

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](#)
 - [Стартер rest](#)
 - [Примеры](#)
 - [Стартер rabbitmq](#)
 - [Пример](#)
 - [Стартер kafkaConsumer](#)
 - [Примеры](#)
 - [Стартер sapInbound](#)
 - [Пример](#)
 - [Стартер scheduler](#)
 - [Пример](#)
 - [Стартер mail_consumer](#)
 - [Пример](#)
 - [Составной Activity](#)
 - [Примитив await_for_message](#)
 - [Пример](#)
 - [Примитив rest_call](#)
 - [Пример](#)
 - [Примитив db_call](#)

- [Пример](#)
- [Примитив send_to_rabbitmq](#)
 - [Пример](#)
- [Примитив send_to_kafka](#)
 - [Пример](#)
- [Примитив send_to_s3](#)
 - [Пример](#)
- [Примитив send_to_sap](#)
 - [Пример](#)
- [Примитив xslt_transform](#)
 - [Пример](#)
- [Примитив transform](#)
 - [Пример](#)
- [Инъекция inject](#)
 - [Пример](#)
- [Условие switch](#)
 - [Пример](#)
- [Parallel](#)
 - [Пример](#)

- [Timer](#)
 - [Пример](#)
- [JS Path](#)
 - [Пример](#)
- [LUA](#)
 - [Пример](#)
- [Примеры](#)
 - [WF1 Схема содержит основные типы активити, исключая: SAP, send_to_sap, await_for_message](#)
 - [WF 2 json to xml, rest call, xml to json, send to sap](#)
 - [WF 3 sap inbound](#)
 - [WF 4 Kafka-kafka](#)
 - [WF 4 Преобразование параметров, пришедших из await_for_message](#)

1 Описание wf/definition

Параметр	Тип	Описание	Обязательность
type	String255	<p>Тип WF. Чаще всего complex. Составной тип примитива, используется для создания БП из нескольких действий. Выполняется и обрабатывается последовательность действий (activity).</p> <p>так же может быть</p> <ul style="list-style-type: none"> • <code>await_for_message</code>. Ожидание вызова от внешней системы/внешнего БП с заранее заданным текстом. • <code>rest_call</code>. Вызов MC (REST API). • <code>db_call</code>. Вызов функции или выборки из БД. • <code>send_to_rabbitmq</code>. Отправка сообщения в очередь rabbitmq • <code>send_to_sap</code>. Отправка idoc в sap • <code>transform</code>. 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: <ul style="list-style-type: none"> • 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 	обязательное

		<ul style="list-style-type: none"> rabbitmq_consumer - старт схемы производится при вычитывании сообщения из очереди rabbitmq mail_consumer - старт схемы производится при вычитывании всех непрочитанных писем почтового ящика 	
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

1.1 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 <code>jp{wf.initVariables}</code>	обязательное

1.1.1 Примеры

На старте 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#",
    "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#",
        "properties": {
            "apild": {
                "type": "string"
            },
            "revisionId": {
                "type": "string"
            }
        },
        "required": [
            "apild",
            "revisionId"
        ],
        "type": "object"
    }
}

```

1.2 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	<p>Схема валидации сообщения. Стандартная json schema https://www.jsonschemavalidator.net/ с описанием механизма проверки входящих переменных.</p> <p>Важно! Если какие-то параметры не будут соответствовать обязательности, указанной в payloadValidateSchema, то по вычитанному сообщению старта схемы не будет.</p>	необязательный
outputTemplate	JsonObject	<p>Объявление переменных на основе полученных в сообщении данных.</p> <p>Например,</p> <pre>{ "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" } }</pre> <p>Важно! Если объявлены переменные в outputTemplate, то необходимо так же описать details.inputValidateSchema, но уже во вложенном объекте payload:</p> <pre>{ "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" } } } }</pre> <p>Т.е. добавляется объект</p> <pre>{ "type": "object", "required": ["payload"], "properties": { "payload": { <здесь описывается outputTemplate> } } }</pre>	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

1.2.1 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}"
  }
}
]
}
}

