | Параметры подключения | обязательный |

Составной authDef  для kafka\_consumer

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | Srting255 | Тип авторизации   * SASL * TLS - подключение mTls   **Важно!** Если нет авторизации, то authDef не нужно отправлять | обязательный, если есть авторизация |
| sasl | Составной sasl | Подключение по sasl | обязательный, если авторизация sasl |
| tls | Составной tls | Подключение по tls | обязательный, если авторизация tls |

Составной sasl

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| protocol | Srting255 | Протокол подключения:  SASL\_SSL  SASL\_PLAINTEXT | обязательный |
| mechanism | Srting255 | Механизм подключения  OAUTHBEARER - только для SASL\_PLAINTEXT  SCRAM-SHA-512 - для SASL\_SSL и SASL\_PLAINTEXT | обязательный |
| username | Srting255 | Логин | обязательный |
| password | Srting255 | Пароль | обязательный |
| sslDef | Составной sslDef | Сертификаты | обязательный для SCRAM-SHA-512 |
| tokenUrl | Srting255 | Url для проверки пользователя | обязательный для OAUTHBEARER |

Составной sslDef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| trustStoreType | Srting255 | Тип сертификата:  PEM | необязательный для OAUTHBEARER |
| trustStoreCertificates | Srting | Тело сертификатов  Важно! Тело сертификата начинается с  "-----BEGIN CERTIFICATE-----\r\n" и заканчивается "\r\n-----END CERTIFICATE-----\r\n" | необязательный для OAUTHBEARER |

Составной tls

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| keyStoreCertificates | Srting | публичный ключ клиента для аутентификации в формате PEM или base64(PEM)  user.crt | обязательный для TLS |
| keyStoreKey | Srting | приватный ключ клиента в формате PEM или base64(PEM)  user.key | обязательный для TLS |
| trustStoreCertificates | Srting | корневой сертификат в формате PEM или base64(PEM)  ca.crt | обязательный для TLS |
| trustStoreType | Srting255 | Тип сертификата:  PEM | обязательный для TLS |

### Примеры

**Указан только outputTemplate**

    "details": {  
        "inputValidateSchema": {  
            "type": "object",  
            "required": [  
                "payload"  
            ],  
            "properties": {  
                "payload": {  
                    "type": "object"  
                }  
            }  
        },  
        "starters": [  
            {  
                "name": "Kafka-kafka-KION",  
                "type": "kafka\_consumer",  
                "kafkaConsumer": {  
                    "topic": "rtk-kion",  
                    "connectionDef": {  
                        "bootstrapServers": "11.111.111.11:9094",  
                        "authDef": {  
                            "type": "TLS",  
                            "tls": {  
                                "trustStoreType": "PEM",  
                                "trustStoreCertificates": "",  
                                "keyStoreKey": "",  
                                "keyStoreCertificates": ""  
                            }  
                        }  
                    },  
                    "payloadValidateSchema": {},  
                    "keyValidateSchema": {},  
                    "headersValidateSchema": {},  
                    "outputTemplate": {  
                        "payload": "jp{payload}"  
                    }  
                }  
            }  
        ]  
    }

**Указана только payloadValidateSchema**

   "details": {  
        "inputValidateSchema": {  
            "type": "object",  
            "required": [  
                "payload"  
            ],  
            "properties": {  
                "payload": {  
                    "properties": {  
                        "transactionID": {  
                            "type": "string"  
                        },  
                        "messageType": {  
                            "type": "string"  
                        },  
                        "version": {  
                            "type": "string"  
                        },  
                        "data": {  
                            "properties": {  
                                "eventType": {  
                                    "type": "string"  
                                },  
                                "requestType": {  
                                    "type": "string"  
                                },  
                                "serviceDeskID": {  
                                    "type": "string"  
                                }  
                            },  
                            "required": [  
                                "eventType"  
                            ],  
                            "type": "object"  
                        }  
                    },  
                    "required": [  
                        "transactionID",  
                        "messageType",  
                        "version",  
                        "data"  
                    ],  
                    "type": "object"  
                }  
            }  
        },  
        "starters": [  
            {  
                "name": "SD-S1-workflow\_test",  
                "type": "kafka\_consumer",  
                "kafkaConsumer": {  
                    "topic": "testing",  
                    "connectionDef": {  
                        "bootstrapServers": "[kafka.ru](http://bootstrap-scram.kafka.s7.intp-dev.mts-corp.ru):443",  
                        "authDef": {  
                            "type": "SASL",  
                            "sasl": {  
                                "protocol": "SASL\_SSL",  
                                "mechanism": "SCRAM-SHA-512",  
                                "username": "",  
                                "password": "",  
                                "sslDef": {  
                                    "trustStoreType": "PEM",  
                                    "trustStoreCertificates": ""  
                                }  
                            }  
                        }  
                    },  
                    "payloadValidateSchema": {  
                        "properties": {  
                            "transactionID": {  
                                "type": "string"  
                            },  
                            "messageType": {  
                                "type": "string"  
                            },  
                            "version": {  
                                "type": "string"  
                            },  
                            "data": {  
                                "properties": {  
                                    "eventType": {  
                                        "type": "string"  
                                    },  
                                    "requestType": {  
                                        "type": "string"  
                                    },  
                                    "serviceDeskID": {  
                                        "type": "string"  
                                    }  
                                },  
                                "required": [  
                                    "eventType"  
                                ],  
                                "type": "object"  
                            }  
                        },  
                        "required": [  
                            "transactionID",  
                            "messageType",  
                            "version",  
                            "data"  
                        ],  
                        "type": "object"  
                    }  
                }  
            }  
        ]  
    }

**Указан payloadValidateSchema и outputTemplate**

    "details": {  
        "inputValidateSchema": {  
            "type": "object",  
            "properties": {  
                "apiId": {  
                    "type": "string"  
                },  
                "revisionId": {  
                    "type": "string"  
                }  
            },  
            "required": [  
                "apiId",  
                "revisionId"  
            ]  
        },  
        "starters": [  
            {  
                "name": "REST-kafka19081712",  
                "type": "kafka\_consumer",  
                "kafkaConsumer": {  
                    "topic": "testing",  
                    "connectionDef": {  
                        "bootstrapServers": "[kafka.ru](http://bootstrap-scram.kafka.s7.intp-dev.mts-corp.ru):443",  
                        "authDef": {  
                            "type": "SASL",  
                            "sasl": {  
                                "protocol": "SASL\_SSL",  
                                "mechanism": "SCRAM-SHA-512",  
                                "username": "",  
                                "password": "",  
                                "sslDef": {  
                                    "trustStoreType": "PEM",  
                                    "trustStoreCertificates": ""  
                                }  
                            }  
                        }  
                    },  
                    "payloadValidateSchema": {  
                        "properties": {  
                            "apiId": {  
                                "type": "string"  
                            },  
                            "revisionId": {  
                                "type": "string"  
                            }  
                        },  
                        "required": [  
                            "apiId",  
                            "revisionId"  
                        ],  
                        "type": "object"  
                    },  
                    "keyValidateSchema": {},  
                    "headersValidateSchema": {},  
                    "outputTemplate": {  
                        "apiId": "jp{payload.apiId}",  
                        "revisionId": "jp{payload.revisionId}",  
                        "kafkaMessagePayload": "jp{payload}"  
                    }  
                }  
            }  
        ]  
    }

## Стартер sapInbound

Составной starters

**Важно!** idoc, пришедший на старт схемы сохраняется в стандартную переменную jp{idoc}

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | sapInbound | обязательное |
| inboundDef | Составной inboundDef | Описание деталей стартера подключения inboundDef | обязательное, если стартер sapInbound |

Составной inboundDef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| name | String255 | Имя стартера | обязательное для всех типов старта кроме rest\_call |
| connectionDef | Составной connectionDef | Параметры подключения | обязательное, если стартер sapInbound |
| props | Составной props | Дополнительные параметры | обязательное, если стартер sapInbound |

Составной connectionDef для sapInbound

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| name | String255 | Имя стартера | обязательное для всех типов старта кроме rest\_call |
| props | Составной props | Параметры подключения | обязательное, если стартер sapInbound |

Составной props для connectionDef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| jco.client.lang | Srting255 | Язык клиента | обязательный |
| jco.client.passwd | Srting255 | Пароль | обязательный |
| jco.client.user | Srting255 | Логин | обязательный |
| jco.client.sysnr | Int | номер SAP-системы | обязательный |
| jco.destination.pool\_capacity | Int | максимальное количество подключений, которые могут находиться в пуле подключений для destination | обязательный |
| jco.destination.peak\_limit | Int | максимальное количество одновременных подключений для destionation | обязательный |
| jco.client.client | Int | номер клиента в SAP-системе | обязательный |
| jco.client.ashost | Srting255 | Хост | обязательный |

Составной props для inboundDef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| jco.server.gwhost | Srting255 | gwhost | обязательный |
| jco.server.progid | Srting255 | progid | обязательный |
| jco.server.gwserv | Srting255 | gwserv | обязательный |
| jco.server.connection\_count | Int | Количество подключений | обязательный |

### Пример

 "details": {  
        "inputValidateSchema": {},  
        "starters": [  
            {  
                "type": "sap\_inbound",  
                "sapInbound": {  
                    "inboundDef": {  
                        "name": "sapInbound-SAP-LP",  
                        "connectionDef": {  
                            "name": "sapConnection-SAP-LP",  
                            "props": {  
                                "jco.client.lang": "EN",  
                                "jco.destination.peak\_limit": 10,  
                                "jco.client.client": 400,  
                                "jco.client.sysnr": 10,  
                                "jco.destination.pool\_capacity": 3,  
                                "jco.client.ashost": "[m-1.teat.ru](http://m-1.teat.ru)",  
                                "jco.client.user": "user",  
                                "jco.client.passwd": "passwd"  
                            }  
                        },  
                        "props": {  
                            "jco.server.gwhost": "/H/[test.ru/S/3310](http://test.ru/S/3310)",  
                            "jco.server.progid": "L\_1",  
                            "jco.server.gwserv": "sap",  
                            "jco.server.connection\_count": 2  
                        }  
                    }  
                }  
            }  
        ]  
    }

## Стартер scheduler

Составной starters

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | scheduler | обязательное |
| name | String255 | Имя стартера | обязательное для всех типов старта кроме rest\_call |
| description | String255 | Описание стартера | необязательное |
| scheduler | Составной scheduler | Описание деталей стартера по расписанию | обязательное, если стартер scheduler |

Составной scheduler

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | cron | обязательное, если стартер scheduler |
| cron | Составной cron | описание расписания  <https://cronexpressiontogo.com/every-2-months> | обязательное, если стартер scheduler |
| startDateTime | Data | Дата начала работы стартера | обязательный |
| outputTemplate | Data | Дата окончания работы стартера | необязательный |

Составной cron

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| dayOfWeek | Srting255 | День недели | обязательно заполнение хотя бы одного параметра cron |
| month | Srting255 | Количество месяцев, через которое повторять запуск | обязательно заполнение хотя бы одного параметра cron |
| dayOfMonth | Srting255 | День месяца | обязательно заполнение хотя бы одного параметра cron |
| hour | Srting255 | Количество месяцев, через которое повторять запуск | обязательно заполнение хотя бы одного параметра cron |
| minute | Srting255 | Количество месяцев, через которое повторять запуск | обязательно заполнение хотя бы одного параметра cron |

### Пример

At 00:00, on day 1 of the month, every 2 months

    "details": {  
        "starters": [  
            {  
                "name": "scheduler\_name",  
                "type":"scheduler",  
                "description": "Тестовый стартер",  
                "scheduler": {  
                    "type": "cron",  
                    "cron": {  
                      "dayOfWeek": "0",  
                      "month": "\*/2",  
                      "dayOfMonth": "1",  
                      "hour": "0",  
                      "minute": "0"  
                    },  
                    "startDateTime": "2025-03-06T08:39:30.446Z",  
                    "endDateTime": "2025-03-06T08:39:30.446Z"  
                  }  
            }  
        ]  
    }

## Стартер mail\_consumer

Составной starters

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | mail\_consumer | обязательное |
| name | String255 | Имя стартера | обязательное для всех типов старта кроме rest\_call |
| description | String255 | Описание стартера | необязательное |
| mailConsumer | Составной mailConsumer | Описание деталей стартера | обязательное, если стартер mail\_consumer |

Составной mailConsumer

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| connectionDef | Составной connectionDef | описание расписания | обязательное, если стартер scheduler |
| mailFilter | Data | Дата начала работы стартера | обязательный |

Составной connectionDef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| protocol | Srting255 | imap | обязательно для mail\_consumer |
| host | Srting255 | Хост подключения | обязательно для mail\_consumer |
| port | Srting255 | Порт | обязательно для mail\_consumer |
| mailAuth | Составной mailAuth | Параметры авторизации | обязательно для mail\_consumer |

Составной mailAuth

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| username | Srting255 | Почта | обязательно для mail\_consumer |
| password | Srting255 | Пароль | обязательно для mail\_consumer |

Составной mailFilter

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| senders | Массив senders | Почта отправителя письма | необязательный |
| subjects | Массив subjects | Тема письма  **Важно!** Полное совпадение | необязательный |
| startMailDateTime | Date | Дата, с которой необходимо начать вычитывать непрочитанные сообщения.  Если параметр не указан, то будут вычитываться все непрочитанные письма. | необязательный |

### Пример

    "details": {  
        "starters": [  
            {  
                "name": "mail\_starter",  
                "type": "mail\_consumer",  
                "description": "Тестовый стартер",  
                "tenantId": "default",  
                "mailConsumer": {  
                    "connectionDef": {  
                        "protocol": "imap",  
                        "host": "[imap.yandex.ru](http://imap.yandex.ru)",  
                        "port": 993,  
                        "mailAuth": {  
                            "username": "test@test.test",  
                            "password": "test"  
                        }  
                    },  
                    "mailFilter": {  
                        "senders": ["mymail@[mail.ru](http://mail.ru)"],  
                        "subjects": ["topic1", "topic2"],  
                        "startMailDateTime": "2025-01-01T10:01:23.000+03:00"  
                    }  
                }  
            }  
        ]  
    }

## Составной Activity

В зависимости от типа activity его описание имеет свои особенности. Здесь указано общее описание, ниже будет представлено описание параметров в зависимости от типа activity.

