

**Пользовательская инструкция по программному обеспечению
«Единый сервис процессов и коммуникаций (ЕСПК)»**

1. Термины

Программное обеспечение «Единый сервис процессов и коммуникаций (ЕСПК)» (далее – «ПО», «сервис») – программное обеспечение, которое применяется в сфере автоматизации корпоративных бизнес-процессов и представляет собой единый API сервис для отправки сообщений через разные каналы коммуникации и разные брокеры сообщений.

Приложение-клиент - приложение, принадлежащее информационной системе клиента, являющегося пользователем ПО.

Внешний сервис - брокер сообщений, с помощью которого происходит отправка сообщения клиенту.

Массовая отправка - отправка одного контента-сообщения на список адресатов.

Адресная отправка - отправка одного сообщения на один адресат.

2. Основные принципы сервиса

- Работает в двух разных режимах: массовый, адресный.
- Входной контракт является единым и универсальным для любого типа e-mail.
- Входной контракт содержит бизнес-поля, которые являются технически необязательными для заполнения, однако при их заполнении необходимо соблюдать применимые требования. Для поддержания качества данных по сообщениям в едином хранилище, для удобства разбора инцидентов и построения отчетов настоятельно рекомендуем не игнорировать необязательные поля.

3. Таблица конфигураций

Описание

Для того, чтобы направить сообщение именно на тот провайдер, с которым был заключен контракт приложения-клиента, необходимо внести соответствующие настройки в таблицу конфигураций.

Возвращает настройки провайдера для внешнего приложения. необходимо для возможности определить определенному пользователю-приложению конкретный канал отправки сообщения согласно данному типу сообщений и поводу рассылки. Также используется для приоритезации сообщений конкретного приложения. Если запись найти не удалось для каждого канала должны храниться дефолтные настройки, который будут возвращены хранимой процедурой.

- default для sms
- default для email

ХП работает с таблицами:

MSG_CONFIGS

Уникальный ключ на таблицу: system_code

Наименование поля	Тип данных	Уникальность	Обязательность	Описание
ISN	varchar (36)	Y	Y	Уникальный идентификатор в таблице
SYSTEM_CODE	varchar (255)	N	Y	Идентификатор из JWT (app_code)
MSG_TYPE	varchar (36)	N	Y	Тип коммуникации
REASON_EXT_ID	varchar(36)	N	N	Идентификатор повода (триггера) отправки сообщения внешней системы

TOPIC_KEY	nvarchar (50)	N	Y	Наименование ключа топика в конфигах в Platformeco
PROVIDER_ISN	numeric	N	Y	Идентификатор провайдера таблицы msg_providers
TOPIC_FEEDBACK_KEY	nvarchar (50)	N	N	Наименование ключа в конфигах в Platformeco для топика feedbackN
CREATED	datetime	N	Y	Дата создания записи

PROVIDERS

Наименование поля	Тип	Уникальность	Обязательность	Описание
ISN	numeric	Y	Y	Уникальный идентификатор в таблице
NAME	varchar (255)	Y	Y	Наименование провайдера
ACCOUNT_NAME	varchar (50)	N	N	Наименование аккаунта в провайдере отправки сообщений

4. Дефиниция сервиса отправки сообщений

Цели

В документе описана транспортная часть процесса интеграции для выпуска договора.

Описание

Выявлена необходимость создания универсального сервиса рассылок сообщений для ИТ-системы клиента. Сервис реализуется таким образом, чтобы его можно было вызвать из любой точки системы через API. Сервис рассылок сообщений выполняет функцию отправки сообщений на адрес клиента по какому-то определенному событию (триггеру). Сообщение может быть доставлено по e-mail.