```

1.3 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	обязательная

			ое, если старт р kafka_ consu mer
connecti onDef	Сост авно й conn ectio nDef	Параметры подключения	обяза тельн ое, если старт р kafka_ consu mer
payload Validate Schema	Json Objec t	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
keyValid ateSche ma	Json Objec t	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
headers Validate Schema	Json Objec t	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	Json Object	<p>Объявление переменных на основе полученных в сообщении данных.</p> <p>Например,</p> <pre>{ "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" }</pre> <p>Важно! Если объявлены переменные в outputTemplate, то необходимо так же описать details.inputValidateSchema, но уже во вложенном объекте payload:</p> <pre>{ "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" } } }</pre> <p>Т.е. добавляется объект</p> <pre>{ "type": "object", "required": ["payload"], "properties": { "payload": { <здесь описывается outputTemplate> } } }</pre>	необязательный
----------------	-------------	--	----------------

Составной connectionDef для kafka_consumer

Параметр	Тип	Описание	Обязательность
bootstrapServers	String255	Адрес подключения	обязательный
authDef	Составной authDef	Параметры подключения	обязательный

Составной authDef для kafka_consumer

Параметр	Тип	Описание	Обязательность
----------	-----	----------	----------------

type	Srting255	Тип авторизации <ul style="list-style-type: none"> • 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

1.3.1 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: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": {
      "apild": {
        "type": "string"
      },
      "revisionId": {
        "type": "string"
      }
    }
  },
  "required": [
    "apild",
    "revisionId"
  ]
},
"starters": [
  {
    "name": "REST-kafka19081712",
    "type": "kafka_consumer",
    "kafkaConsumer": {
      "topic": "testing",
      "connectionDef": {
        "bootstrapServers": "kafka.ru:443",
        "authDef": {
          "type": "SASL",
          "sasl": {
            "protocol": "SASL_SSL",
            "mechanism": "SCRAM-SHA-512",
            "username": "",
            "password": "",
            "sslDef": {
              "trustStoreType": "PEM",
              "trustStoreCertificates": ""
            }
          }
        }
      }
    }
  }
]

```

```

    }
  },
  "payloadValidateSchema": {
    "properties": {
      "apild": {
        "type": "string"
      },
      "revisionId": {
        "type": "string"
      }
    },
    "required": [
      "apild",
      "revisionId"
    ],
    "type": "object"
  },
  "keyValidateSchema": {},
  "headersValidateSchema": {},
  "outputTemplate": {
    "apild": "jp{payload.apild}",
    "revisionId": "jp{payload.revisionId}",
    "kafkaMessagePayload": "jp{payload}"
  }
}
]
}

```

1.4 Стартер sapInbound

Составной starters

Важно! idoc, пришедший на старт схемы сохраняется в стандартную переменную jr{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 destination	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

1.4.1 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",
        "jco.client.user": "user",
        "jco.client.passwd": "passwd"
    },
    "props": {
        "jco.server.gwhost": "/H/test.ru/S/3310",
        "jco.server.progid": "L_1",
        "jco.server.gwserv": "sap",
        "jco.server.connection_count": 2
    }
}
}
```

1.5 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

1.5.1 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"
  }
}
]
}

```

1.6 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

1.6.1 Пример

```
"details": {
  "starters": [
    {
      "name": "mail_starter",
      "type": "mail_consumer",
```