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String255 | ИД activity.  Важно! В рамках одного БП не должно быть activity с одинаковым ИД. | обязательное |
| description | String255 | Описание шага | необязательное |
| transition | String255 | ИД следующего activity.  Если БП или ветка заканчивается на данном activity, то значение будет  "transition": null | обязательное |
| type | String255 | Тип activity, может быть следующих типов:  **1. workflow\_call**. Выполнение ранее заведенного БП (в данном случае заполняется блок workflowRef) или описанного далее Primitive или элемента Pipeline, в данном случае указывается его тип в блоке workflowDef:   * + await\_for\_message. Ожидание вызова от внешней системы/внешнего БП с заранее заданным текстом.   + rest\_call. Вызов МС (REST API).   + db\_call. Вызов функции или выборки из БД.   + send\_to\_rabbitmq. Отправка сообщения в очередь rabbitmq   + send\_to\_sap. Отправка idoc в sap   + xslt\_transform. XSLT трансформация.   **2. inject**. Вставка констант, принудительное изменение параметров БП или задание определенного поведения БП с помощью инъекций параметров. Можно использовать для начала цикла или тестирования.  **3. switch**. Выполнение условия (если ..., то, иначе...).  **4. timer**. Автоматический запуск через заданный промежуток времени.  **5. transform**. Трансформация xml\_to\_json или json\_to\_xml  **6. parallel**. Выполнение в параллели нескольких заданных действий. | обязательное |
| workflowCall | Составной WorkflowCall | Подробное описание параметров activity с типом workflow\_call. | обязательное, если type = "**workflow\_call**" или "**inject**" |
| injectData | JsonObject | Описание вставляемых данных в формате Json. | обязательное, если type = "**inject**" |
| outputFilter | String255 | Описывает трансформацию исходящих данных после завершения activity, используется для переименования пришедших параметров, объявление переменных (в данном случае переменные необходимо в явном виде указать в outputValidateSchema).  На данный момент после запуска activity выходные переменные движок добавляет в общий скоуп и передает в следующую activity, даже если не указан outputFilter. | необязательное |
| dataConditions | Составной dataConditions | Описание проверяемых условий и действий при успешном их выполнении. Скрипт условия в формате spel или json path.  Например: (spel{#name == 'exit'}) или jp{args.name}  **Важно!**При указании нескольких условий, обратите внимание на порядок их указания. При успешном выполнении первого по счету условия будет совершен переход к указанному в нем transition, не смотря на то, что может подходить еще одно из условий. | обязательное, если type = "**switch**" |
| defaultTransition | Составной defaultCondition | Описание поведения WF, если результат вызова всех condition = false. | обязательное, если type = "**switch**" |
| branches | Массив ИД activity | Список id activity данного WF, которые будут выполняться параллельно. Пример:  "branches": [                     "AR-1-ApiStatusCREATING",                     "AR-2-APIFirewallStatusCREATING"                 ] | обязательное, если type = "**parallel**" |
| completionType | String255 | Тип завершения параллельного activity, может быть следующих видов:  - **anyOf**. Завершение действия parallel возможно, когда завершится хотя бы одно из указанных действий  - **allOf**. Завершение действия parallel возможно, когда завершатся все указанные действия | обязательное, если type = "**parallel**" |
| timerDuration | String256  ISO 8601 duration format | Таймер, через какое время произойдет переход к следующему по схеме activity | обязательное, если type = "**timer**" |
| transform | Составной transform | Параметры трансформации | обязательное, если type = "**transform**" |

 Составной WorkflowCall

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| args | JsonObject | Входные параметры для примитива | необязательное |
| workflowRef | Составной Ref | Ссылка на сохраненный ранее шаблон WF.  **Важно!** businessKey подпроцесса будет сгенерирован автоматически. Если необходимо знать businessKey подпроцесса, то можно задать его выражением в workflowCall.args : {"businessKey": "spel{#wf.businessKey + '-1'}"} | обязательное, если нет workflowDef |
| workflowDef | Составной SimpleWorkflowDefinition | Описание вызываемого подпроцесса. | обязательное, если нет workflowRef |
| retryConfig | Составной retryConfig | Описание политики retry. Для всех activity | необязательный |
| failActivityResult | Составной failActivityResult | Переход к следующему шагу при неуспешном завершении при завершении retry. Для всех activity | необязательный |

Составной retryConfig

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| initialInterval | String256  ISO 8601 duration format | Интервал времени до первого повтора, по умолчанию 1 секунда  Например:  "PT20.345S" - 20.345 секунд  "PT15M" - 15 минут  "PT10H" - 10 часов  "P2D" - 2 дня  "P2DT3H4M" - 2 дня, 3 часа, 4 минуты | необязательный |
| maxInterval | String256  ISO 8601 duration format | Максимальный интервал времени между попытками (должен быть больше чем initialInterval), по умолчанию не ограниченный  Например:  "PT20.345S" - 20.345 секунд  "PT15M" - 15 минут  "PT10H" - 10 часов  "P2D" - 2 дня  "P2DT3H4M" - 2 дня, 3 часа, 4 минуты | необязательный |
| maxAttempts | Int | Максимальное количество повторов | необязательный |
| backoffCoefficient | float | Коэффициент увеличения интервала после каждого повтора(минимальное значение 1.0) по умолчанию 2.0. Число дробное | необязательный |

Составной failActivityResult

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| retryStates | Массив статусов | Если необходим переход к следующему шагу при неуспешном завершении при завершении retry, то указывается:  "retryStates": [                             "RETRY\_STATE\_MAXIMUM\_ATTEMPTS\_REACHED"                         ] | необязательный |
| variables | JsonString | Объявление переменных при выходе из retry для перехода к следующему шагу.  Важно! Если в данной activity указаны outputFilter, то все они должны быть перечислены в variables | обязательный, если у пользователя заданы outputFilter в данной activity |

Составной Ref

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String-UUID | id шаблона БП из справочника | обязательный, если нет name |
| name | Srting255 | Наименование шаблона | обязательный ,если нет id |
| version | Int | Версия шаблона | необязательный |
| tenantId | Srting255 | id системы которая использует шаблон | необязательный, по умолчанию "default" |

## Примитив await\_for\_message

await\_for\_message чаще всего используется для асинхронного взаимодействия, когда после вызова REST ожидается callback от внешней системы.

Формат callback, где messageName - равен указанному awaitForMessageConfig.MessageName, а в variables отправляются параметры от внешней системы, которые далее используются в схеме:

{  
    "businessKey": "API\_Firewall\_plunt\_33",  
    "messageName": "FirewallSdId",  
    "variables": {  
      "sdId": "RA0000006172339"  
    }  
}

**Важно!** Данные, пришедшие при вызове await\_for\_message сохраняются в стандартную переменную wf.consumedMessages.

Составной Activity

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String255 | ИД activity.  Важно! В рамках одного БП не должно быть activity с одинаковым ИД. | обязательное |
| description | String255 | Описание шага | необязательное |
| transition | String255 | ИД следующего activity.  Если БП или ветка заканчивается на данном activity, то значение будет  "transition": null | обязательное |
| type | String255 | workflow\_call | обязательное |
| workflowCall | Составной WorkflowCall | Описание activity с типом workflow\_call. | обязательное |

Составной WorkflowCall

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| args | JsonObject | Аргументы на вход activity | необязательное |
| workflowDef | Составной SimpleWorkflowDefinition | Описание вызываемого подпроцесса. | обязательное, если нет workflowRef |

Составной SimpleWorkflowDefinition

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | await\_for\_message | обязательное |
| details | Составной details | Детали подпроцесса | обязательное |

Составной details

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| awaitForMessageConfig | Составной awaitForMessageConfig | Описание ожидаемого сообщения от внешней системы. | обязательное |

Составной awaitForMessageConfig

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| MessageName | String255 | Наименование сообщения, которое отправит внешняя система.  Важно! Параметры messageName в вызове [**wf/instance/message**](https://workflow-engine.internal.intp-dev.mts-corp.ru/swagger-ui/index.html#/Workflow%20instance/signalToWorkflowInstance) и в описании activity должны быть равны | обязательное |

### Пример

**Пример кода**

 Пример activity await\_for\_message

| {  "id": "await\_for\_message",  "description": "Ожидание SdId",  "type": "workflow\_call",  "workflowCall": {  "workflowDef": {  "type": "await\_for\_message",  "details": {  "awaitForMessageConfig": {  "messageName": "FirewallSdId"  }  }  }  },  "transition": "REST"  } |
| --- |

**Code Block** **1 Пример**

## Примитив rest\_call

Составной Activity

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String255 | ИД activity.  Важно! В рамках одного БП не должно быть activity с одинаковым ИД. | обязательное |
| description | String255 | Описание шага | необязательное |
| transition | String255 | ИД следующего activity.  Если БП или ветка заканчивается на данном activity, то значение будет  "transition": null | обязательное |
| type | String255 | workflow\_call | обязательный |
| workflowCall | Составной WorkflowCall | Описание activity с типом workflow\_call. | обязательный |
| outputFilter | String255 | Описывает трансформацию исходящих данных после завершения activity, используется для переименования пришедших параметров, объявление переменных (в данном случае переменные необходимо в явном виде указать в outputValidateSchema).  На данный момент после запуска activity выходные переменные движок добавляет в общий скоуп и передает в следующую activity, даже если не указан outputFilter. | необязательный |

Составной WorkflowCall

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| args | JsonObject | Аргументы на вход activity | необязательный |
| retryConfig | Составной retryConfig | Описание политики retry | необязательный |
| workflowDef | Составной SimpleWorkflowDefinition | Описание вызываемого подпроцесса. | обязательный, если нет workflowRef |

Составной retryConfig

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| initialInterval | String256  ISO 8601 duration format | Интервал времени до первого повтора, по умолчанию 1 секунда  Например:  "PT20.345S" - 20.345 секунд  "PT15M" - 15 минут  "PT10H" - 10 часов  "P2D" - 2 дня  "P2DT3H4M" - 2 дня, 3 часа, 4 минуты | необязательный |
| maxInterval | String256  ISO 8601 duration format | Максимальный интервал времени между попытками (должен быть больше чем initialInterval), по умолчанию не ограниченный  Например:  "PT20.345S" - 20.345 секунд  "PT15M" - 15 минут  "PT10H" - 10 часов  "P2D" - 2 дня  "P2DT3H4M" - 2 дня, 3 часа, 4 минуты | необязательный |
| maxAttempts | Int | Максимальное количество повторов | необязательный |
| backoffCoefficient | float | Коэффициент увеличения интервала после каждого повтора(минимальное значение 1.0) по умолчанию 2.0. Число дробное | необязательный |

Составной SimpleWorkflowDefinition

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | rest\_call | обязательный |
| details | Составной details | Описание деталей вызова | обязательный |

Составной details

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| inputValidateSchema | JsonObject | Стандартная json schema <https://www.jsonschemavalidator.net/>] с описанием механизма проверки входящих переменных | необязательный |
| outputValidateSchema | JsonObject | Стандартная json schema <https://www.jsonschemavalidator.net/>] с описанием механизма проверки исходящих переменных | необязательный |
| restCallConfig | Составной restCallConfig | Описание деталей вызова | обязательный |

Составной restCallConfig

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| resultHandlers | Список значений составных predicate | Описание условий успешного вызова  Можно задать несколько predicate, в каждом из которых описать условия. Если заданы несколько predicate, то при проверке успешности они будут обрабатываться как ИЛИ  **Пример**  "resultHandlers": [ { "predicate": { "respCode": 200, "respValueAnyOf": [ { "jsonPath": "jp{body.status.status.statusId}", "values": [ "executor-reject", "rejected", "approvement-reject", "approvement-deadline", "revoke" ] } ] } }, { "predicate": { "respCode": 205, "respValueAnyOf": [ { "jsonPath": "jp{body.status.status.statusId}", "values": [ "executor-reject", "rejected", "approvement-reject", "approvement-deadline", "revoke" ] } ] } } ] | необязательный |
| restCallTemplateRef | Составной restCallTemplateRef | Описание на вызов шаблона rest запроса из справочника | обязательный, если нет restCallTemplateDef |
| restCallTemplateDef | Составной restCallTemplateDef | Параметры вызова rest запроса | обязательный, если нет restCallTemplateRef |

Составной predicate

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| respCode | Int | Код ответа | обязательный, если нет respCodeInterval |
| respCodes | Массив значений respCode | Список кодов ответов | обязательный, если нет respCodeInterval или respCode |
| respCodeInterval | Составной respCodeInterval | Интервал ответов | обязательный, если нет respCode |
| respValueAnyOf | Массив объектов pathValueValidation | Описание значений переменных  Можно задать несколько pathValueValidation, в каждом из которых описать условия. Если заданы несколько pathValueValidation, то при проверке успешности они будут обрабатываться как ИЛИ  **Пример**  "respValueAnyOf": [   { "jsonPath": "jp{body.prop1}",  "values": [ "val" ],  "and": {  "jsonPath": "jp{body.prop2}", "values": [ "val2" ] | необязательный |

Составной respCodeInterval

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| from | Int | Значение кода ответа от которого начинается интервал | необязательный |
| to | Int | Значение кода ответа на котором заканчивается интервал | необязательный |

Составной PathValueValidation

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| jsonPath | Srting255 | Путь к переменной | обязательный |
| values | Массив (String255 или boolean или int или double) | Значения переменной | обязательный |
| and | Составной pathValueValidation | Описание значений еще одной переменной. Заполняется, если необходимо добавить условие И с указанием значений еще одной переменной.  **Пример**  "respValueAnyOf": [   { "jsonPath": "jp{body.prop1}",  "values": [ "val" ],  "and": {  "jsonPath": "jp{body.prop2}", "values": [ "val2" ] | необязательный |

Составной restCallTemplateRef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String-UUID | id шаблона rest запроса из справочника | обязательный, если нет name |
| name | Srting255 | Наименование шаблона | обязательный ,если нет id |
| version | Int | Версия шаблона | необязательный |
| tenantId | Srting255 | id системы которая использует шаблон | необязательный, по умолчанию "default" |

Составной restCallTemplateDef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| method | Srting255 | Тип метода подключения POST, PUT и т.д. | обязательный, если не указан curl |
| url | Srting255 | url подключения | обязательный, если не указан curl |
| bodyTemplate | Srting255 | Тело запроса | необязательный |
| headers | Составной headers | headers запроса | обязательный, если не указан curl |
| curl | Srting255 | экранированный curl запроса  **Важно!** Параметры выгружаемого curl должны быть следующими:  Generate multiline snippet - **F**  Use long form options - **F**  Line continuation character - **\**  Quote Type - **SINGLE**  Set request timeout - **0**  Follow redirects - **T**  Trim request body fields - **F**  Use Silent Mode **- F** | обязательный, если не указаны method, url, headers |
| authDef | Составной authDef | Параметры авторизации | обязательный |

Составной headers

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| <headers> | Srting255 | header запроса | обязательный, если не указан curl |

Составной authDef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | Srting255 | Тип авторизации. Один из "basic", "oauth2" | обязательный |
| basic | Составной basic | Параметры для basic авторизации | обязательный, если тип авторизации "basic" |
| oauth2 | Составной oauth2 | Параметры для oauth2 авторизации | обязательный, если тип авторизации "oauth2" |

Составной basic

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| login | Srting255 | Логин | обязательный, если тип авторизации "basic" |
| password | Srting255 | Пароль | обязательный, если тип авторизации "basic" |

Составной oauth2

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| issuerLocation | Srting255 | url для проверки | обязательный, если тип авторизации "oauth2" |
| clientId | Srting255 | Id клиента | обязательный, если тип авторизации "oauth2" |
| clientSecret | Srting255 | Secret клиента | обязательный, если тип авторизации "oauth2" |
| grantType | Srting255 | На данный момент client\_credentials | обязательный, если тип авторизации "oauth2" |

### Пример

**Пример**

Пример rest\_call с описанной схемой исходящих параметров, объявлением переменных и параметрами retry, для сохранения параметров в БП

| {  "id": "activity-1",  "type": "workflow\_call",  "description": "",  "workflowCall": {  "workflowDef": {  "type": "rest\_call",  "details": {  "restCallConfig": {  "restCallTemplateDef": {  "method": "POST",  "url": "https://preprod.crm.ru/rest/",  "bodyTemplate": {  "fields": "jp{payload}"  },  "headers": {  "Content-Type": "application/json"  }  }  },  "outputValidateSchema": {  "properties": {  "result": {  "properties": {  "lead": {  "type": "object"  }  },  "required": [  "lead"  ],  "type": "object"  }  },  "required": [  "result"  ],  "type": "object"  }  }  },  "retryConfig": {  "maxAttempts": 20,  "initialInterval": "PT5S",  "maxInterval": "PT30M",  "backoffCoefficient": 1.5  }  },  "transition": "activity-2",  "outputFilter": {  "RESTbody": "jp{body}"  }  } |
| --- |

**Code Block** **2 Пример**

Пример рест с cURL

            {  
                "id": "activity-2",  
                "type": "workflow\_call",  
                "description": "Получение статуса",  
                "workflowCall": {  
                    "workflowDef": {  
                        "type": "rest\_call",  
                        "details": {  
                            "restCallConfig": {  
                                "restCallTemplateDef": {  
                                    "curl": "curl -L '[https://workflow.test.ru/api/v1/wf/search](https://workflow-engine-dpprod.intp.mts-corp.ru/api/v1/wf/definition/search)' -H 'accept: \*/\*' -H 'Content-Type: application/json' -d '{\n    \"name\": \"MaxActivitiShema\",\n    \"offset\": 0,\n    \"limit\": 25\n}'",  
                                    "authDef": {  
                                        "type": "oauth2",  
                                        "oauth2": {  
                                            "issuerLocation": "<https://isso.mts.ru/auth/realms/mts>",  
                                            "clientId": "clientId",  
                                            "clientSecret": "clientSecret",  
                                            "grantType": "client\_credentials"  
                                        }  
                                    }  
                                }  
                            },  
                            "outputValidateSchema": {  
                                "type": "array",  
                                "items": [  
                                    {  
                                        "type": "object",  
                                        "properties": {  
                                            "id": {  
                                                "type": "string"  
                                            },  
                                            "type": {  
                                                "type": "string"  
                                            },  
                                            "name": {  
                                                "type": "string"  
                                            },  
                                            "description": {  
                                                "type": "string"  
                                            },  
                                            "tenantId": {  
                                                "type": "string"  
                                            },  
                                            "createTime": {  
                                                "type": "string"  
                                            },  
                                            "changeTime": {  
                                                "type": "string"  
                                            },  
                                            "version": {  
                                                "type": "integer"  
                                            },  
                                            "status": {  
                                                "type": "string"  
                                            },  
                                            "ownerLogin": {  
                                                "type": "string"  
                                            }  
                                        },  
                                        "required": [  
                                            "id",  
                                            "type",  
                                            "name",  
                                            "description",  
                                            "tenantId",  
                                            "status"  
                                        ]  
                                    }  
                                ]  
                            }  
                        }  
                    },  
                    "retryConfig": {}  
                },  
                "outputFilter": {  
                    "sd\_body": "jp{$.body}",  
                    "sd\_status": "jp{$.body[0].status}",  
                    "wf\_id": "jp{$.body[0].id}"  
                },  
                "transition": "activity-16"  
            }