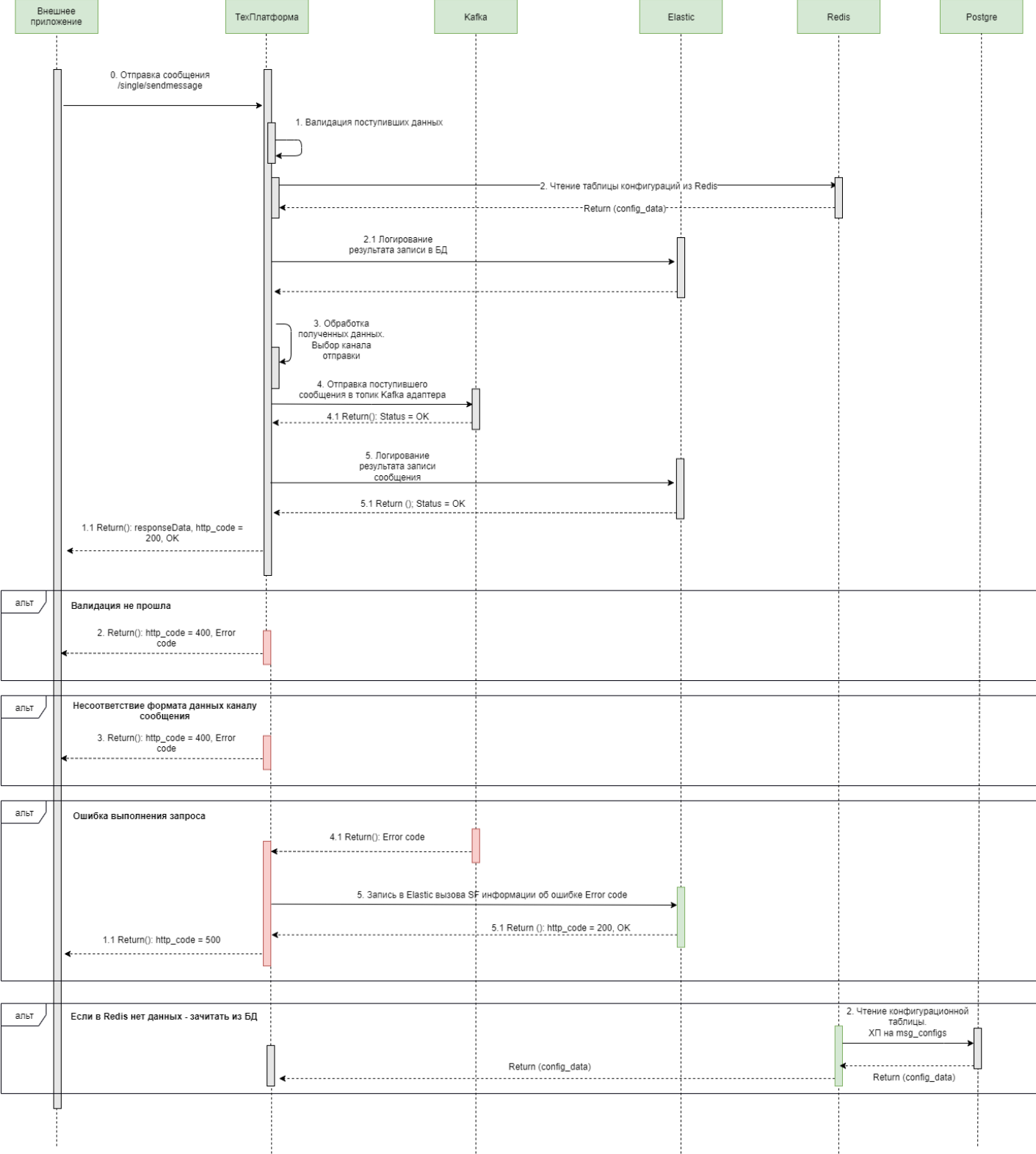
5. Функциональные требования

Алгоритм работы

Потребитель вызывает метод POST `https://.../msg/single/sendmessage`

1. Валидация параметров по формату
 1. Если на этапе валидации произошла ошибка вернуть ошибку с кодом 400. Завершение сценария.
 2. Если валидация прошла успешно, перейти к следующему пункту.
2. Чтение конфигурационной таблицы из redis. Логирование результата записи в Elastic.
 1. Если в кэше Redis не оказалось данных, необходимо зачитать данные из таблицы конфигураций.
 2. Если валидация прошла успешно, перейти к следующему пункту.
3. Выбор канала отправки сообщения: sms/email. Формирование данных для отправки в топик Kafka (топик любого из существующих адаптеров для отправки сообщения).
 1. Если записи нет в таблице конфигураций сгенерировать ошибку с кодом 403. Завершение сценария.
 2. Если формат данных не соответствует типу канала отправки сгенерировать ошибку с кодом 400. Завершение сценария.
 3. Если выбор канала прошел успешно, перейти к следующему пункту.
4. Отправка сообщения в топик Kafka.
 1. Если на этапе записи сообщения в топик Kafka произошла ошибка, ретранслировать ошибку с кодом 500.
5. Запись информации о полученном результате в лог Elastic