```
"description": "Тестовый стартер",
"tenantId": "default",
"mailConsumer": {
  "connectionDef": {
    "protocol": "imap",
    "host": "imap.yandex.ru",
    "port": 993,
    "mailAuth": {
      "username": "test@test.test",
      "password": "test"
    }
  },
  "mailFilter": {
    "senders": ["mymail@mail.ru"],
    "subjects": ["topic1", "topic2"],
    "startMailDateTime": "2025-01-01T10:01:23.000+03:00"
  }
}
}
```

1.7 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	<p>ИД следующего activity.</p> <p>Если БП или ветка заканчивается на данном activity, то значение будет "transition": null</p>	обязательное
type	String255	<p>Тип activity, может быть следующих типов:</p> <p>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:</p> <ul style="list-style-type: none"> • <ul style="list-style-type: none"> ○ await_for_message. Ожидание вызова от внешней системы/внешнего БП с заранее заданным текстом. ○ rest_call. Вызов MC (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. <p>2. inject. Вставка констант, принудительное изменение параметров БП или задание определенного поведения БП с помощью инъекций параметров. Можно использовать для начала цикла или тестирования.</p> <p>3. switch. Выполнение условия (если ..., то, иначе...).</p> <p>4. timer. Автоматический запуск через заданный промежуток времени.</p> <p>5. transform. Трансформация xml_to_json или json_to_xml</p> <p>6. parallel. Выполнение в параллели нескольких заданных действий.</p>	обязательное
workflowCall	Составной WorkflowCall	<p>Подробное описание параметров activity с типом workflow_call.</p>	обязательное, если type = "workflow_call" или "inject"

injectData	JsonObject	Описание вставляемых данных в формате Json.	обязательное, если type = "inject"
outputFilter	String255	<p>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).</p> <p>На данный момент после запуска activity выходные переменные движок добавляет в общий скоуп и передает в следующую activity, даже если не указан outputFilter.</p>	необязательное
dataConditions	Составной dataConditions	<p>Описание проверяемых условий и действий при успешном их выполнении. Скрипт условия в формате spel или json path.</p> <p>Например: (spel{#name == 'exit'}) или jp{args.name}</p> <p>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.</p>	обязательное, если type = "switch"
defaultTransition	Составной defaultCondition	Описание поведения WF, если результат вызова всех condition = false.	обязательное, если type = "switch"
branches	Массив ИД activity	<p>Список id activity данного WF, которые будут выполняться параллельно. Пример:</p> <pre>"branches": ["AR-1-ApiStatusCREATING", "AR-2-APIFirewallStatusCREATING"]</pre>	обязательное, если type = "parallel"
completionType	String255	<p>Тип завершения параллельного activity, может быть следующих видов:</p> <ul style="list-style-type: none"> - 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 <code>workflowCall.args : {"businessKey": "spel{#wf.businessKey + '-1'}"}</code>	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	необязательный

		"P2DT3H4M" - 2 дня, 3 часа, 4 минуты	
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"

1.8 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	обязательное

		"transition": null	
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 и в описании activity должны быть равны	обязательное

1.8.1 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

1.9 Примитив rest_call

Составной Activity

Параметр	Тип	Описание	Обязательность
id	String255	ИД activity.	обязательное

		Важно! В рамках одного БП не должно быть activity с одинаковым ИД.	
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	необязательный

		"PT15M" - 15 минут "PT10H" - 10 часов "P2D" - 2 дня "P2DT3H4M" - 2 дня, 3 часа, 4 минуты	
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	<p>Описание условий успешного вызова</p> <p>Можно задать несколько predicate, в каждом из которых описать условия. Если заданы несколько predicate, то при проверке успешности они будут обрабатываться как ИЛИ</p> <p>Пример</p> <pre>"resultHandlers": [{ "predicate": { "respCode": 200, "respValueAnyOf": [{ "jsonPath": "jp{body.status.status.statusId}", "values": ["executor-reject", "rejected", "approval-reject", "approval-deadline", "revoke"] }] } }, { "predicate": { "respCode": 205, "respValueAnyOf": [{ "jsonPath": "jp{body.status.status.statusId}", "values": ["executor-reject", "rejected", "approval-reject", "approval-deadline", "revoke"] }] } }]</pre>	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	<p>Описание значений переменных</p> <p>Можно задать несколько pathValueValidation, в каждом из которых описать условия. Если заданы несколько pathValueValidation, то при проверке успешности они будут обрабатываться как ИЛИ</p> <p>Пример</p> <pre>"respValueAnyOf": [{ "jsonPath": "jp{body.prop1}", "values": ["val"], "and": { "jsonPath": "jp{body.prop2}", "values": ["val2"] }]</pre>	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.	необязательный

		Пример <pre>"respValueAnyOf": [{ "jsonPath": "jp{body.prop1}", "values": ["val"], "and": { "jsonPath": "jp{body.prop2}", "values": ["val2"]</pre>	
--	--	--	--

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	обязательный, если не указаны method, url, headers

		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	
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"

1.9.1 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' -H 'accept: */*' -H 'Content-Type: application/json' -d '{\n  \"name\": \"MaxActiviti\nShema\", \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}"
}
}