 Пример REST с успешным завершением при выходе из ретраев

            {  
                "id": "SD-2-SDAPIpublicationStatus",  
                "description": "Запрос статуса заявки Service Desk",  
                "type": "workflow\_call",  
                "workflowCall": {  
                    "retryConfig": {  
                        "initialInterval": "PT1S",  
                        "maxInterval": "PT5S",  
                        "maxAttempts": 10,  
                        "backoffCoefficient": 1.2  
                    },  
                    "failActivityResult": {  
                        "retryStates": [  
                            "RETRY\_STATE\_MAXIMUM\_ATTEMPTS\_REACHED"  
                        ],  
                        "variables": {  
                            "sd\_status": "ERROR",  
                            "sd\_body": "ERROR"  
                        }  
                    },  
                    "workflowDef": {  
                        "type": "rest\_call",  
                        "details": {  
                            "inputValidateSchema": {},  
                            "outputValidateSchema": {  
                                "type": "object",  
                                "required": [  
                                    "status"  
                                ],  
                                "properties": {  
                                    "status": {  
                                        "$ref": "#/definitions/root\_status"  
                                    }  
                                },  
                                "definitions": {  
                                    "root\_status": {  
                                        "type": "object",  
                                        "properties": {  
                                            "status": {  
                                                "$ref": "#/definitions/status\_status"  
                                            }  
                                        },  
                                        "required": [  
                                            "status"  
                                        ]  
                                    },  
                                    "status\_status": {  
                                        "type": "object",  
                                        "properties": {  
                                            "statusId": {  
                                                "type": "string"  
                                            }  
                                        },  
                                        "required": [  
                                            "statusId"  
                                        ]  
                                    }  
                                }  
                            },  
                            "restCallConfig": {  
                                "resultHandlers": [  
                                    {  
                                        "predicate": {  
                                            "respCode": 200,  
                                            "respValueAnyOf": [  
                                                {  
                                                    "jsonPath": "jp{body.status.status.statusId}",  
                                                    "values": [  
                                                        "executor-reject",  
                                                        "rejected",  
                                                        "approvement-reject",  
                                                        "approvement-deadline",  
                                                        "revoke"  
                                                    ]  
                                                }  
                                            ]  
                                        }  
                                    }  
                                ],  
                                "restCallTemplateDef": {  
                                    "curl": "curl --location --request GET '<http://wiremock:8080/api/herald/v1/application/RA0-solved>' \\\r\n--header 'accept: application/json'",  
                                    "authDef": {  
                                        "type": "oauth2",  
                                        "oauth2": {  
                                            "issuerLocation": "<https://isso.mts.ru/auth/realms/mts>",  
                                            "clientId": "clientId",  
                                            "clientSecret": "clientSecret",  
                                            "grantType": "client\_credentials"  
                                        }  
                                    }  
                                }  
                            }  
                        }  
                    }  
                },  
                "transition": "Flow",  
                "outputFilter": {  
                    "sd\_status": "jp{body.status.status.statusId}",  
                    "sd\_body": "jp{body}"  
                }  
            }

## Примитив db\_call

Составной Activity

Важно! Результат вызова БД сохраняется в стандартную переменную jp{databaseCallResult}

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String255 | ИД activity.  Важно! В рамках одного БП не должно быть activity с одинаковым ИД. | обязательное |
| description | String255 | Описание шага | необязательное |
| transition | String255 | ИД следующего activity.  Если БП или ветка заканчивается на данном activity, то значение будет  "transition": null | обязательное |
| type | String255 | workflow\_call | обязательный |
| workflowCall | Составной WorkflowCall | Описание activity с типом workflow\_call. | обязательный |
| outputFilter | String255 | Описывает трансформацию исходящих данных после завершения activity, используется для переименования пришедших параметров, объявление переменных (в данном случае переменные необходимо в явном виде указать в outputValidateSchema).  На данный момент после запуска activity выходные переменные движок добавляет в общий скоуп и передает в следующую activity, даже если не указан outputFilter. | необязательный |

Составной WorkflowCall

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| args | JsonObject | Аргументы на вход activity  **Важно!** Если в селекте используются переменные с данными из схемы, то их обращение описывается в args  **Важно!** При вызове процедуры или функции в args описываются входные параметры. Например, | необязательный |
| workflowDef | Составной SimpleWorkflowDefinition | Описание вызываемого подпроцесса. | обязательный, если нет workflowRef |

Составной SimpleWorkflowDefinition

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | db\_call | обязательный |
| details | Составной details | Описание деталей вызова | обязательный |

Составной details

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| databaseCallConfig | Составной databaseCallConfig | Описание деталей вызова | обязательный |

Составной databaseCallConfig

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| databaseCallRef | Составной databaseCallRef | Параметры шаблона подключения к БД из справочника | обязательный, если нет databaseCallDef |
| databaseCallDef | Составной databaseCallDef | Параметры подключения к БД | обязательный, если нетdatabaseCallRef |
| dataSourceId | Srting255 | ID шаблона dataSource из справочника | обязательный, если нет dataSourceDef |
| dataSourceDef | Составной dataSourceDef | Параметры для авторизации | обязательный, если нет dataSourceId |

Составной databaseCallRef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String-UUID | id шаблона параметров подключения к БД из справочника | обязательный, если нет name |
| name | Srting255 | Наименование шаблона | обязательный ,если нет id |
| version | Int | Версия шаблона | необязательный |
| tenantId | Srting255 | id системы которая использует шаблон | необязательный, по умолчанию "default" |

Составной databaseCallDef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | Тип подключения, на данный момент возможно один из двух типов подключения:   * function * select * procedure | обязательный |
| sql | JSONString | Описание запроса выборки в формате select.  **Важно!** для postgre селект должен заканчиваться ";"  для всех остальных типов БД нет (без ";") | обязательный, если type равен "select" |
| schema | String255 | Схема подключения | обязательный для function |
| catalog | String255 | Каталог подключения | необязательный |
| functionName | String255 | Имя функции | обязательный, если type равен "function" |
| inParameters | JSONString | Описание входных параметров для функции и процедуры, например  { "\_doc\_num": "VARCHAR",   "\_sta\_con": "VARCHAR",   "\_sta\_txt": "VARCHAR"} | необязательный, используется для "function" и procedure |
| outParameters | JSONString | Описание выходных параметров для функции, например  { "res": "INTEGER"} | необязательный, используется для "function" и procedure |

Составной dataSourceDef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| url | String255 | url для подключения | обязательный |
| className | String255 | Тип драйвера (БД)  oracle.jdbc.OracleDriver - Oracle  org.postgresql.Driver - postgre  com.microsoft.sqlserver.jdbc.SQLServerDriver - MsSQL | обязательный |
| userName | String255 | Имя пользователя | обязательный |
| userPass | String255 | Пароль пользователя | обязательный |

### Пример

**Пример**

БД select

| {  "id": "activity-1",  "type": "workflow\_call",  "description": "",  "workflowCall": {  "workflowDef": {  "type": "db\_call",  "details": {  "databaseCallConfig": {  "databaseCallDef": {  "type": "select",  "sql": "select \*,\r\n COALESCE(to\_char(public.\"Reestr\_M\".expiration\_date, 'MM-DD-YYYY HH24:MI:SS'), '') AS expiration\_date\_text,\r\n COALESCE(to\_char(public.\"Reestr\_M\".issued\_date, 'MM-DD-YYYY HH24:MI:SS'), '') AS issued\_date\_text\r\nfrom public.\"reestr\_MVS\"\r\nwhere public.\"Reestr\_M\".\"SNILS\"=:SNILS \r\nand public.\"Reestr\_M\".\"status\"= :status \r\nand public.\"Reestr\_M\".\"form\"= :form \r\nand public.\"Reestr\_M\".\"number\" IS NOT NULL\r\nand public.\"Reestr\_M\".\"type\" IS NOT NULL\r\nand public.\"reestr\_MVS\".\"issued\_date\" IS NOT NULL\r\nand public.\"Reestr\_M\".\"expiration\_date\" IS NOT NULL\r\nand public.\"Reestr\_M\".\"principal\_i\" IS NOT NULL\r\nand public.\"Reestr\_M\".\"stored\_in\" IS NOT NULL;"  },  "dataSourceDef": {  "className": "org.postgresql.Driver",  "url": "jdbc:postgresql://11.11.1.11:5432/test",  "userName": "",  "userPass": ""  }  }  }  },  "args": {  "SNILS": "jp{SNILS}",  "status": "jp{status}",  "form": "jp{form}"  }  }  } |
| --- |

**Code Block** **3 Пример**

БД селект Оракл

            {  
                "id": "activity-1",  
                "type": "workflow\_call",  
                "description": "",  
                "workflowCall": {  
                    "workflowDef": {  
                        "type": "db\_call",  
                        "details": {  
                            "databaseCallConfig": {  
                                "databaseCallDef": {  
                                    "type": "select",  
                                    "sql": "select \* from table(d.pkg\_1c.d001(to\_date('29.01.2025 0:37:48','[dd.mm](http://dd.mm).yyyy hh24:mi:ss')))"  
                                },  
                                "dataSourceDef": {  
                                    "className": "oracle.jdbc.OracleDriver",  
                                    "url": "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS\_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=11.111.0.111)(PORT=1521)))(CONNECT\_DATA=(SERVICE\_NAME=TS)))",  
                                    "userName": "",  
                                    "userPass": ""  
                                }  
                            }  
                        }  
                    }  
                },  
                "transition": null  
            }

БД procedure

            {  
                "id": "activity-1",  
                "type": "workflow\_call",  
                "description": "",  
                "workflowCall": {  
                    "workflowDef": {  
                        "type": "db\_call",  
                        "details": {  
                            "databaseCallConfig": {  
                                "databaseCallDef": {  
                                    "type": "procedure",  
                                    "schema": "bdtest",  
                                    "inParameters": {  
                                        "in": "INTEGER"  
                                    }  
                                },  
                                "dataSourceDef": {  
                                    "className": "org.postgresql.Driver",  
                                    "url": "jdbc:[postgresql://111.11.111.111:5432/postgres](about:blank)",  
                                    "userName": "",  
                                    "userPass": ""  
                                }  
                            }  
                        }  
                    },  
                    "args": {  
                        "in": "text"  
                    },  
                    "failActivityResult": {  
                        "retryStates": [  
                            "RETRY\_STATE\_MAXIMUM\_ATTEMPTS\_REACHED"  
                        ],                        "variables": {  
                            "status": "ERROR"  
                        }  
                    }  
                },  
                "transition": null  
            }

БД function

            {  
                "id": "activity-1",  
                "type": "workflow\_call",  
                "description": "",  
                "workflowCall": {  
                    "workflowDef": {  
                        "type": "db\_call",  
                        "details": {  
                            "databaseCallConfig": {  
                                "databaseCallDef": {  
                                    "type": "function",  
                                    "schema": "bdtest",  
                                    "catalog": "functions",  
                                    "functionName": "functionName",  
                                    "outParameters": {  
                                        "result": "INTEGER"  
                                    },  
                                    "inParameters": {  
                                        "in": "INTEGER"  
                                    }  
                                },  
                                "dataSourceDef": {  
                                    "className": "org.postgresql.Driver",  
                                    "url": "jdbc:[postgresql://111.11.111.111:5432/postgres](about:blank)",  
                                    "userName": "",  
                                    "userPass": ""  
                                }  
                            }  
                        }  
                    },  
                    "args": {  
                        "in": "text"  
                    },  
                    "failActivityResult": {  
                        "retryStates": [  
                            "RETRY\_STATE\_MAXIMUM\_ATTEMPTS\_REACHED"  
                        ],                        "variables": {  
                            "status": "ERROR"  
                        }  
                    }  
                },  
                "transition": null  
            }

## Примитив send\_to\_rabbitmq

Составной Activity

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String255 | ИД activity.  Важно! В рамках одного БП не должно быть activity с одинаковым ИД. | обязательное |
| description | String255 | Описание шага | необязательное |
| transition | String255 | ИД следующего activity.  Если БП или ветка заканчивается на данном activity, то значение будет  "transition": null | обязательное |
| type | String255 | workflow\_call | обязательный |
| workflowCall | Составной WorkflowCall | Описание activity с типом workflow\_call. | обязательный |
| outputFilter | String255 | Описывает трансформацию исходящих данных после завершения activity, используется для переименования пришедших параметров, объявление переменных (в данном случае переменные необходимо в явном виде указать в outputValidateSchema).  На данный момент после запуска activity выходные переменные движок добавляет в общий скоуп и передает в следующую activity, даже если не указан outputFilter. | необязательный |

Составной WorkflowCall

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| args | JsonObject | Аргументы на вход activity | необязательный |
| workflowDef | Составной SimpleWorkflowDefinition | Описание вызываемого подпроцесса. | обязательный, если нет workflowRef |

Составной SimpleWorkflowDefinition

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | send\_to\_rabbitmq | обязательный |
| details | Составной details | Описание деталей вызова | обязательный |

Составной details

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| sendToRabbitmqConfig | Составной sendToRabbitmqConfig | Описание деталей вызова | обязательный |

Составной sendToRabbitmqConfig

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| connectionRef | Составной connectionRef | Ссылка на шаблон с параметрами подключения | обязательный, если нет connectionDef |
| connectionDef | Составной connectionDef | Параметры подключения | обязательный, если нет connectionRef |
| exchange | Srting255 | Обменник | обязательный |
| routingKey | Srting255 | Ключ маршрутизации | обязательный |
| message | Srting255 | Тело сообщения | обязательный |
| messageProperties | Составной | Параметры сообщения, например,  "messageProperties": { "contentType": "application/xml" , "priority": "2"} | обязательный |

Составной connectionRef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String-UUID | id шаблона параметров подключения к Rabbitmq из справочника | обязательный, если нет name |
| name | Srting255 | Наименование шаблона | обязательный ,если нет id |
| version | Int | Версия шаблона | необязательный |
| tenantId | Srting255 | id системы которая использует шаблон | необязательный, по умолчанию "default" |

Составной connectionDef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| userName | Srting255 | Имя пользователя | обязательный |
| userPass | Srting255 | Пароль пользователя | обязательный |
| addresses | Массив адресов | Адрес подключения и порт | обязательный |
| virtualHost | Srting255 | Виртуальный хост | обязательный |

### Пример

**Пример**

| {  "id": "send\_to\_rabbit",  "type": "workflow\_call",  "workflowCall": {  "workflowDef": {  "type": "send\_to\_rabbitmq",  "details": {  "sendToRabbitmqConfig": {  "connectionDef": {  "userName": "userName",  "userPass": "userPass",  "virtualHost": "/",  "addresses": [  "10.2.3.1:5672"  ]  },  "exchange": "amq.direct",  "routingKey": "rk-to-testq",  "message": "jp{xsltTransformResult}",  "messageProperties": {  "contentType": "application/xml"  }  }  }  }  }  } |
| --- |

**Code Block** **4 Пример**

## Примитив send\_to\_kafka

Составной Activity

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String255 | ИД activity.  Важно! В рамках одного БП не должно быть activity с одинаковым ИД. | обязательное |
| description | String255 | Описание шага | необязательное |
| transition | String255 | ИД следующего activity.  Если БП или ветка заканчивается на данном activity, то значение будет  "transition": null | обязательное |
| type | String255 | workflow\_call | обязательный |
| workflowCall | Составной WorkflowCall | Описание activity с типом workflow\_call. | обязательный |
| outputFilter | String255 | Описывает трансформацию исходящих данных после завершения activity, используется для переименования пришедших параметров, объявление переменных (в данном случае переменные необходимо в явном виде указать в outputValidateSchema).  На данный момент после запуска activity выходные переменные движок добавляет в общий скоуп и передает в следующую activity, даже если не указан outputFilter. | необязательный |

Составной WorkflowCall

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| workflowDef | Составной SimpleWorkflowDefinition | Описание вызываемого подпроцесса. | обязательный, если нет workflowRef |

Составной SimpleWorkflowDefinition

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | send\_to\_kafka | обязательный |
| details | Составной details | Описание деталей вызова | обязательный |

Составной details

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| sendToKafkaConfig | Составной sendToKafkaConfig | Описание деталей вызова | обязательный |