Функциональная архитектура



Протокол взаимодействия информационных систем:

- JSON request: [/single/sendmessage](#)
- Получение данных из: json
- Чтение таблицы конфигурация из БД Postgre

Управление жизненным циклом

Запрос события

Https Метод POST: [/single/sendmessage](#)

Номер потока на схеме: 1

Входные данные;

Заголовок:

Формат получаемых данных

Уникальным идентификатором сообщения является ключ uuid.

Блок email должны быть заполнены.

Тело запроса:

Поле json		Тип данных	Обязательность	Пример заполнения	Описание
clientId		number	Нет	912345	Идентификатор клиента
ruleId		number	Нет	23432	Идентификатор продукта (если уместно)
userId		number	Нет	111223	Идентификатор пользователя в системе.
messageId		string	Нет	1234	Идентификатор сообщения во внешней системе
reasonId		string	Нет	12345	Идентификатор повод (внешняя система, dicti справочник). Например, триггерные рассылки по поводу дня рождения.
message		object			Блок для отправки
	address	string	Да	Адрес корпоративной электронной почты	Адрес получателя отправки сообщения
	type	string	Да	e-mail	Тип сообщения.
	sender	string	Нет	default = _____	Адрес отправителя
	title	string	Нет	С днем рождения!	Тема сообщения
	templateId	string	Нет	123445	Шаблон email сообщения

Поле json			Тип данных	Обязательность	Пример заполнения	Описание
		parameters	array	Нет		Массив параметров
		name	string	Да		Наименование параметра
		value	string	Да		Значение параметра
		body	string	Да, если не указан templated	"Здравствуйте! Уважаем(ый/ая) _____! Просим Вас ознакомиться с локальным нормативно-правовым актом"	Текст сообщения (текстовые сообщения всех каналов). Возможно использование html-разметки.
		attachments	array			Массив идентификаторов вложений сообщения
		store	string	Да	"serverName"	Идентификатор хранилища / тома
		identifier	string	Да	"/AISBLOBS /folder1/fileName.ext"	в NFS - путь расположения файла, в других хранилищах (S3) - уникальный идентификатор файла
		fileName	string	Да	"Приложение1.docx"	Имя файла в сообщении
		additionalReceivers	array	Нет		Массив дополнительных получателей сообщения (в копии, альтернативный почтовый адрес)
		address	string	Да	Адрес корпоративной электронной почты	Адрес электронный почты пользователя, указанный при регистрации
		clientId	string	Нет		Идентификатор клиента почтового адреса. Тот редкий случай, когда адрес можно сопоставить с идентификатором клиента, в остальных случаях не указывается (=null).
		addressType	string	Нет	"TO" / "CC" / "BCC" - для email	Тип получателя email сообщения

Тело ответа:

№ п/п	Параметр	Описание
1	code	200
2	description	OK

Тело ответа в случае отсутствия ошибки:

Параметр	Обязат. заполнение	Тип данных	Описание
errorCode	N	string	Код ошибки
errorMessage	N	string	Текст ошибки
errorId	N	string	ID ошибки
internalError	N	string	Внутренняя ошибка