```

1.10 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	необязательный

		Важно! При вызове процедуры или функции в args описываются входные параметры. Например,	
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 нет databaseCallRef
dataSourceId	String255	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	String255	Наименование шаблона	обязательный, если нет id

version	Int	Версия шаблона	необязательный
tenantId	String255	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: <ul style="list-style-type: none"> function select procedure 	обязательный
sql	JSONString	Описание запроса выборки в формате select. Важно! для postgres селекты должны 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

1.10.1 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.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",
            "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",
            "userName": "",
            "userPass": ""
          }
        }
      }
    },
    "args": {

```

```

        "in": "text"
    },
    "failActivityResult": {
        "retryStates": [
            "RETRY_STATE_MAXIMUM_ATTEMPTS_REACHED"
        ],
        "variables": {
            "status": "ERROR"
        }
    }
},
"transition": null
}

```

1.11 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	<p>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).</p> <p>На данный момент после запуска activity выходные переменные движок добавляет в общий скоуп и передает в следующую activity, даже если не указан outputFilter.</p>	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

1.11.1 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

1.12 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 топики	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 <ul style="list-style-type: none"> • 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

1.12.1 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-----\nMIIGVjCCBD6gAwIA\n-----END CERTIFICATE-----BEGIN
CERTIFICATE-----\nMIIGVjCCBD6gAwIA\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 HT",
        "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
}

```

1.13 Примитив send_to_s3

Составной Activity

Параметр	Тип	Описание	Обязательность
id	String255	ИД activity.	обязательное

		Важно! В рамках одного БП не должно быть activity с одинаковым ИД.	
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

1.13.1 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",
        "authDef": {
          "type": "accessKey",
          "accessKeyAuth": {
            "accessKey": "accessKey",
            "secretKey": "secretKey"
          }
        }
      }
    }
  }
},
"transition": "activity-15"
}

```

1.14 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	обязательное

		"transition": null	
type	String255	workflow_call	обязательный
workflowCall	Составной WorkflowCall	Описание activity с типом workflow_call.	обязательный
outputFilter	String255	<p>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).</p> <p>На данный момент после запуска activity выходные переменные движок добавляет в общий скоуп и передает в следующую activity, даже если не указан outputFilter.</p>	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 destination	обязательный
jco.client.client	Int	номер клиента в SAP-системе	обязательный
jco.client.ashost	Srting25 5	Хост	обязательный

Составной idoc

Параметр	Тип	Описание	Обязательность
xml	Srting25 5	Тело документа в формате xml. В данном параметре можно указать переменную, в которой сохранен документ в БП. Например, "xml" : "jp{sap_xml}"	обязательный

1.14.1 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

1.15 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"

1.15.1 Пример

Пример

```
{
  "id": "xslt_transform",
  "type": "workflow_call",
  "workflowCall": {
    "workflowDef": {
      "type": "xslt_transform",
      "details": {
        "xsltTransformConfig": {
          "xsltTemplate": "<?xml: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()|@*'V>\r\n    <Vxsl:copy>\r\n    <Vxsl:template> \r\n    <xsl:template
match='SENDER_1C\text()'>RS52.TVR-\u0042\u0043\u0043A\u0041\u0042 \u0041\u0043A\u0043B\u0043\u0043\u0043A\u00438\<Vxsl:template>
\r\n<Vxsl: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"
}

```

1.16 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: <ul style="list-style-type: none"> xml_to_json json_to_xml 	обязательный
target	Составной	Цель трансформации. Например, можно указать переменную, в которой находится xml-документ, <pre>"target": { "idoc_json": "jp{xsltTransformResult}" }</pre> В данном случае указано, что переменная idoc_json будет сохранена в xsltTransformResult	обязательный

1.16.1 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\r\n    pairs(persons) do\r\n      if(v.guidPerson == uid) then\r\n        return v;\r\n      end\r\n    end\r\n  end\r\n  if(mis ~= nil) then\r\n    local newTranz = mis[\"ТранзакцияНовая\"];\r\n    if(newTranz ~= nil) then\r\n      for k, v in pairs(mis[\"ТранзакцияНовая\"]) do\r\n        if(v ~= nil) then\r\n          local fiz = v[\"ФизлицоУИД\"];\r\n          if(fiz ~= nil) 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              person.SNILS = misSnils;\r\n            end\r\n          end\r\n        end\r\n      end\r\n    end\r\n  end\r\n  end\r\nend\r\nreturn sap;\r\n}lua"
}
}
```