Составной sendToKafkaConfig

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| connectionRef | Составной connectionRef | Ссылка на шаблон с параметрами подключения | обязательный, если нет connectionDef |
| connectionDef | Составной connectionDef | Параметры подключения | обязательный, если нет connectionRef |
| topic | Srting255 | Наименование топика | обязательный |
| Key | Srting255 | Ключ | обязательный |
| message | Составной message | Тело сообщения | обязательный |
| messageProperties | Составной | Параметры сообщения, например,  "messageProperties": { "contentType": "application/xml" , "priority": "2"} | обязательный |

Составной connectionRef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String-UUID | id шаблона параметров подключения к кафке из справочника | обязательный, если нет name |
| name | Srting255 | Наименование шаблона | обязательный ,если нет id |
| version | Int | Версия шаблона | необязательный |
| tenantId | Srting255 | id системы которая использует шаблон | необязательный, по умолчанию "default" |

Составной message

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| payload | Srting255 | Тело сообщения | обязательный |

Составной connectionDef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| bootstrapServers | Srting255 | Адрес подключения | обязательный |
| authDef | Составной authDef | Параметры подключения | обязательный |

Составной authDef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | Srting255 | Тип авторизации   * SASL * TLS - подключение mTls   **Важно!** Если нет авторизации, то authDef не нужно отправлять | обязательный, если есть авторизация |
| sasl | Составной sasl | Подключение по sasl | обязательный, если авторизация sasl |
| tls | Составной tls | Подключение по tls | обязательный, если авторизация tls |

Составной sasl

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| protocol | Srting255 | Протокол подключения:  SASL\_SSL  SASL\_PLAINTEXT | обязательный |
| mechanism | Srting255 | Механизм подключения  OAUTHBEARER - только для SASL\_PLAINTEXT  SCRAM-SHA-512 - для SASL\_SSL и SASL\_PLAINTEXT | обязательный |
| username | Srting255 | Логин | обязательный |
| password | Srting255 | Пароль | обязательный |
| sslDef | Составной sslDef | Сертификаты | обязательный для SCRAM-SHA-512 |
| tokenUrl | Srting255 | Url для проверки пользователя | обязательный для OAUTHBEARER |

Составной sslDef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| trustStoreType | Srting255 | Тип сертификата:  PEM | необязательный для OAUTHBEARER |
| trustStoreCertificates | Srting | Тело сертификатов  Важно! Тело сертификата начинается с  "-----BEGIN CERTIFICATE-----\r\n" и заканчивается "\r\n-----END CERTIFICATE-----\r\n" | необязательный для OAUTHBEARER |

Составной tls

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| keyStoreCertificates | Srting | публичный ключ клиента для аутентификации в формате PEM или base64(PEM)  user.crt | обязательный для TLS |
| keyStoreKey | Srting | приватный ключ клиента в формате PEM или base64(PEM)  user.key | обязательный для TLS |
| trustStoreCertificates | Srting | корневой сертификат в формате PEM или base64(PEM)  ca.crt | обязательный для TLS |
| trustStoreType | Srting255 | Тип сертификата:  PEM | обязательный для TLS |

### Пример

**Пример**

SCRAM-SHA-512

| {  "id": "activity-13",  "type": "workflow\_call",  "description": "",  "workflowCall": {  "workflowDef": {  "type": "send\_to\_kafka",  "details": {  "sendToKafkaConfig": {  "topic": "testing",  "key": "jp{sd\_status}",  "message": {  "payload": {  "messageType": "RequestCreated",  "createdAt": "lua{return math.floor(os.time()+0.5)}lua",  "version": "1.0.0",  "transactionID": "jp{$.transactionID}",  "data": {  "applicationID": "",  "serviceID": "jp{$.data.serviceDeskID}",  "eventType": "jp{$.data.eventType}",  "error": {  "code": "jp{$.respCode}",  "description": "Неизвестная ошибка"  }  }  }  },  "connectionDef": {  "bootstrapServers": "bootstrap-kafka.ru:443",  "authDef": {  "type": "SASL",  "sasl": {  "protocol": "SASL\_SSL",  "mechanism": "SCRAM-SHA-512",  "username": "username",  "password": "password",  "sslDef": {  "trustStoreType": "PEM",  "trustStoreCertificates": "-----BEGIN CERTIFICATE-----\nMIIGVjCCBD6gAwIAt\n-----END CERTIFICATE----------BEGIN CERTIFICATE-----\nMIIGVjCCBD6gAwIAt\n-----END CERTIFICATE-----"  }  }  }  }  }  }  }  },  "transition": **null**  } |
| --- |

**Code Block** **5 Пример**

 TLS

{  
    "id": "activity-13",  
    "type": "workflow\_call",  
    "description": "",  
    "workflowCall": {  
        "workflowDef": {  
            "type": "send\_to\_kafka",  
            "details": {  
                "sendToKafkaConfig": {  
                    "topic": "testing",  
                    "key": "jp{sd\_status}",  
                    "message": {  
                        "payload": {  
                            "description": "Тестовое сообщение от НТ",  
                            "field1": "jp{sd\_status}",  
                            "field2": "jp{sd\_body}"  
                        }  
                    },  
                    "connectionDef": {  
                        "bootstrapServers": "11.111.111.11:9094",  
                        "authDef": {  
                            "type": "TLS",  
                            "tls": {  
                                "trustStoreType": "PEM",  
                                "trustStoreCertificates": "",  
                                "keyStoreKey": "",  
                                "keyStoreCertificates": ""  
                            }  
                        }  
                    }  
                }  
            }  
        }  
    },  
    "transition": null  
}

## Примитив send\_to\_s3

Составной Activity

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String255 | ИД activity.  Важно! В рамках одного БП не должно быть activity с одинаковым ИД. | обязательное |
| description | String255 | Описание шага | необязательное |
| transition | String255 | ИД следующего activity.  Если БП или ветка заканчивается на данном activity, то значение будет  "transition": null | обязательное |
| type | String255 | workflow\_call | обязательный |
| workflowCall | Составной WorkflowCall | Описание activity с типом workflow\_call. | обязательный |
| outputFilter | String255 | Описывает трансформацию исходящих данных после завершения activity, используется для переименования пришедших параметров, объявление переменных (в данном случае переменные необходимо в явном виде указать в outputValidateSchema).  На данный момент после запуска activity выходные переменные движок добавляет в общий скоуп и передает в следующую activity, даже если не указан outputFilter. | необязательный |

Составной WorkflowCall

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| args | JsonObject | Аргументы на вход activity | необязательный |
| workflowDef | Составной SimpleWorkflowDefinition | Описание вызываемого подпроцесса. | обязательный, если нет workflowRef |

Составной SimpleWorkflowDefinition

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | send\_to\_s3 | обязательный |
| details | Составной details | Описание деталей вызова | обязательный |

Составной details

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| sendToS3Config | Составной sendToS3Config | Описание деталей вызова | обязательный |

Составной sendToS3Config

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| connectionRef | Составной connectionRef | Ссылка на шаблон с параметрами подключения | обязательный, если нет connectionDef |
| connectionDef | Составной connectionDef | Параметры подключения | обязательный, если нет connectionRef |
| bucket | Srting255 | Бакет | обязательный |
| region | Srting255 | Регион | обязательный |
| s3File | Составной s3File | Параметры файла | обязательный |

Составной connectionRef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String-UUID | id шаблона параметров подключения к Rabbitmq из справочника | обязательный, если нет name |
| name | Srting255 | Наименование шаблона | обязательный ,если нет id |
| version | Int | Версия шаблона | необязательный |
| tenantId | Srting255 | id системы которая использует шаблон | необязательный, по умолчанию "default" |

Составной connectionDef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| endpoint | Srting255 | Адрес подключения | обязательный |
| authDef | Составной authDef | Виртуальный хост | обязательный |

Составной authDef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | Srting255 | accessKey | обязательный |
| accessKeyAuth | Srting255 | Параметры авторизации | обязательный |

Составной authDef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| accessKey | Srting255 | Ключ доступа | обязательный |
| secretKey | Srting255 | Секрет | обязательный |

Составной s3File

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| filePath | Srting255 | Название файла и его расширение | обязательный |
| content | Srting255 | Переменная, из которой будет сохранены данные в тело файла | обязательный |

### Пример

**Пример**

            {  
                "id": "activity-14",  
                "type": "workflow\_call",  
                "description": "",  
                "workflowCall": {  
                    "workflowDef": {  
                        "type": "send\_to\_s3",  
                        "details": {  
                            "sendToS3Config": {  
                                "bucket": "bucket1",  
                                "region": "msk-1",  
                                "s3File": {  
                                    "filePath": "loadTestFile.txt",  
                                    "content": "jp{sd\_body}"  
                                },  
                                "connectionDef": {  
                                    "endpoint": "[https://s3.ru](https://s3.mts-corp.ru)",  
                                    "authDef": {  
                                        "type": "accessKey",  
                                        "accessKeyAuth": {  
                                            "accessKey": "accessKey",  
                                            "secretKey": "secretKey"  
                                        }  
                                    }  
                                }  
                            }  
                        }  
                    }  
                },  
                "transition": "activity-15"  
            }

## Примитив send\_to\_sap

Составной Activity

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String255 | ИД activity.  Важно! В рамках одного БП не должно быть activity с одинаковым ИД. | обязательное |
| description | String255 | Описание шага | необязательное |
| transition | String255 | ИД следующего activity.  Если БП или ветка заканчивается на данном activity, то значение будет  "transition": null | обязательное |
| type | String255 | workflow\_call | обязательный |
| workflowCall | Составной WorkflowCall | Описание activity с типом workflow\_call. | обязательный |
| outputFilter | String255 | Описывает трансформацию исходящих данных после завершения activity, используется для переименования пришедших параметров, объявление переменных (в данном случае переменные необходимо в явном виде указать в outputValidateSchema).  На данный момент после запуска activity выходные переменные движок добавляет в общий скоуп и передает в следующую activity, даже если не указан outputFilter. | необязательный |

Составной WorkflowCall

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| args | JsonObject | Аргументы на вход activity | обязательный |
| workflowDef | Составной SimpleWorkflowDefinition | Описание вызываемого подпроцесса. | обязательный, если нет workflowRef |

Составной SimpleWorkflowDefinition

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | send\_to\_sap | обязательный |
| details | Составной details | Описание деталей вызова | обязательный |

Составной details

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| sendToSapConfig | Составной sendToSapConfig | Описание деталей вызова | обязательный |

Составной sendToSapConfig

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| connectionRef | Составной connectionRef | Ссылка на шаблон с параметрами подключения | обязательный, если нет connectionDef |
| connectionDef | Составной connectionDef | Параметры подключения | обязательный, если нет connectionRef |
| idoc | Srting255 | Параметры отправляемого документа | обязательный |

Составной connectionRef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String-UUID | id шаблона параметров подключения к Sap из справочника | обязательный, если нет name |
| name | Srting255 | Наименование шаблона | обязательный ,если нет id |
| version | Int | Версия шаблона | необязательный |
| tenantId | Srting255 | id системы которая использует шаблон | необязательный, по умолчанию "default" |

Составной connectionDef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| props | Составной | Параметры подключения к SAP | обязательны все параметры подключения |

Составной props

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| jco.client.lang | Srting255 | Язык клиента | обязательный |
| jco.client.passwd | Srting255 | Пароль | обязательный |
| jco.client.user | Srting255 | Логин | обязательный |
| jco.client.sysnr | Int | номер SAP-системы | обязательный |
| jco.destination.pool\_capacity | Int | максимальное количество подключений, которые могут находиться в пуле подключений для destination | обязательный |
| jco.destination.peak\_limit | Int | максимальное количество одновременных подключений для destionation | обязательный |
| jco.client.client | Int | номер клиента в SAP-системе | обязательный |
| jco.client.ashost | Srting255 | Хост | обязательный |

Составной idoc

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| xml | Srting255 | Тело документа в фомате xml. В данном параметре можно указать переменную, в которой сохранен документ в БП. Например,  "xml" : "jp{sap\_xml}" | обязательный |

### Пример

**Пример**

| {  "id": "send\_to\_sap",  "type": "workflow\_call",  "workflowCall": {  "workflowDef": {  "type": "send\_to\_sap",  "details": {  "sendToSapConfig": {  "connectionDef": {  "props": {  "jco.client.lang": "EN",  "jco.client.passwd": "passwd",  "jco.client.sysnr": 10,  "jco.destination.pool\_capacity": 3,  "jco.destination.peak\_limit": 10,  "jco.client.client": 400,  "jco.client.user": "user",  "jco.client.ashost": "t.t.ru"  }  },  "idoc": {  "xml": "jp{sap\_xml}"  }  }  }  }  }  } |
| --- |

**Code Block** **6 Пример**

## Примитив xslt\_transform

Составной Activity

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String255 | ИД activity.  Важно! В рамках одного БП не должно быть activity с одинаковым ИД. | обязательное |
| description | String255 | Описание шага | необязательное |
| transition | String255 | ИД следующего activity.  Если БП или ветка заканчивается на данном activity, то значение будет  "transition": null | обязательное |
| type | String255 | workflow\_call | обязательный |
| workflowCall | Составной WorkflowCall | Описание activity с типом workflow\_call. | обязательный |
| outputFilter | String255 | Описывает трансформацию исходящих данных после завершения activity, используется для переименования пришедших параметров, объявление переменных (в данном случае переменные необходимо в явном виде указать в outputValidateSchema).  На данный момент после запуска activity выходные переменные движок добавляет в общий скоуп и передает в следующую activity, даже если не указан outputFilter. | необязательный |

Составной WorkflowCall

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| args | JsonObject | Аргументы на вход activity | обязательный |
| workflowDef | Составной workflowDef | Описание вызываемого подпроцесса. | обязательный, если нет workflowRef |

Составной workflowDef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | xslt\_transform | обязательный |
| details | Составной details | Описание деталей вызова | обязательный |

Составной details

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| xsltTransformConfig | Составной xsltTransformConfig | Описание деталей вызова | обязательный |

Составной xsltTransformConfig

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| xsltTemplateRef | СоставнойxsltTemplateRef | Ссылка на шаблон трансформации | обязательный, если нет xsltTemplate |
| xsltTransformTargetRef | Составной xsltTransformTargetRef | Ссылка на документ для трансформации | обязательный, если нет xsltTransformTarget |
| xsltTemplate | Srting255 | Шаблон трансформации | обязательный, если нет xsltTemplateRef |
| xsltTransformTarget | Srting255 | Трансформируемый документ | обязательный, если нет xsltTransformTargetRef |

Составной xsltTemplateRef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String-UUID | id шаблона параметров подключения к Rabbitmq из справочника | обязательный, если нет name |
| name | Srting255 | Наименование шаблона | обязательный ,если нет id |
| version | Int | Версия шаблона | необязательный |
| tenantId | Srting255 | id системы которая использует шаблон | необязательный, по умолчанию "default" |

Составной xsltTransformTargetRef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String-UUID | id шаблона параметров трансформации | обязательный, если нет name |
| name | Srting255 | Наименование шаблона | обязательный ,если нет id |
| version | Int | Версия шаблона | необязательный |
| tenantId | Srting255 | id системы которая использует шаблон | необязательный, по умолчанию "default" |

### Пример

**Пример**

| {  "id": "xslt\_transform",  "type": "workflow\_call",  "workflowCall": {  "workflowDef": {  "type": "xslt\_transform",  "details": {  "xsltTransformConfig": {  "xsltTemplate": "<xsl:stylesheet version=\"1.0\" xmlns:xsl=\"http:\/\/www.w3.org\/1999\/XSL\/Transform\" xmlns:foo=\"http:\/\/www.foo.org\/\" xmlns:bar=\"http:\/\/www.bar.org\">\r\n <xsl:template match=\"node()|@\*\">\r\n <xsl:copy>\r\n <xsl:apply-templates select=\"node()|@\*\"\/>\r\n <\/xsl:copy>\r\n <\/xsl:template> \r\n <xsl:template match=\"SENDER\_1C\/text()\">RS52.TVR-\"\u0442\u0435\u043A\u0441\u0442 \u0441\u043A\u043B\u0435\u0439\u043A\u0438\"<\/xsl:template> \r\n<\/xsl:stylesheet>",  "xsltTransformTarget": "jp{idoc}"  }  }  }  }  } |
| --- |