JSONсхема передаваемых данных, включающая в себя коды ошибок:**Спецификация Swagger**

```
{
  "openapi": "3.0.0",
  "info": {
    "version": "1.0",
    "title": "Сервис адресной отправки сообщений",
    "description": "Сервис предназначен для отправки сообщения клиенту по одному из каналов: sms, email, и др. Подходит для триггерных рассылок.",
    "contact": {
      "name": "ФИО",
      "email": "адрес"
    }
  },
  "servers": [
    {
      "url": "https://***",
      "description": "Среда разработки тестовая"
    },
    {
      "url": "https://***/msg",
      "description": "Тестовое окружение (INT)"
    },
    {
      "url": "https://***",
      "description": "Среда разработки бой"
    },
    {
      "url": "https:// ***/msg",
      "description": "Боевое окружение (INT)"
    }
  ],
  "externalDocs": {
    "description": "_____ ",
    "url": "https:// _____ "
  },
  "security": [
    {
      "JWT": []
    }
  ],
  "paths": {
    "/single/sendmessage": {
      "post": {
        "tags": [
          "Отправка сообщения клиенту по EMAIL "
        ],
        "description": "",
        "requestBody": {
          "content": {
```



```
"application/json": {
  "schema": {
    "type": "object",
    "properties": {
      "clientId": {
        "type": "number",
        "description": "Идентификатор клиента"
      },
      "ruleId": {
        "type": "number",
        "description": "Идентификатор продукта"
      },
      "userId": {
        "type": "number",
        "description": "Идентификатор пользователя в системе "
      },
      "messageId": {
        "type": "string",
        "description": "Идентификатор сообщения во внешней системе"
      },
      "reasonId": {
        "type": "string",
        "description": "Идентификатор повод (внешняя система, dicti справочник)"
      },
      "message": {
        "type": "object",
        "required": [
          "body", "address"
        ],
        "description": "Блок для отправки сообщения",
        "properties": {
          "address": {
            "type": "string",
            "description": "Адрес получателя отправки сообщения"
          },
          "type": {
            "type": "string",
            "description": "Тип сообщения: email",
            "enum": [
              "EMAIL"
            ]
          }
        ]
      },
      "sender": {
        "type": "string",
        "description": "Адрес отправителя"
      },
      "title": {
        "type": "string",
        "description": "Тема сообщения (email)"
      },
      "templateId": {
        "type": "string",
        "description": "Шаблон email сообщения"
      },
      "body": {
        "type": "string",
        "description": "Текст сообщения (текстовые сообщения всех каналов)"
      },
      "parameters": {
        "type": "array",
        "description": "Массив параметров",

```

```

        "items": {
            "$ref": "#/components/schemas/Parameter"
        }
    },
    "attachments": {
        "type": "array",
        "description": "Массив идентификаторов вложений сообщения",
        "items": {
            "$ref": "#/components/schemas/Attachment"
        }
    },
    "additionalReceivers": {
        "type": "array",
        "description": "Адреса дополнительных получателей сообщений",
        "items": {
            "$ref": "#/components/schemas/Reciever"
        }
    }
}
}
}
}
}
},
"responses": {
    "200": {
        "description": "OK"
    },
    "400": {
        "description": "Отправлены некорректные входные данные",
        "content": {
            "application/xml": {
                "schema": {
                    "$ref": "#/components/schemas/ResponseDataError"
                }
            }
        }
    },
    "403": {
        "description": "Неверны авторизационные данные, указанные в запросе, или запрещен доступ к запрашиваемому ресурсу",
        "content": {
            "application/xml": {
                "schema": {
                    "$ref": "#/components/schemas/ResponseDataError"
                }
            }
        }
    },
    "500": {
        "description": "Ошибка работы сервиса",
        "content": {
            "application/xml": {
                "schema": {
                    "$ref": "#/components/schemas/ResponseDataError"
                }
            }
        }
    }
}
}
}
}
}

```

```

    }
  },
  "components": {
    "securitySchemes": {
      "JWT": {
        "type": "http",
        "description": "Аутентификация по JWT",
        "scheme": "bearer",
        "bearerFormat": "JWT"
      }
    }
  },
  "schemas": {
    "ResponseDataError": {
      "type": "object",
      "required": [
        "error"
      ],
      "properties": {
        "statusCode": {
          "type": "string",
          "description": "Статус код"
        },
        "code": {
          "type": "string",
          "description": "Код ошибки"
        },
        "error": {
          "type": "string",
          "description": "Расшифровка статус кода"
        },
        "message": {
          "type": "string",
          "description": "Описание ошибки"
        }
      }
    },
    "Reciever": {
      "type": "object",
      "required": [
        "address"
      ],
      "properties": {
        "address": {
          "type": "string",
          "description": "Адрес электронный почты пользователя"
        },
        "clientId": {
          "type": "string",
          "description": "Идентификатор клиента почтового адреса"
        },
        "addressType": {
          "type": "string",
          "description": "Тип получателя (актуально для email сообщения)"
        }
      }
    },
    "Attachment": {
      "type": "object",
      "required": [
        "store",
        "identifier",
        "fileName"
      ]
    }
  }
}