1.17 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	обязательное

		"transition": null	
type	String255	inject	обязательный
injectData	Составной	<p>Описание атрибутов для вставки. Например,</p> <p>"injectData" : { "try_count" : 0 }</p> <p>Можно объявлять константы с заданным текстом, переменные из полученных данных схемы, использовать преобразования с помощью lua</p>	обязательный

1.17.1 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

1.18 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.	необязательное

		Если пустое, то означает, что в данной ветке при прохождении condition = true WF будет завершен.	
--	--	--	--

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

1.18.1 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

1.19 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: <pre>"branches": ["AR-1-ApiStatusCREATING", "AR-2-APIFirewallStatusCREATING"]</pre>	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"

1.19.1 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

1.20 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

1.20.1 Пример

Пример

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

Code Block 13 Пример

2 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	<p>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).</p> <p>Пример:</p> <pre>"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}",</pre>	необязательное,

		<pre> "apiRevisionsZoneInformation": "jp{body.revisions[0].zoneInformation[*].contour}", "APIPublicationData": "jp{body}" } </pre>	
curl	rest_call	<p>экранированный curl запроса</p> <p>Пример:</p> <pre> "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\", \"groupName\": \"КЦ ИТ integration Platform\", \"owner\": \"aykomaro5\", \"people\": [], \"comment\": \"Необходимо предоставить сетевой доступ для API rest-demo/1 на портале https://fw.mts.ru 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 -> 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 6) Создать -> Добавить правило\n 7) Источник: \n MTC-Inside\n IP-адреса: https://confluence.mts.ru/pages/viewpage.action?pageId=646533052\n Назначение:\n MTC-Inside (в большинстве случаев MTC-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, ожидаем статуса \"\"Согласование\"\"\n 12) Копируем номер заявки из поля Изменение: RAXXXXXXXXXXXXXX\n 13) Вызываем метод: https://workflow-engine.intp-dev.mts-corp.ru/api/v1/wf/instance/message\n 14) Закрыть </pre>	необязательное

[illegible]

		13) Вызываем метод: https://workflow-engine.intp-dev.mts-corp.ru/api/v1/wf/instance/message 14) Закрыть настоящую заявку", "recipient": { "fullname": "Комаров Андрей Юрьевич", "username": "aykomaro5", "email": "aykomaro5@mts.ru"} }	
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 формате 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}",	необязательное

		<pre>"_sta_con": "jp{idoc_json.IDOC.EDI_DC40.SEGMENT}", "_sta_txt": "jp{idoc_json.IDOC.EDI_DC40.IDOCTYP}" }</pre>	
--	--	---	--

2.1 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"
      ]
    },
    "solld": {
      "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"
        ]
      }
    },
    {
      "type": "object",
      "properties": {
        "contour": {
          "type": "object",
          "properties": {
            "cluster_id": {
              "type": "string"
            },
            "zone": {
              "type": "string"
            },
            "backend": {
              "type": "string"
            }
          }
        },
        "required": [
          "cluster_id",
          "zone",
          "backend"
        ]
      }
    }
  ]
},
"required": [
  "contour"
]

```

```

        "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{apild}' \\r\\n--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

3 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: <pre>lua{local positive = {'executor-reject', 'rejected', 'approval-reject', 'approval-deadline', 'revoke', 'solved', 'done', 'closed', 'error'};\nlocal function has_value (tab, val)\n for index, value in\n ipairs(tab) do\n if value == val then\n return true\n end\n end\n return false\nend\nreturn wf.vars.try_count.try_count > 10 or has_value(positive, wf.vars.sd_status)}lua</pre>	обязательное, если не указан скрипт в формате spel
injectData	inject	Описание атрибутов для вставки. Например, <pre>"injectData": {"try_count_n": "lua{return wf.vars.try_count_n + 1}lua" }</pre>	необязательное
url	rest_call	url подключения Пример: <pre>https://web3.ru/rest/integral.lead.addluafunction 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 = {};\ntoValueEnty("", wf.vars.order, buff); return '?fields' .. table.concat(buff, '&fields'); }lua</pre>	необязательное
bodyTemplate	rest_call	Тело запроса Пример:	необязательное