**Code Block** **7 Пример**

            {  
                "id": "activity-1",  
                "type": "workflow\_call",  
                "description": "",  
                "workflowCall": {  
                    "workflowDef": {  
                        "type": "transform",  
                        "details": {  
                            "transformConfig": {  
                                "type": "xml\_to\_json",  
                                "target": {  
                                    "json": "jp{idoc}"  
                                }  
                            },  
                            "outputValidateSchema": {  
                                "type": "object",  
                                "required": [  
                                    "json"  
                                ],  
                                "properties": {  
                                    "json": {  
                                        "type": "object",  
                                        "required": [  
                                            "IDOC"  
                                        ],  
                                        "properties": {  
                                            "IDOC": {  
                                                "type": "object",  
                                                "required": [  
                                                    "EDI\_DC40"  
                                                ]  
                                            }  
                                        }  
                                    }  
                                }  
                            }  
                        }  
                    }  
                },  
                "transition": "activity-2"  
            }

## Примитив transform

Составной Activity

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String255 | ИД activity.  Важно! В рамках одного БП не должно быть activity с одинаковым ИД. | обязательное |
| description | String255 | Описание шага | необязательное |
| transition | String255 | ИД следующего activity.  Если БП или ветка заканчивается на данном activity, то значение будет  "transition": null | обязательное |
| type | String255 | workflow\_call | обязательный |
| workflowCall | Составной WorkflowCall | Описание activity с типом workflow\_call. | обязательный |
| outputFilter | String255 | Описывает трансформацию исходящих данных после завершения activity, используется для переименования пришедших параметров, объявление переменных (в данном случае переменные необходимо в явном виде указать в outputValidateSchema).  На данный момент после запуска activity выходные переменные движок добавляет в общий скоуп и передает в следующую activity, даже если не указан outputFilter. | необязательный |

Составной WorkflowCall

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| args | JsonObject | Аргументы на вход activity | обязательный |
| workflowDef | Составной workflowDef | Описание вызываемого подпроцесса. | обязательный, если нет workflowRef |

Составной workflowDef

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | transform | обязательный |
| details | Составной details | Описание деталей вызова | обязательный |

Составной details

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| transformConfig | Составной transformConfig | Описание деталей вызова | обязательный |

Составной transformConfig

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| type | String255 | Тип трансформации. На данный момент доступно два вида трансформации:   * xml\_to\_json * json\_to\_xml | обязательный |
| target | Составной | Цель трансформации. Например, можно указать переменную, в которой находится xml-документ,  "target": { "idoc\_json": "jp{xsltTransformResult}" }  В данном случае указано, что переменная idoc\_json будет сохранена в xsltTransformResult | обязательный |

### Пример

**Пример**

| {  "id": "transform",  "type": "workflow\_call",  "workflowCall": {  "workflowDef": {  "type": "transform",  "details": {  "transformConfig": {  "type": "xml\_to\_json",  "target": {  "xml": {  "idoc\_json": "jp{xsltTransformResult}"  }  }  }  }  }  }  } |
| --- |

**Code Block** **8 Пример**

            {  
                "id": "activity-15",  
                "type": "workflow\_call",  
                "description": "",  
                "workflowCall": {  
                    "workflowDef": {  
                        "type": "transform",  
                        "details": {  
                            "transformConfig": {  
                                "type": "xml\_to\_json",  
                                "target": {  
                                    "sap\_json": "jp{sap\_xml}",  
                                    "mis\_json": "jp{mis\_xml}"  
                                }  
                            }  
                        }  
                    }  
                },  
                "transition": "activity-8",  
                "outputFilter": {  
                    "sap\_json\_modified": "lua{\r\nlocal sap = wf.vars.sap\_json; \r\nlocal mis = wf.vars.mis\_json;\r\n\r\nfunction findPerson(persons, uid)\r\n  for k, v in pairs(persons) do\r\n\tif(v.guidPerson == uid) then\r\n\t  return v;\r\n\tend\r\n  end\r\nend \r\nif(mis ~= nil) then\r\n  local newTranz = mis['ТранзакцияНовая'];\r\n  if(newTranz ~= nill) then\r\n    for k, v in pairs(mis['ТранзакцияНовая']) do\r\n      if(v ~= nill) then\r\n        local fiz = v['ФизлицоУИД'];\r\n\t    if(fiz ~= nill) then\r\n          local uid = fiz[''];\r\n          local misSnils = v['СНИЛС'][''];\r\n          local person = findPerson(sap.Person, uid);\r\n          if(person ~= nil) then \r\n\t        person.SNILS = misSnils;\r\n          end\t\r\n\t    end\r\n\t  end\r\n    end  \r\n  end\r\nend\r\nreturn sap;\r\n}lua"  
                }  
            }

## Инъекция inject

Составной Activity

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String255 | ИД activity.  Важно! В рамках одного БП не должно быть activity с одинаковым ИД. | обязательное |
| description | String255 | Описание шага | необязательное |
| transition | String255 | ИД следующего activity.  Если БП или ветка заканчивается на данном activity, то значение будет  "transition": null | обязательное |
| type | String255 | inject | обязательный |
| injectData | Составной | Описание атрибутов для вставки. Например,  "injectData" : { "try\_count" : 0 }  Можно объявлять константы с заданным текстом, переменные из полученных данных схемы, использовать преобразования с помощью lua | обязательный |

### Пример

**Пример**

| {  "id": "init",  "type": "inject",  "description": "Status=INIT",  "injectData": {  "Status": "200",                     "id": "jp{json.IDOC.EDI\_DC40.DOCNUM}",                      "email200":"lua{\r\nfunction sort\_by\_last\_status(data)\r\n local sorted = {} \r\n local result = {ok={}}\r\n for \_, status in ipairs(data.status) do \r\n for \_, item in pairs(status.items) do \r\n local destination = item.destination\r\n local status = item.status \r\n if not sorted[destination] or status > sorted[destination] then\r\n sorted[destination] = status\r\n end\r\n end\r\n end \r\n for e, s in pairs(sorted) do \r\n if s > 199 and s < 300 then\r\n table.insert(result.ok, {email=e})\r\n end\r\n end \r\n return result\r\nend \r\n\r\nif wf.consumedMessages == nil then\r\n return {ok={}}\r\nend\r\nreturn sort\_by\_last\_status(wf.consumedMessages)\r\n}lua"                  },  "transition": "SendingStatuses"  } |
| --- |

**Code Block** **9 Пример**

## Условие switch

Составной Activity

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String255 | ИД activity.  Важно! В рамках одного БП не должно быть activity с одинаковым ИД. | обязательное |
| description | String255 | Описание шага | необязательное |
| type | String255 | switch | обязательный |
| dataConditions | Массив из составных dataConditions | Описание проверяемых условий и действий при успешном их выполнении.  **Важно!**При указании нескольких условий, обратите внимание на порядок их указания. При успешном выполнении первого по счету условия будет совершен переход к указанному в нем transition, не смотря на то, что может подходить еще одно из условий. | обязательное, если type = "switch" |
| defaultTransition | Составной DefaultDataTransition | Описание поведения WF, если результат вызова всех condition = false. | обязательное, если type = "switch" |

Составной dataConditions

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| condition | String400 | Скрипт условия в формате lua.  Например: lua{return next(wf.vars.email200.ok) ~= nil}lua | обязательное |
| conditionDescription | String400 | Описание условия | необязательное |
| transition | String400 | Указывает на activity id, к которому переходит процесс, если результат вызова condition = true.  Если пустое, то означает, что в данной ветке при прохождении condition = true WF будет завершен. | необязательное |

Составной DefaultDataTransition

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| transition | String255 | Указывает на activity id, к которому переходит процесс, если результат вызова всех condition = false.  Если пустое, то означает, что в данной ветке при прохождении condition = false WF будет завершен. | обязательное |
| conditionDescription | String400 | Описание условия | необязательное |

### Пример

**Пример**

| {  "id": "activity-12",  "description": "",  "type": "switch",  "dataConditions": [  {  "transition": "activity-14",  "condition": "lua{return next(wf.vars.email200.ok) ~= nil}lua\r\n",  "conditionDescription": ""  }  ],  "defaultCondition": {  "transition": "activity-13"  }  } |
| --- |

**Code Block** **10 Пример**

4 выхода из условия: 3 условия и 1, если ни одно условие не совпадает

| {  "id": "activity-1",  "type": "switch",  "description": "",  "dataConditions": [  {  "transition": "activity-3",  "condition": "lua{return wf.vars.data.eventType == 'create' and (wf.vars.data.requestType == 'service' or wf.vars.data.requestType == 'access')}lua",  "conditionDescription": ""  },  {  "transition": "activity-12",  "condition": "lua{return wf.vars.data.eventType == 'create' and wf.vars.data.requestType == 'incident'}lua",  "conditionDescription": ""  },  {  "transition": "activity-13",  "condition": "lua{return wf.vars.data.eventType == 'create' and wf.vars.data.requestType == 'inquiry'}lua",  "conditionDescription": ""  }  ],  "defaultCondition": {  "transition": "activity-2",  "conditionDescription": "Unrecognized payload"  }  } |
| --- |

**Code Block** **11 Пример**

## Parallel

Составной Activity

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String255 | ИД activity.  Важно! В рамках одного БП не должно быть activity с одинаковым ИД. | обязательное |
| description | String255 | Описание шага | необязательное |
| transition | String255 | ИД следующего activity.  Если БП или ветка заканчивается на данном activity, то значение будет  "transition": null | обязательное |
| type | String255 | switch | обязательный |
| branches | Массив ИД activity | Список id activity данного WF, которые будут выполняться параллельно. Пример:  "branches": [                     "AR-1-ApiStatusCREATING",                     "AR-2-APIFirewallStatusCREATING"                 ] | обязательное, если type = "**parallel**" |
| completionType | String255 | Тип завершения параллельного activity, может быть следующих видов:  - **anyOf**. Завершение действия parallel возможно, когда завершится хотя бы одно из указанных действий  - **allOf**. Завершение действия parallel возможно, когда завершатся все указанные действия | обязательное, если type = "parallel" |

### Пример

**Пример**

Параллель, где в одной ветке выполняется одно действие AR-1, а во второй AR-2 и AR-3, далее выход из параллели в AR-4

| {  "id": "SendingStatuses",  "type": "parallel",  "branches": [  "AR-1",  "AR-2"  ],  "completionType": "allOf",  "transition": "AR-4"  },  {  "id": "AR-1",  ...  "transition": **null**  },  {  "id": "AR-2",  ...  "transition": "AR-3"  },  {  "id": "AR-3",  ...  "transition": **null**  },  {  "id": "AR-4"  ...             } |
| --- |

**Code Block** **12 Пример**

## Timer

Составной Activity

| **Параметр** | **Тип** | **Описание** | **Обязательность** |
| --- | --- | --- | --- |
| id | String255 | ИД activity.  Важно! В рамках одного БП не должно быть activity с одинаковым ИД. | обязательное |
| description | String255 | Описание шага | необязательное |
| transition | String255 | ИД следующего activity.  Если БП или ветка заканчивается на данном activity, то значение будет  "transition": null | обязательное |
| type | String255 | timer | обязательный |
| timerDuration | String256  ISO 8601 duration format | Таймер, через какое время произойдет переход к следующему activity.  Например:  "PT20.345S" - 20.345 секунд  "PT15M" - 15 минут  "PT10H" - 10 часов  "P2D" - 2 дня  "P2DT3H4M" - 2 дня, 3 часа, 4 минуты | обязательный |

### Пример

**Пример**

| {  "id": "timer0",  "type": "timer",  "description": "timer 1 секунда",  "timerDuration": "PT1S",  "transition": "SD-2-SDAPIpublicationStatus"  } |
| --- |

**Code Block** **13 Пример**

# JS Path

Для задания переменных и указания к ним пути можно использовать JS Path.

Формат:  jp{args.name}

Его можно использовать во всех параметрах схемы. Чаще всего он используется в следующих параметрах:

| **Параметр** | **Тип activity** | **Описание** | **Обязательность использования JS Path** |
| --- | --- | --- | --- |
| outputFilter | rest\_call | Описывает трансформацию исходящих данных после завершения activity, используется для переименования пришедших параметров, объявление переменных (в данном случае переменные необходимо в явном виде указать в outputValidateSchema).  Пример:  "outputFilter": {  "apiName": "jp{body.name}",  "apiVersion": "jp{body.version}",  "apiProtocol": "jp{body.protocol}",  "apiUrl": "jp{body.apiUrl}",  "apiRevisionsId": "jp{body.revisions[0].id}",  "apiRevisionsOwnerLogin": "jp{body.revisions[0].owner.login}",  "apiRevisionsOwnerEmail": "jp{body.revisions[0].owner.email}",  "apiRevisionsOwnerName": "jp{body.revisions[0].owner.name}",  "apiRevisionsOwnerPosition": "jp{body.revisions[0].owner.position}",  "apiRevisionsDescription": "jp{body.revisions[0].description}",  "apiRevisionsSpecificationUrl": "jp{body.revisions[0].specificationUrl}",  "apiRevisionsIms": "jp{body.revisions[0].ims}",  "apiRevisionsPpinfo": "jp{body.revisions[0].ppinfo}",  "apiRevisionsZoneInformation": "jp{body.revisions[0].zoneInformation[\*].contour}",  "APIPublicationData": "jp{body}"  } | необязательное, |
| curl | rest\_call | экранированный curl запроса  Пример:  "curl": "curl -L '<http://wiremock:8080/api/herald/v1/blueprint/378a0ee1-42b5-4762-af8b-7e8b1892556b/execute>' -H 'accept: application/json, text/plain, \*/\*' -H 'content-type: application/json' --data-raw '{\r\n    \"context\": {\r\n        \"receiver\": [\r\n            \"aykomaro5\"\r\n        ],\r\n        \"subscribers\": []\r\n    },\r\n    \"applicationBody\": {\r\n        \"group\": {\r\n            \"groupID\": \"SGP000000085087\",\r\n            \"groupName\": \"КЦ ИТ integration Platform\",\r\n            \"owner\": \"aykomaro5\",\r\n            \"people\": []\r\n        },\r\n        \"comment\": \"Необходимо предоставить сетевой доступ для API rest-demo/1 на портале [https://fw.mts.ru.\\n](https://fw.mts.ru.n) Ссылка на карточку API: \\n Идентификатор ревизии API: jp{revisionId}\\n Протокол: rest \\n  \\n\\n Пошаговая инструкция:\\n 1) Извлечь \\\"Адреса бэкендов API\\\", предоставленные выше\\n 2) Извлечь ip-адреса командой nslookup\\n 2.1) Если порт не указан явно в адресе, например: <http://api.mts-corp.ru:8080>, то порты считаем стандартно: http - 80, https-443.\\n 3) Переходим на [fw.mts.ru](http://fw.mts.ru) -> Создать заявку\\n 4) Выбираем подходящую систему IMS из списка - нас интересует Data plane слой, например, \\\"PFL.007 - Integration Platform - PROD - Data Plane\\\"\\n 5) В описании указываем причину заявки, например:\\n\\\"Доступ необходим в рамках публикации API \\\"rest-demo\\\" в Интеграционной платформе.\\n Ссылка на карточку API: [https://portal.intp-uat.mts-corp.ru/api/154?version=158\\\"\\n](#votjveyy36b1) 6) Создать -> Добавить правило\\n 7) Источник: \\n МТС-Inside\\nIP-адреса: [https://confluence.mts.ru/pages/viewpage.action?pageId=646533052\\n](https://confluence.mts.ru/pages/viewpage.action?pageId=646533052n) Назначение:\\n МТС-Inside (в большинстве случаев МТС-Inside)\\n IP-адреса: полученные с помощью nslookup для \\\"Адреса бэкендов API\\\"\\n Протоколы/порты: 80/443 согласно транспорту, если не задано явно в \\\"Адреса бэкендов API\\\"\\n Описание: \*Дублируем описание заявки\*\\n Срок действия: \\\"бессрочно\\\" для Прод-среды в Промышленной эксплуатации, иначе год.\\n Действие: Добавить\\n 8) Создать -> Отправить \\n 9) В случае необходимости, добавить системы/сети. Обычно определяются автоматически.\\n 10) -> Отправить\\n 11) Открываем карточку созданной заявки, дожидаемся статуса \\\"Согласование\\\"\\n 12) Копируем номер заявки из поля Изменение: RAXXXXXXXXXXXXX\\n 13) Вызываем метод: [https://workflow-engine.intp-dev.mts-corp.ru/api/v1/wf/instance/message\\n](https://workflow-engine.intp-dev.mts-corp.ru/api/v1/wf/instance/messagen) 14) Закрыть настоящую заявку\",\r\n        \"recipient\": {\r\n            \"fullname\": \"Комаров Андрей Юрьевич\",\r\n            \"username\": \"aykomaro5\",\r\n            \"email\": \"aykomaro5@[mts.ru](http://mts.ru)\"\r\n        }\r\n    }\r\n}'" | необязательное |
| url | rest\_call | url подключения  Пример:  ["curl": "http://wiremock:8080/api-request/apis/jp{revisionId}/firewall](http://wiremock:8080/api-request/apis/jp%7BrevisionId%7D/firewall)" | необязательное |
| bodyTemplate | rest\_call | Тело запроса  Пример:  "bodyTemplate": "{\r\n    \"context\": {\r\n        \"receiver\": [\r\n            \"aykomaro5\"\r\n        ],\r\n        \"subscribers\": []\r\n    },\r\n    \"applicationBody\": {\r\n        \"group\": {\r\n            \"groupID\": \"SGP000000085087\",\r\n            \"groupName\": \"КЦ ИТ integration Platform\",\r\n            \"owner\": \"aykomaro5\",\r\n            \"people\": []\r\n        },\r\n        \"comment\": \"Необходимо предоставить сетевой доступ для API rest-demo/1 на портале [https://fw.mts.ru.\\n](https://fw.mts.ru.//n) Ссылка на карточку API: \\n Идентификатор ревизии API: jp{revisionId}\\n Протокол: rest \\n  \\n\\n Пошаговая инструкция:\\n 1) Извлечь \\\"Адреса бэкендов API\\\", предоставленные выше\\n 2) Извлечь ip-адреса командой nslookup\\n 2.1) Если порт не указан явно в адресе, например: [http://api.mts-corp.ru:8080](http://api.mts-corp.ru:8080/), то порты считаем стандартно: http - 80, https-443.\\n 3) Переходим на [fw.mts.ru](http://fw.mts.ru/) -> Создать заявку\\n 4) Выбираем подходящую систему IMS из списка - нас интересует Data plane слой, например, \\\"PFL.007 - Integration Platform - PROD - Data Plane\\\"\\n 5) В описании указываем причину заявки, например:\\n\\\"Доступ необходим в рамках публикации API \\\"rest-demo\\\" в Интеграционной платформе.\\n Ссылка на карточку API: [https://portal.intp-uat.mts-corp.ru/api/154?version=158\\\"\\n](#votjveyy36b1) 6) Создать -> Добавить правило\\n 7) Источник: \\n МТС-Inside\\nIP-адреса: [https://confluence.mts.ru/pages/viewpage.action?pageId=646533052\\n](https://confluence.mts.ru/pages/viewpage.action?pageId=646533052n) Назначение:\\n МТС-Inside (в большинстве случаев МТС-Inside)\\n IP-адреса: полученные с помощью nslookup для \\\"Адреса бэкендов API\\\"\\n Протоколы/порты: 80/443 согласно транспорту, если не задано явно в \\\"Адреса бэкендов API\\\"\\n Описание: \*Дублируем описание заявки\*\\n Срок действия: \\\"бессрочно\\\" для Прод-среды в Промышленной эксплуатации, иначе год.\\n Действие: Добавить\\n 8) Создать -> Отправить \\n 9) В случае необходимости, добавить системы/сети. Обычно определяются автоматически.\\n 10) -> Отправить\\n 11) Открываем карточку созданной заявки, дожидаемся статуса \\\"Согласование\\\"\\n 12) Копируем номер заявки из поля Изменение: RAXXXXXXXXXXXXX\\n 13) Вызываем метод: [https://workflow-engine.intp-dev.mts-corp.ru/api/v1/wf/instance/message\\n](https://workflow-engine.intp-dev.mts-corp.ru/api/v1/wf/instance/message//n) 14) Закрыть настоящую заявку\",\r\n        \"recipient\": {\r\n            \"fullname\": \"Комаров Андрей Юрьевич\",\r\n            \"username\": \"aykomaro5\",\r\n            \"email\": \"aykomaro5@[mts.ru](http://mts.ru/)\"\r\n        }\r\n    }\r\n}'" | необязательное |
| injectData | inject | Описание атрибутов для вставки.  Пример:  "injectData" : { "jp{body.version}" : 0 } | необязательное |
| target | xml\_to\_json | Трансформируемый документ  Пример:  "target": {"idoc\_json": "jp{idoc}"} | необязательное |
| xsltTransformTarget | xslt\_transform | Трансформируемый документ  Пример:  "xsltTransformTarget": "jp{idoc}" | необязательное |
| idoc.xml | send\_to\_sap | Тело документа в фомате xml. В данном параметре можно указать переменную, в которой сохранен документ в БП.  Пример:  "idoc": {"xml": "jp{xsltTransformResult}"} | необязательное |
| message | send\_to\_rabbit | Тело сообщения  Пример:  "message": "jp{xsltTransformResult}" | необязательное |
| args | db\_call | Пример:  "args": {                         "\_doc\_num": "jp{idoc\_json.IDOC.EDI\_DC40.DOCNUM}",                         "\_sta\_con": "jp{idoc\_json.IDOC.EDI\_DC40.SEGMENT}",                         "\_sta\_txt": "jp{idoc\_json.IDOC.EDI\_DC40.IDOCTYP}"                     } | необязательное |

## Пример

**Пример**

| {  "id": "AR-3-APIPublicationData",  "description": "Получение данных о публикации API",  "type": "workflow\_call",  "workflowCall": {  "workflowDef": {  "type": "rest\_call",  "details": {  "inputValidateSchema": {},  "outputValidateSchema": {  "type": "object",  "properties": {  "id": {  "type": "string"  },  "name": {  "type": "string"  },  "version": {  "type": "string"  },  "protocol": {  "type": "string"  },  "revisions": {  "type": "array",  "items": [  {  "type": "object",  "properties": {  "id": {  "type": "string"  },  "owner": {  "type": "object",  "properties": {  "login": {  "type": "string"  },  "email": {  "type": "string"  },  "name": {  "type": "string"  },  "position": {  "type": "string"  }  },  "required": [  "login",  "email",  "name",  "position"  ]  },  "description": {  "type": [  "string",  "null"  ]  },  "specificationUrl": {  "type": "string"  },  "ims": {  "type": [  "string",  "null"  ]  },  "ppinfo": {  "type": [  "string",  "null"  ]  },  "solId": {  "type": [  "string",  "null"  ]  },  "masterContour": {  "type": "object",  "properties": {  "cluster\_id": {  "type": "string"  },  "zone": {  "type": "string"  },  "backend": {  "type": "string"  }  },  "required": [  "cluster\_id",  "zone",  "backend"  ]  },  "zoneInformation": {  "type": "array",  "items": [  {  "type": "object",  "properties": {  "contour": {  "type": "object",  "properties": {  "cluster\_id": {  "type": "string"  },  "zone": {  "type": "string"  },  "backend": {  "type": "string"  }  },  "required": [  "cluster\_id",  "zone",  "backend"  ]  }  },  "required": [  "contour"  ]  }  ]  }  },  "required": [  "id",  "owner",  "specificationUrl",  "masterContour",  "zoneInformation"  ]  }  ]  },  "apiUrl": {  "type": "string"  }  },  "required": [  "id",  "name",  "version",  "protocol",  "revisions",  "apiUrl"  ]  },  "restCallConfig": {  "restCallTemplateDef": {  "curl": "curl --location --request GET 'http://wiremock:8080/api-request/apis/jp{apiId}' \r\nn--header 'Content-Type: application/json'",  "authDef": {  "type": "oauth2",  "oauth2": {  "issuerLocation": "https://isso.mts.ru/auth/realms/mts",  "clientId": "clientId",  "clientSecret": "clientSecret",  "grantType": "client\_credentials"  }  }  }  }  }  }  },  "transition": "CREATING",  "outputFilter": {  "apiName": "jp{body.name}",  "apiVersion": "jp{body.version}",  "apiProtocol": "jp{body.protocol}",  "apiUrl": "jp{body.apiUrl}",  "apiRevisionsId": "jp{body.revisions[0].id}",  "apiRevisionsOwnerLogin": "jp{body.revisions[0].owner.login}",  "apiRevisionsOwnerEmail": "jp{body.revisions[0].owner.email}",  "apiRevisionsOwnerName": "jp{body.revisions[0].owner.name}",  "apiRevisionsOwnerPosition": "jp{body.revisions[0].owner.position}",  "apiRevisionsDescription": "jp{body.revisions[0].description}",  "apiRevisionsSpecificationUrl": "jp{body.revisions[0].specificationUrl}",  "apiRevisionsIms": "jp{body.revisions[0].ims}",  "apiRevisionsPpinfo": "jp{body.revisions[0].ppinfo}",  "apiRevisionsZoneInformation": "jp{body.revisions[0].zoneInformation[\*].contour}",  "APIPublicationData": "jp{body}"  }  } |
| --- |

**Code Block** **14 Пример**

# LUA

Для задания условий и действий можно использовать язык Lua.

Скрипт описывается в формате JsonString lua{-- действие }lua

**Важно!** В lua скрипте **нельзя**обращаться к переменным с помощью JsonPath, вместо этого необходимо указывать прямое обращение к данным.

**Все объявленные в схеме переменные хранятся в wf.vars.**

Он используется в следующих параметрах:

| **Параметр** | **Тип activity** | **Описание** | **Обязательность использования lua** |
| --- | --- | --- | --- |
| condition | switch | Скрипт условия в формате lua  Пример:  lua{local positive = {'executor-reject', 'rejected', 'approvement-reject', 'approvement-deadline', 'revoke', 'solved', 'done', 'closed', 'error'};\r\nlocal function has\_value (tab, val)\r\n    for index, value in ipairs(tab) do\r\n        if value == val then\r\n            return true\r\n        end\r\n    end\r\n    return false\r\nend\r\nreturn wf.vars.try\_count.try\_count > 10 or has\_value(positive, wf.vars.sd\_status)}lua | обязательное, если не указан скрипт в формате spel |
| injectData | inject | Описание атрибутов для вставки. Например,  "injectData": {"try\_count\_n": "lua{return wf.vars.try\_count\_n + 1}lua" } | необязательное |
| url | rest\_call | url подключения  Пример:  [https://web3.ru/rest/integral.lead.addlua{function](https://web3.crm.mts.ru/rest/9621/30beqghnv9s8h1su/mtscrmintegration.siebel.lead.addlua%7bfunction) toValueEnty(path, value, buff)  if type(value) == 'table' then   for k, v in pairs(value) do  toValueEnty(path .. '[' .. k .. ']', v, buff);   end   else      table.insert(buff, wf.urlencode(path) .. '=' .. wf.urlencode(value));  end  end local buff = {}; toValueEnty('', wf.vars.order, buff); return '?fields' ..  table.concat(buff, '&amp;fields'); }lua | необязательное |
| bodyTemplate | rest\_call | Тело запроса  Пример:  "{     "order": {         "specversion": "1.0",         "datacontenttype": "application/json",         "id": "jp{json.IDOC.DOCNUM}",         "source": "/rtk/sap/",         "time": "lua{\r\nCREDAT = wf.vars.json.IDOC.CREDAT\r\nCRETIM = wf.vars.json.IDOC.CRETIM\r\nlocal function safe\_sub(str, start, finish)\r\n\tlocal s = string.sub(str, start, math.min(finish, #str))\r\n\treturn s ~= \"\" and s or \"00\"\r\nend \r\nyear = safe\_sub(CREDAT, 1, 4)\r\nmonth = safe\_sub(CREDAT, 5, 6)\r\nday = safe\_sub(CREDAT, 7, 8) \r\nhour = safe\_sub(CRETIM, 1, 2)\r\nminute = safe\_sub(CRETIM, 3, 4)\r\nsecond = safe\_sub(CRETIM, 5, 6) \r\niso\_date = string.format(\r\n\t'%s-%s-%sT%s:%s:%s.00000Z',\r\n\tyear, month, day,\r\n\thour, minute, second\r\n) \r\nreturn iso\_date \r\n}lua",         "subject": "jp{json.IDOC.DELIVERYORDERID}",         "type": "rtkSap\_orderStatus",         "correlationid": "lua{\n-- Функция для генерации случайного UUID версии 4\nlocal function generateUUID()\n    local random = math.random\n    local template ='xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx'\n    return string.gsub(template, '[xy]', function (c)\n        local v = (c == 'x') and random(0, 15) or random(8, 11)\n        return string.format('%x', v)\n    end)\nend\n\n-- Генерация и возврат UUID\nreturn generateUUID()\n}lua",         "data": {             "deliveryOrderId": "jp{json.IDOC.DELIVERYORDERID}",             "state": "jp{json.IDOC.LNG}"         }     } }" | необязательное |
| injectData | inject | Описание атрибутов для вставки.  Пример:  {"email200":"lua{\r\nfunction sort\_by\_last\_status(data)\r\n    local sorted = {} \r\n  local result = {ok={}}\r\n    for \_, status in ipairs(data.status) do \r\n        for \_, item in pairs(status.items) do \r\n            local destination = item.destination\r\n            local status = item.status       \r\n            if not sorted[destination] or status > sorted[destination] then\r\n                sorted[destination] = status\r\n            end\r\n        end\r\n    end \r\n  for e, s in pairs(sorted) do \r\n    if s > 199 and s < 300 then\r\n      table.insert(result.ok, {email=e})\r\n    end\r\n  end \r\n    return result\r\nend \r\n\r\nif wf.consumedMessages == nil then\r\n  return {ok={}}\r\nend\r\nreturn sort\_by\_last\_status(wf.consumedMessages)\r\n}lua"} | необязательное |
| outputFilter | rest\_call  xml\_to\_json  json\_to\_xml | Объявление переменных на основе полученных данных  Пример:  {"sap\_json\_modified":"lua{\r\nlocal sap = wf.vars.sap\_json; \r\nlocal mis = wf.vars.mis\_json;\r\n\r\nfunction findPerson(persons, uid)\r\n  for k, v in pairs(persons) do\r\n\tif(v.guidPerson == uid) then\r\n\t  return v;\r\n\tend\r\n  end\r\nend \r\nif(mis ~= nil) then\r\n  local newTranz = mis['ТранзакцияНовая'];\r\n  if(newTranz ~= nill) then\r\n    for k, v in pairs(mis['ТранзакцияНовая']) do\r\n      if(v ~= nill) then\r\n        local fiz = v['ФизлицоУИД'];\r\n\t    if(fiz ~= nill) then\r\n          local uid = fiz[''];\r\n          local misSnils = v['СНИЛС'][''];\r\n          local person = findPerson(sap.Person, uid);\r\n          if(person ~= nil) then \r\n\t        person.SNILS = misSnils;\r\n          end\t\r\n\t    end\r\n\t  end\r\n    end  \r\n  end\r\nend\r\nreturn sap;\r\n}lua"} | необязательное |
| message | send\_to\_rabbit  send\_to\_kafka | Тело сообщения  Пример:  "createdAt": "lua{return math.floor(os.time()+0.5)}lua" | необязательное |
| args | db\_call | Пример:  "args": {                         "\_create\_date": ""createdAt": "lua{return math.floor(os.time()+0.5)}lua"",                         "\_sta\_con": "jp{idoc\_json.IDOC.EDI\_DC40.SEGMENT}",                         "\_sta\_txt": "jp{idoc\_json.IDOC.EDI\_DC40.IDOCTYP}"                     } | необязательное |

## Пример

**Пример**

| {  "id": "loop\_switch",  "type": "switch",  "description": "Конечный статус или try\_count > 10?",  "dataConditions": [  {  "condition": "lua{local positive = {'executor-reject', 'rejected', 'approvement-reject', 'approvement-deadline', 'revoke', 'solved', 'done', 'closed', 'error'};\r\nlocal function has\_value (tab, val)\r\n    for index, value in ipairs(tab) do\r\n        if value == val then\r\n            return true\r\n        end\r\n    end\r\n    return false\r\nend\r\nreturn wf.vars.try\_count.try\_count > 10 or has\_value(positive, wf.vars.sd\_status)}lua",  "transition": "map\_status",  "conditionDescription": "sd\_status any of [executor-reject, rejected, approvement-reject, approvement-deadline, revoke, solved, done, closed, error] or try\_count > 10]"  }  ],  "defaultCondition": {  "transition": "inc\_try\_count"  }  } |
| --- |