```

```

    },
    "properties": {
      "store": {
        "type": "string",
        "description": "Идентификатор хранилища / тома"
      },
      "identifier": {
        "type": "string",
        "description": "В NFS - путь расположения файла, в других хранилищах (S3) - уникальный идентификатор файла"
      },
      "fileName": {
        "type": "string",
        "description": "Имя файла в сообщении"
      }
    }
  },
  "Parameter": {
    "type": "object",
    "required": [
      "name",
      "value"
    ],
    "properties": {
      "name": {
        "type": "string",
        "description": "Наименование параметра"
      },
      "value": {
        "type": "string",
        "description": "Значение параметра"
      }
    }
  }
}

```

Конфигурационная таблица

Прежде чем сформировать полученное сообщение для отправки в топик Kafka необходимо зачитать настройки провайдера для данного сообщения.

Вызвать sharedFlow по работе с кэшем [Redis](#) и передать следующие параметры:

Наименование поля	Тип данных	Пример заполнения	Описание, источник
system_code	string	"MM"	Уникальный идентификатор из JWT (app_code)
msg_type	string	"email"	Тип сообщения из входного параметра message.type
reason_ext_id	string	12345	Идентификатор повода (триггера) отправки сообщения reasonId

6. Нефункциональные требования к решению

Сервис должен выдерживать нагрузку при обращении более 6,5 млн/сутки.

Время отклика: не более 1000 ms

Ограничение объема передаваемых данных: 3Мб

7. Аутентификация

Аутентификация выполняется через JWT токен на стороне корпоративного oidc.

8. Логирование

Логированию подлежат следующие этапы:

- Обращения к БД и результат обращения
- Обращение и результат записи в Kafka

Дополнительные поля для логирования:

- system_code - идентификатор внешнего приложения, инициирующего отправку сообщения
- messageId- внутренний идентификатор сообщения
- messageExternalId - внешний идентификатор сообщения

Также необходимо в начале дефиниции сохранить тело json запроса, с которым произошло обращение к сервису.

9. Ошибки

Все возникающие ошибки необходимо зафиксировать а логах ElasticSearch.

Код ошибки	Условие	Текст ошибки
400	Если в качестве канала был выбран провайдер SendSay проверить, что в блоке additionalRecievers отсутствуют адреса	"Для типа отправки {type} по каналу {provider_name} можно указать только один адрес получателя."
400	Если во входных параметрах был выбран тип email и в качестве провайдера настроено SMTP, проверить что templateId не заполнен И не заполнен массив параметров.	"Для типа отправки {type} (email) по каналу {provider_name} (ExchangeSMTP) не предусмотрено использование шаблонов templateId и параметров шаблона parameters."
400	Если тип отправки "email" и не указан sender ИЛИ title.	"Для типа отправки {type} (email) необходимо заполнить поле {sender/title}"
403	В таблице MSG_CONGIGS не найдено соответствующей записи для указанного system_code и типа коммуникации type	"Не найдены настройки для отправки сообщений с типом "{type}" для приложения: {system_code}. Необходимо заполнить таблицу конфигураций."
400	Если обнаружено несоответствие расширения имени файла и имени вложения при наличии точки в названии (с конца строки)	"Обнаружено несоответствие расширений в названии файла ({identifier}) и наименовании вложения ({fileName})."
400	Если расширение в файле не соответствует списку разрешенных.	"Обнаружен недопустимый формат расширения {extension} для файла во вложении"
500	Если произошла системная ошибка	"Ошибка сервиса отправки сообщений"

10. Подключение

Для установки и правильной эксплуатации программного обеспечения «Единый сервис процессов и коммуникаций (ЕСПК)», необходимо иметь сервер с установленным на него продуктом Platformeco вендора ООО «Платформеко» (<https://platformeco.ru/>)

Для активации программного обеспечения необходимо обратиться к технической поддержке ООО «Лоджия».

Техническая поддержка:

Пн-Пт, 9:00-18:00 по МСК

Тел.: +7(495) 730-23-20, доб. 1111

Адрес электронной почты: helpdesk@loodsen.ru