		<pre> "order": { "specversion": "1.0", "datacontenttype": "application/json", "id": "jpjson.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": "jpjson.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": "jpjson.IDOC.DELIVERYORDERID}", "state": "jpjson.IDOC.LNG}" } } </pre>	
injectData	inject	<p>Описание атрибутов для вставки.</p> <p>Пример:</p> <pre> {"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 </pre>	необязательное

		<pre> wf.consumedMessages == nil then\r\n return {ok={}}\r\nend\r\nreturn sort_by_last_status(wf.consumedMessages)\r\n}lua" </pre>	
outputFilter	rest_call xml_to_json json_to_xml	<p>Объявление переменных на основе полученных данных</p> <p>Пример:</p> <pre> {"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 ~= nil) then\r\n for k, v in pairs(mis["ТранзакцияНовая"]) do\r\n if(v ~= nil) then\r\n local fiz = v["ФизлицоУИД"];\r\n\t if(fiz ~= nil) 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\t person.SNILS = misSnils;\r\n end\r\n\t\t end\r\n\t\t end\r\n end \r\n end\r\nend\r\nend\r\nreturn sap;\r\n}lua"} </pre>	необязательное
message	send_to_rabbit send_to_kafka	<p>Тело сообщения</p> <p>Пример:</p> <pre> "createdAt": "lua{return math.floor(os.time()+0.5)}lua" </pre>	необязательное
args	db_call	<p>Пример:</p> <pre> "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}" } </pre>	необязательное

3.1 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

4 Примеры

4.1 WF1 Схема содержит основные типы активити, исключая: SAP, send_to_sap, await_for_message

```

{
  "type": "complex",
  "name": "MaxActivitiSchema",
  "description": "Схема содержит основные типы активности, 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\":\n  \"MaxActivitiSchema\", \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 HT",
              "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
='MaxActivitiSchema';"
                    }
                }
            }
        }
    }
}
},

```

```

        "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)

4.2 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}"
                }
              }
            }
          }
        }
      },
      {
        "id": "activity-14",
        "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.xml}"
  }
},
{
  "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\treturn v;\r\n\tend\r\nend\r\n\r\nif(mis ~= nil) then\r\n    local newTranz =
mis['ТранзакцияНовая'];\r\n    if(newTranz ~= nil) then\r\n        for k, v in pairs(mis['ТранзакцияНовая']) do\r\n            if(v ~= nil) then\r\n                local fiz =
v['ФизлицоУИД'];\r\n                if(fiz ~= nil) 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                        person.SNILS = misSnils;\r\n                        end\r\n                    end\r\n                end\r\n            end\r\n        end\r\n    end\r\nend\r\n\r\nreturn sap;\r\n}\r\nlua"
        }
    }
}

```

```

},
{
  "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"
            }
          },
          "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#",
    "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"
  }
]

```



```

    }
  ]
}

```

4.3 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"
                ]
            }
        }
    },
    {
        "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\tthour, minute, second\r\n) \r\nreturn
iso_date \r\n}lua",
            "subject": "jp{json.IDOC.E1ADHDR.E1STATE.ZOEB5_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:443",
          "authDef": {
            "type": "SASL",
            "sas": {
              "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": 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",
                            "jco.client.user": "user",
                            "jco.client.passwd": "passwd"
                        }
                    }
                },
                "props": {
                    "jco.server.gwhost": "/H/test.ru/S/3310",
                    "jco.server.progid": "L_1",
                    "jco.server.gwserv": "sap",
                    "jco.server.connection_count": 2
                }
            }
        }
    ]
}
}

```

```

    }
  ]
}

```

4.4 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:443",
                "authDef": {
                  "type": "SASL",
                  "sas": {
                    "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}"
    }
  }
}
]
}

```

4.5 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}\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"
      },
      {
        "email": "mazuyev2@mts.ru"
      },
      {
        "email": "mazuyev3@mts.ru"
      },
      {
        "email": "mazuyev4@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"
    }
  ]
}

```



```
}  
}  
]  
}
```