**Code Block** **15 Пример**

# Примеры

## WF1 Схема содержит основные типы активити, исключая: SAP, send\_to\_sap, await\_for\_message

| {  "type": "complex",  "name": "MaxActivitiShema",  "description": "Схема содержит основные типы активити, исключая: SAP, send\_to\_sap, await\_for\_message",     "details": {  "inputValidateSchema": {},  "starters": [  {  "type": "rest\_call"  }  ]  },  "compiled": {  "activities": [  {  "type": "inject",  "description": "inject try\_count = 0",  "injectData": {  "try\_count": 0,  "applicationId": "35639557-4fb3-4501-ad0b-f45db3356395"  },  "id": "activity-1",  "transition": "activity-2"  },  {  "id": "activity-2",  "type": "workflow\_call",  "description": "Получение статуса executor-solved от SD",  "workflowCall": {  "workflowDef": {  "type": "rest\_call",  "details": {  "restCallConfig": {  "restCallTemplateDef": {  "curl": "curl -L 'https://test.ru/api/v1/wf/search' -H 'accept: \*/\*' -H 'Content-Type: application/json' -d '{\n \"name\": \"MaxActivitiShema\",\n \"offset\": 0,\n \"limit\": 25\n}'",  "authDef": {  "type": "oauth2",  "oauth2": {  "issuerLocation": "https://isso.mts.ru/auth/realms/mts",  "clientId": "clientId",  "clientSecret": "clientSecret",  "grantType": "client\_credentials"  }  }  }  },  "outputValidateSchema": {  "type": "array",  "items": [  {  "type": "object",  "properties": {  "id": {  "type": "string"  },  "type": {  "type": "string"  },  "name": {  "type": "string"  },  "description": {  "type": "string"  },  "tenantId": {  "type": "string"  },  "createTime": {  "type": "string"  },  "changeTime": {  "type": "string"  },  "version": {  "type": "integer"  },  "status": {  "type": "string"  },  "ownerLogin": {  "type": "string"  }  },  "required": [  "id",  "type",  "name",  "description",  "tenantId",  "status"  ]  }  ]  }  }  },  "retryConfig": {}  },  "outputFilter": {  "sd\_body": "jp{$.body}",  "sd\_status": "jp{$.body[0].status}",  "wf\_id": "jp{$.body[0].id}"  },  "transition": "activity-16"  },  {  "id": "activity-11",  "type": "parallel",  "description": "",  "branches": [  "activity-12",  "activity-13"  ],  "completionType": "anyOf",  "transition": "activity-14"  },  {  "id": "activity-12",  "type": "workflow\_call",  "description": "",  "workflowCall": {  "workflowDef": {  "type": "rest\_call",  "details": {  "restCallConfig": {  "restCallTemplateDef": {  "curl": "curl -L 'https://test.ru/api/v1/jp{wf\_id}' -H 'accept: application/json'",  "authDef": {  "type": "oauth2",  "oauth2": {  "issuerLocation": "https://isso.mts.ru/auth/realms/mts",  "clientId": "clientId",  "clientSecret": "clientSecret",  "grantType": "client\_credentials"  }  }  }  }  }  }  },  "transition": **null**  },  {  "id": "activity-13",  "type": "workflow\_call",  "description": "",  "workflowCall": {  "workflowDef": {  "type": "send\_to\_kafka",  "details": {  "sendToKafkaConfig": {  "topic": "testing",  "key": "jp{sd\_status}",  "message": {  "payload": {  "description": "Тестовое сообщение от НТ",  "field1": "jp{sd\_status}",  "field2": "jp{sd\_body}"  }  },  "connectionDef": {  "bootstrapServers": "bootstrap.kafka.ru:443",  "authDef": {  "type": "SASL",  "sasl": {  "protocol": "SASL\_SSL",  "mechanism": "SCRAM-SHA-512",  "username": "username",  "password": "password",  "sslDef": {  "trustStoreType": "PEM",  "trustStoreCertificates": "-----BEGIN CERTIFICATE-----\nMIIGVjCCBD6g----END CERTIFICATE-----"  }  }  }  }  }  }  }  },  "transition": **null**  },  {  "id": "activity-14",  "type": "workflow\_call",  "description": "",  "workflowCall": {  "workflowDef": {  "type": "send\_to\_s3",  "details": {  "sendToS3Config": {  "bucket": "bucket1",  "region": "ru-1",  "s3File": {  "filePath": "loadTestFile.txt",  "content": "jp{sd\_body}"  },  "connectionDef": {  "endpoint": "https://s3.ru",  "authDef": {  "type": "accessKey",  "accessKeyAuth": {  "accessKey": "accessKey",  "secretKey": "secretKey"  }  }  }  }  }  }  },  "transition": "activity-15"  },  {  "id": "activity-15",  "type": "workflow\_call",  "description": "",  "workflowCall": {  "workflowDef": {  "type": "send\_to\_rabbitmq",  "details": {  "sendToRabbitmqConfig": {  "connectionDef": {  "virtualHost": "/",  "addresses": [  "11.111.111.111:5672"  ],  "userName": "userName",  "userPass": "userPass"  },  "exchange": "amq.direct",  "routingKey": "testq",  "message": "jp{sd\_body}",  "messageProperties": {  "contentType": "application/xml",  "priority": "2",  "contentEncoding": "gzip",  "headers": {  "x-message-ttl": 60000,  "key3": **true**  },  "replyTo": "419519\_replyTo",  "expiration": 419519,  "messageId": "419519\_messageId",  "type": "419519\_type",  "userId": "guest",  "appId": "419519\_appId",  "clusterId": "419519\_clusterId"  }  }  }  }  },  "transition": "activity-17"  },  {  "id": "activity-16",  "type": "workflow\_call",  "description": "",  "workflowCall": {  "workflowDef": {  "type": "db\_call",  "details": {  "databaseCallConfig": {  "databaseCallDef": {  "type": "select",  "sql": "SELECT id, type, name, tenant\_id, description, create\_time, ver, convert\_from(lo\_get(compiled),'UTF8') as convert\_from\_compiled, convert\_from(lo\_get(details),'UTF8') as convert\_from\_details FROM ss\_workflow\_engine.definition d where name ='MaxActivitiShema';"  },  "dataSourceDef": {  "className": "org.postgresql.Driver",  "url": "jdbc:postgresql://ip.test:5432/ss\_workflow\_engine",  "userName": "userName",  "userPass": "userPass"  }  }  }  }  },  "transition": "activity-11"  },  {  "id": "activity-17",  "type": "workflow\_call",  "description": "",  "workflowCall": {  "workflowDef": {  "type": "transform",  "details": {  "transformConfig": {  "type": "json\_to\_xml",  "target": {  "xml": {  "idoc\_json": "jp{sd\_body}"  }  }  }  }  },  "retryConfig": {}  }  }  ],  "start": "activity-1"  },  "flowEditorConfig": {  "startMetadata": {  "position": {  "x": 34,  "y": 77  }  },  "activityMetadata": {  "activity-1": {  "position": {  "x": 197.5,  "y": 77  },  "mock": {  "data": {}  }  },  "activity-2": {  "position": {  "x": 406.5,  "y": 77  },  "mock": {  "data": {  "bodyExample": {},  "bodySchema": {},  "headers": {}  }  },  "ims": [  ""  ]  },  "activity-11": {  "position": {  "x": 824.5,  "y": 77  },  "mock": {  "data": {}  }  },  "activity-12": {  "position": {  "x": 1033.5,  "y": 26  },  "mock": {  "data": {  "bodyExample": {},  "bodySchema": {},  "headers": {}  }  },  "ims": [  ""  ]  },  "activity-13": {  "position": {  "x": 1033.5,  "y": 128  },  "mock": {  "data": {}  },  "ims": [  ""  ]  },  "activity-14": {  "position": {  "x": 1242.5,  "y": 77  },  "mock": {  "data": {}  },  "ims": [  ""  ]  },  "activity-15": {  "position": {  "x": 1451.5,  "y": 77  },  "mock": {  "data": {}  },  "ims": [  ""  ]  },  "activity-16": {  "position": {  "x": 607.6274752475248,  "y": 84.71533197437392  },  "mock": {  "data": {  "resultList": []  }  },  "ims": [  ""  ]  },  "activity-17": {  "position": {  "x": 1665.0660376799588,  "y": 77.01318922121393  },  "mock": {  "data": {}  }  },  "activity-18": {  "position": {  "x": 1896.2057296248076,  "y": 78.06951389473056  },  "mock": {  "data": {}  }  }  },  "horizontalLayout": **true**  }  } |
| --- |

**Code Block** **16 Декларативный язык (Workflowlanguage)**

## WF 2 json\_to\_xml, rest\_call, xml\_to\_json, send\_to\_sap

{  
    "name": "name",  
    "type": "complex",  
    "description": "",  
    "compiled": {  
        "activities": [  
            {  
                "id": "activity-8",  
                "type": "workflow\_call",  
                "description": "",  
                "workflowCall": {  
                    "workflowDef": {  
                        "type": "transform",  
                        "details": {  
                            "transformConfig": {  
                                "type": "json\_to\_xml",  
                                "target": {  
                                    "sap\_xml\_modified": "jp{sap\_json\_modified}"  
                                }  
                            }  
                        }  
                    }  
                },  
                "transition": "activity-14"  
            },  
            {  
                "id": "activity-13",  
                "type": "workflow\_call",  
                "description": "zup-mis",  
                "workflowCall": {  
                    "workflowDef": {  
                        "type": "rest\_call",  
                        "details": {  
                            "restCallConfig": {  
                                "restCallTemplateDef": {  
                                    "curl": "curl -L '<http://wiremock:8080/zup-mis/123>' -H 'Content-Type: application/json' -d '{\r\n    \"status\": 200,\r\n    \"headers\": {\r\n      \"Content-Type\": \"application/json\"\r\n    }\r\n  }'"  
                                }  
                            },  
                            "outputValidateSchema": {  
                                "type": "string",  
                                "stringFormat": "xml"  
                            }  
                        }  
                    }  
                },  
                "transition": "activity-15",  
                "outputFilter": {  
                    "body": "jp{body.xmI}"  
                }  
            },  
            {  
                "id": "activity-14",  
                "type": "workflow\_call",  
                "description": "Отправка результата",  
                "workflowCall": {  
                    "workflowDef": {  
                        "type": "rest\_call",  
                        "details": {  
                            "restCallConfig": {  
                                "restCallTemplateDef": {  
                                    "method": "POST",  
                                    "bodyTemplate": "jp{sap\_xml\_modified }",  
                                    "url": "<http://wiremock:8080/medialog/post>",  
                                    "headers": {  
                                        "Content-Type": "application/json"  
                                    },  
                                    "authDef": {  
                                        "type": "oauth2",  
                                        "oauth2": {  
                                            "grantType": "client\_credentials"  
                                        }  
                                    }  
                                }  
                            }  
                        }  
                    }  
                }  
            },  
            {  
                "id": "activity-15",  
                "type": "workflow\_call",  
                "description": "",  
                "workflowCall": {  
                    "workflowDef": {  
                        "type": "transform",  
                        "details": {  
                            "transformConfig": {  
                                "type": "xml\_to\_json",  
                                "target": {  
                                    "sap\_json": "jp{sap\_xml}",  
                                    "mis\_json": "jp{mis\_xml}"  
                                }  
                            }  
                        }  
                    }  
                },  
                "transition": "activity-8",  
                "outputFilter": {  
                    "sap\_json\_modified": "lua{\r\nlocal sap = wf.vars.sap\_json; \r\nlocal mis = wf.vars.mis\_json;\r\n\r\nfunction findPerson(persons, uid)\r\n  for k, v in pairs(persons) do\r\n\tif(v.guidPerson == uid) then\r\n\t  return v;\r\n\tend\r\n  end\r\nend \r\nif(mis ~= nil) then\r\n  local newTranz = mis['ТранзакцияНовая'];\r\n  if(newTranz ~= nill) then\r\n    for k, v in pairs(mis['ТранзакцияНовая']) do\r\n      if(v ~= nill) then\r\n        local fiz = v['ФизлицоУИД'];\r\n\t    if(fiz ~= nill) then\r\n          local uid = fiz[''];\r\n          local misSnils = v['СНИЛС'][''];\r\n          local person = findPerson(sap.Person, uid);\r\n          if(person ~= nil) then \r\n\t        person.SNILS = misSnils;\r\n          end\t\r\n\t    end\r\n\t  end\r\n    end  \r\n  end\r\nend\r\nreturn sap;\r\n}lua"  
                }  
            },  
            {  
                "id": "activity-16",  
                "type": "workflow\_call",  
                "description": "Отправка в SAP",  
                "workflowCall": {  
                    "workflowDef": {  
                        "type": "send\_to\_sap",  
                        "details": {  
                            "sendToSapConfig": {  
                                "connectionDef": {  
                                    "props": {  
                                        "jco.client.lang": "EN",  
                                        "jco.client.passwd": "123",  
                                        "jco.client.sysnr": 10,  
                                        "jco.destination.pool\_capacity": 3,  
                                        "jco.destination.peak\_limit": 10,  
                                        "jco.client.client": 400,  
                                        "jco.client.user": "user",  
                                        "jco.client.ashost": "[test.ru](http://msk-02sprpts.tsretail.ru)"  
                                    }  
                                },  
                                "idoc": {  
                                    "xml": "jp{sap\_xml}"  
                                }  
                            }  
                        }  
                    }  
                },  
                "transition": null  
            },  
            {  
                "id": "activity-17",  
                "type": "parallel",  
                "description": "",  
                "branches": [  
                    "activity-13",  
                    "activity-16"  
                ],  
                "completionType": "anyOf"  
            }  
        ],  
        "start": "activity-17"  
    },  
    "flowEditorConfig": {  
        "startMetadata": {  
            "position": {  
                "x": 34,  
                "y": 104  
            },  
            "isDeveloperMode": false,  
            "isDeveloperModeVerify": false,  
            "isDeveloperModeError": false,  
            "developerModeErrors": []  
        },  
        "activityMetadata": {  
            "activity-8": {  
                "position": {  
                    "x": 824.5,  
                    "y": 42  
                },  
                "mock": {  
                    "data": {}  
                },  
                "isDeveloperMode": false,  
                "isDeveloperModeVerify": false,  
                "isDeveloperModeError": false,  
                "developerModeErrors": []  
            },  
            "activity-13": {  
                "position": {  
                    "x": 406.5,  
                    "y": 42  
                },  
                "mock": {  
                    "data": {  
                        "bodyExample": {},  
                        "bodySchema": {},  
                        "headers": {}  
                    }  
                },  
                "ims": [  
                    ""  
                ],  
                "isDeveloperMode": false,  
                "isDeveloperModeVerify": false,  
                "isDeveloperModeError": false,  
                "developerModeErrors": []  
            },  
            "activity-14": {  
                "position": {  
                    "x": 1033.5,  
                    "y": 42  
                },  
                "mock": {  
                    "data": {  
                        "bodyExample": {},  
                        "bodySchema": {},  
                        "headers": {}  
                    }  
                },  
                "ims": [  
                    ""  
                ],  
                "isDeveloperMode": false,  
                "isDeveloperModeVerify": false,  
                "isDeveloperModeError": false,  
                "developerModeErrors": []  
            },  
            "activity-15": {  
                "position": {  
                    "x": 615.5,  
                    "y": 42  
                },  
                "mock": {  
                    "data": {}  
                },  
                "isDeveloperMode": false,  
                "isDeveloperModeVerify": false,  
                "isDeveloperModeError": false,  
                "developerModeErrors": []  
            },  
            "activity-16": {  
                "position": {  
                    "x": 406.5,  
                    "y": 166  
                },  
                "mock": {  
                    "data": {}  
                },  
                "ims": [  
                    ""  
                ],  
                "isDeveloperMode": false,  
                "isDeveloperModeVerify": false,  
                "isDeveloperModeError": false,  
                "developerModeErrors": []  
            },  
            "activity-17": {  
                "position": {  
                    "x": 197.5,  
                    "y": 104  
                },  
                "mock": {  
                    "data": {}  
                },  
                "isDeveloperMode": false,  
                "isDeveloperModeVerify": false,  
                "isDeveloperModeError": false,  
                "developerModeErrors": []  
            }  
        },  
        "horizontalLayout": true  
    },  
    "details": {  
        "inputValidateSchema": {  
            "$schema": "[http://json-schema.org/draft-04/schema#](http://json-schema.org/draft-04/schema)",  
            "type": "object",  
            "properties": {  
                "mis\_xml": {  
                    "type": "string",  
                    "stringFormat": "xml"  
                },  
                "sap\_xml": {  
                    "type": "string",  
                    "stringFormat": "xml"  
                },  
                "sap\_idoc": {  
                    "type": "string",  
                    "stringFormat": "xml"  
                }  
            },  
            "required": [  
                "sap\_xml"  
            ]  
        },  
        "starters": [  
            {  
                "type": "rest\_call"  
            }  
        ]  
    }  
}

## WF 3 sap\_inbound

{  
    "name": "SAP-LP",  
    "type": "complex",  
    "description": "",  
    "compiled": {  
        "activities": [  
            {  
                "id": "activity-1",  
                "type": "workflow\_call",  
                "description": "",  
                "workflowCall": {  
                    "workflowDef": {  
                        "type": "transform",  
                        "details": {  
                            "transformConfig": {  
                                "type": "xml\_to\_json",  
                                "target": {  
                                    "json": "jp{idoc}"  
                                }  
                            },  
                            "outputValidateSchema": {  
                                "type": "object",  
                                "required": [  
                                    "json"  
                                ],  
                                "properties": {  
                                    "json": {  
                                        "type": "object",  
                                        "required": [  
                                            "IDOC"  
                                        ],  
                                        "properties": {  
                                            "IDOC": {  
                                                "type": "object",  
                                                "required": [  
                                                    "EDI\_DC40"  
                                                ]  
                                            }  
                                        }  
                                    }  
                                }  
                            }  
                        }  
                    }  
                },  
                "transition": "activity-2"  
            },  
            {  
                "id": "activity-2",  
                "type": "inject",  
                "description": "",  
                "injectData": {  
                    "order": {  
                        "specversion": "1.0",  
                        "datacontenttype": "application/json",  
                        "id": "jp{json.IDOC.EDI\_DC40.DOCNUM}",  
                        "source": "/rtk/sap/",  
                        "time": "lua{\r\nCREDAT = wf.vars.json.IDOC.EDI\_DC40.CREDAT\r\nCRETIM = wf.vars.json.IDOC.EDI\_DC40.CRETIM\r\nlocal function safe\_sub(str, start, finish)\r\n\tlocal s = string.sub(str, start, math.min(finish, #str))\r\n\treturn s ~= \"\" and s or \"00\"\r\nend \r\nyear = safe\_sub(CREDAT, 1, 4)\r\nmonth = safe\_sub(CREDAT, 5, 6)\r\nday = safe\_sub(CREDAT, 7, 8) \r\nhour = safe\_sub(CRETIM, 1, 2)\r\nminute = safe\_sub(CRETIM, 3, 4)\r\nsecond = safe\_sub(CRETIM, 5, 6) \r\niso\_date = string.format(\r\n\t'%s-%s-%sT%s:%s:%s.00000Z',\r\n\tyear, month, day,\r\n\thour, minute, second\r\n) \r\nreturn iso\_date \r\n}lua",  
                        "subject": "jp{json.IDOC.E1ADHDR.E1STATE.ZOEBS\_ALEAUD01.DELIVERYORDERID}",  
                        "type": "rtkSap\_orderStatus",  
                        "correlationid": "lua{\n-- Функция для генерации случайного UUID версии 4\nlocal function generateUUID()\n    local random = math.random\n    local template ='xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx'\n    return string.gsub(template, '[xy]', function (c)\n        local v = (c == 'x') and random(0, 15) or random(8, 11)\n        return string.format('%x', v)\n    end)\nend\n\n-- Генерация и возврат UUID\nreturn generateUUID()\n}lua",  
                        "data": {  
                            "deliveryOrderId": "jp{json.IDOC.E1ADHDR.E1STATE.ZOEBS\_ALEAUD01.DELIVERYORDERID}",  
                            "state": "jp{json.IDOC.E1ADHDR.E1STATE.STAPA4\_LNG}"  
                        }  
                    }  
                },  
                "transition": "activity-3"  
            },  
            {  
                "id": "activity-3",  
                "type": "workflow\_call",  
                "description": "",  
                "workflowCall": {  
                    "workflowDef": {  
                        "type": "send\_to\_kafka",  
                        "details": {  
                            "sendToKafkaConfig": {  
                                "topic": "spa-1",  
                                "key": "jp{wf.businessKey}",  
                                "message": {  
                                    "payload": " jp{order}"  
                                },  
                                "connectionDef": {  
                                    "bootstrapServers": "[bootstrap.kafka.ru](http://bootstrap-dpprod.kafka.intp.mts-corp.ru):443",  
                                    "authDef": {  
                                        "type": "SASL",  
                                        "sasl": {  
                                            "protocol": "SASL\_SSL",  
                                            "mechanism": "OAUTHBEARER",  
                                            "username": "",  
                                            "password": "",  
                                            "sslDef": {  
                                                "trustStoreType": "PEM",  
                                                "trustStoreCertificates": ""  
                                            },  
                                            "tokenUrl": "[https://isso.mts.ru/auth/](https://isso.mts.ru/auth/realms/mts/protocol/openid-connect/token)"  
                                        }  
                                    }  
                                }  
                            }  
                        }  
                    }  
                }  
            }  
        ],  
        "start": "activity-1"  
    },  
    "flowEditorConfig": {  
        "startMetadata": {  
            "position": {  
                "x": 34,  
                "y": 44  
            },  
            "isDeveloperMode": false,  
            "isDeveloperModeVerify": false,  
            "isDeveloperModeError": false,  
            "developerModeErrors": []  
        },  
        "activityMetadata": {  
            "activity-1": {  
                "position": {  
                    "x": 197.5,  
                    "y": 44  
                },  
                "mock": {  
                    "data": {  
                        "result": {}  
                    }  
                },  
                "isDeveloperMode": false,  
                "isDeveloperModeVerify": false,  
                "isDeveloperModeError": false,  
                "developerModeErrors": []  
            },  
            "activity-2": {  
                "position": {  
                    "x": 406.5,  
                    "y": 44  
                },  
                "mock": {  
                    "data": {  
                        "result": {}  
                    }  
                },  
                "isDeveloperMode": false,  
                "isDeveloperModeVerify": false,  
                "isDeveloperModeError": false,  
                "developerModeErrors": []  
            },  
            "activity-3": {  
                "position": {  
                    "x": 615.5,  
                    "y": 44  
                },  
                "mock": {  
                    "data": {  
                        "result": {}  
                    }  
                },  
                "ims": [  
                    ""  
                ],  
                "isDeveloperMode": false,  
                "isDeveloperModeVerify": false,  
                "isDeveloperModeError": false,  
                "developerModeErrors": []  
            }  
        },  
        "horizontalLayout": true  
    },  
    "details": {  
        "inputValidateSchema": {},  
        "starters": [  
            {  
                "type": "sap\_inbound",  
                "sapInbound": {  
                    "inboundDef": {  
                        "name": "sapInbound-SAP-LP",  
                        "connectionDef": {  
                            "name": "sapConnection-SAP-LP",  
                            "props": {  
                                "jco.client.lang": "EN",  
                                "jco.destination.peak\_limit": 10,  
                                "jco.client.client": 400,  
                                "jco.client.sysnr": 10,  
                                "jco.destination.pool\_capacity": 3,  
                                "jco.client.ashost": "[m-1.teat.ru](http://msk-02sprpts.tsretail.ru)",  
                                "jco.client.user": "user",  
                                "jco.client.passwd": "passwd"  
                            }  
                        },  
                        "props": {  
                            "jco.server.gwhost": "/H/test[.ru/S/3310](http://msk-02sprpts.tsretail.ru/S/3310)",  
                            "jco.server.progid": "L\_1",  
                            "jco.server.gwserv": "sap",  
                            "jco.server.connection\_count": 2  
                        }  
                    }  
                }  
            }  
        ]  
    }  
}

## WF 4 Kafka-kafka

{  
    "name": "Kafka-kafka",  
    "type": "complex",  
    "description": "Перекладывание сообщений из кафки в кафку",  
    "compiled": {  
        "activities": [  
            {  
                "id": "activity-1",  
                "type": "workflow\_call",  
                "description": "",  
                "workflowCall": {  
                    "workflowDef": {  
                        "type": "send\_to\_kafka",  
                        "details": {  
                            "sendToKafkaConfig": {  
                                "topic": "45",  
                                "key": "",  
                                "message": {  
                                    "payload": "jp{payload}"  
                                },  
                                "connectionDef": {  
                                    "bootstrapServers": "[bootstrap.kafka.ru](http://bootstrap-dpprod.kafka.intp.mts-corp.ru):443",  
                                    "authDef": {  
                                        "type": "SASL",  
                                        "sasl": {  
                                            "protocol": "SASL\_SSL",  
                                            "mechanism": "OAUTHBEARER",  
                                            "username": "",  
                                            "password": "",  
                                            "sslDef": {  
                                                "trustStoreType": "PEM",  
                                                "trustStoreCertificates": ""  
                                            },  
                                            "tokenUrl": "<https://isso.mts.ru/auth/>"  
                                        }  
                                    }  
                                }  
                            }  
                        }  
                    }  
                }  
            }  
        ],  
        "start": "activity-1"  
    },  
    "flowEditorConfig": {  
        "startMetadata": {  
            "position": {  
                "x": 34,  
                "y": 34  
            },  
            "ims": [  
                "7b598ae7-17c8-416f-a491-3991ff8f5d3a"  
            ],  
            "isDeveloperMode": false,  
            "isDeveloperModeVerify": false,  
            "isDeveloperModeError": false,  
            "developerModeErrors": []  
        },  
        "activityMetadata": {  
            "activity-1": {  
                "position": {  
                    "x": 197.5,  
                    "y": 34  
                },  
                "mock": {  
                    "data": {  
                        "result": {}  
                    }  
                },  
                "ims": [  
                    ""  
                ],  
                "isDeveloperMode": false,  
                "isDeveloperModeVerify": false,  
                "isDeveloperModeError": false,  
                "developerModeErrors": []  
            }  
        },  
        "horizontalLayout": true  
    },  
    "details": {  
        "inputValidateSchema": {  
            "type": "object",  
            "required": [  
                "payload"  
            ],  
            "properties": {  
                "payload": {  
                    "type": "object"  
                }  
            }  
        },  
        "starters": [  
            {  
                "name": "Kafka-kafka-KION",  
                "type": "kafka\_consumer",  
                "kafkaConsumer": {  
                    "topic": "rtk-kion",  
                    "connectionDef": {  
                        "bootstrapServers": "11.111.111.11:9094",  
                        "authDef": {  
                            "type": "TLS",  
                            "tls": {  
                                "trustStoreType": "PEM",  
                                "trustStoreCertificates": "",  
                                "keyStoreKey": "",  
                                "keyStoreCertificates": ""  
                            }  
                        }  
                    },  
                    "payloadValidateSchema": {},  
                    "keyValidateSchema": {},  
                    "headersValidateSchema": {},  
                    "outputTemplate": {  
                        "payload": "jp{payload}"  
                    }  
                }  
            }  
        ]  
    }  
}

## WF 4 Преобразование параметров, пришедших из await\_for\_message

{  
    "name": "Campaign",  
    "type": "complex",  
    "description": "",  
    "compiled": {  
        "activities": [  
            {  
                "id": "activity-4",  
                "type": "workflow\_call",  
                "description": "Статус просмотра",  
                "workflowCall": {  
                    "workflowDef": {  
                        "type": "await\_for\_message",  
                        "details": {  
                            "awaitForMessageConfig": {  
                                "messageName": "status"  
                            }  
                        }  
                    }  
                },  
                "transition": "activity-6"  
            },  
            {  
                "id": "activity-5",  
                "type": "inject",  
                "description": "Обработка статусов",  
                "transition": "activity-12",  
                "injectData": {  
                    "email200": "lua{\r\nfunction sort\_by\_last\_status(data)\r\n    local sorted = {} \r\n  local result = {ok={}}\r\n    for \_, status in ipairs(data.status) do \r\n        for \_, item in pairs(status.items) do \r\n            local destination = item.destination\r\n            local status = item.status       \r\n            if not sorted[destination] or status > sorted[destination] then\r\n                sorted[destination] = status\r\n            end\r\n        end\r\n    end \r\n  for e, s in pairs(sorted) do \r\n    if s > 199 and s < 300 then\r\n      table.insert(result.ok, {email=e})\r\n    end\r\n  end \r\n    return result\r\nend \r\n\r\nif wf.consumedMessages == nil then\r\n  return {ok={}}\r\nend\r\nreturn sort\_by\_last\_status(wf.consumedMessages)\r\n}lua"  
                }  
            },  
            {  
                "id": "activity-6",  
                "type": "timer",  
                "description": "",  
                "timerDuration": "P0DT0H2M0S",  
                "transition": "activity-5"  
            },  
            {  
                "id": "activity-11",  
                "type": "inject",  
                "description": "",  
                "transition": "activity-4",  
                "injectData": {  
                    "subscribers": [  
                        {  
                            "email": "mazuyev1@[mts.ru](http://mts.ru)"  
                        },  
                        {  
                            "email": "mazuyev2@[mts.ru](http://mts.ru)"  
                        },  
                        {  
                            "email": "mazuyev3@[mts.ru](http://mts.ru)"  
                        },  
                        {  
                            "email": "mazuyev4@[mts.ru](http://mts.ru)"  
                        }  
                    ]  
                }  
            },  
            {  
                "id": "activity-12",  
                "description": "",  
                "type": "switch",  
                "dataConditions": [  
                    {  
                        "transition": "activity-14",  
                        "condition": "lua{return next(wf.vars.email200.ok) ~= nil}lua\r\n",  
                        "conditionDescription": ""  
                    }  
                ],  
                "defaultCondition": {  
                    "transition": "activity-13"  
                }  
            },  
            {  
                "id": "activity-13",  
                "type": "inject",  
                "description": "",  
                "injectData": {  
                    "email": "Очистить"  
                }  
            },  
            {  
                "id": "activity-14",  
                "type": "inject",  
                "description": "",  
                "injectData": {  
                    "email": "Ок"  
                }  
            }  
        ],  
        "start": "activity-11"  
    },  
    "flowEditorConfig": {  
        "startMetadata": {  
            "position": {  
                "x": 184,  
                "y": 34  
            },  
            "isDeveloperMode": false,  
            "isDeveloperModeVerify": false,  
            "isDeveloperModeError": false,  
            "developerModeErrors": []  
        },  
        "activityMetadata": {  
            "activity-4": {  
                "position": {  
                    "x": 184,  
                    "y": 306  
                },  
                "mock": {  
                    "data": {  
                        "message": {}  
                    }  
                },  
                "ims": [  
                    ""  
                ],  
                "isDeveloperMode": false,  
                "isDeveloperModeVerify": false,  
                "isDeveloperModeError": false,  
                "developerModeErrors": []  
            },  
            "activity-5": {  
                "position": {  
                    "x": 184,  
                    "y": 574  
                },  
                "mock": {  
                    "data": {  
                        "result": {}  
                    }  
                },  
                "isDeveloperMode": false,  
                "isDeveloperModeVerify": false,  
                "isDeveloperModeError": false,  
                "developerModeErrors": []  
            },  
            "activity-6": {  
                "position": {  
                    "x": 184,  
                    "y": 440  
                },  
                "mock": {  
                    "data": {}  
                },  
                "isDeveloperMode": false,  
                "isDeveloperModeVerify": false,  
                "isDeveloperModeError": false,  
                "developerModeErrors": []  
            },  
            "activity-11": {  
                "position": {  
                    "x": 184,  
                    "y": 162  
                },  
                "mock": {  
                    "data": {  
                        "result": {}  
                    }  
                },  
                "isDeveloperMode": false,  
                "isDeveloperModeVerify": false,  
                "isDeveloperModeError": false,  
                "developerModeErrors": []  
            },  
            "activity-12": {  
                "position": {  
                    "x": 184,  
                    "y": 700  
                },  
                "mock": {  
                    "data": {}  
                },  
                "isDeveloperMode": false,  
                "isDeveloperModeVerify": false,  
                "isDeveloperModeError": false,  
                "developerModeErrors": []  
            },  
            "activity-13": {  
                "position": {  
                    "x": 288.5,  
                    "y": 820  
                },  
                "mock": {  
                    "data": {  
                        "result": {}  
                    }  
                },  
                "isDeveloperMode": false,  
                "isDeveloperModeVerify": false,  
                "isDeveloperModeError": false,  
                "developerModeErrors": []  
            },  
            "activity-14": {  
                "position": {  
                    "x": 79.5,  
                    "y": 820  
                },  
                "mock": {  
                    "data": {  
                        "result": {}  
                    }  
                },  
                "isDeveloperMode": false,  
                "isDeveloperModeVerify": false,  
                "isDeveloperModeError": false,  
                "developerModeErrors": []  
            }  
        },  
        "horizontalLayout": false  
    },  
    "details": {  
        "inputValidateSchema": {},  
        "starters": [  
            {  
                "type": "rest\_call"  
            }  
        ]  
    }  
}