



Описание функциональных характеристик
программного обеспечения
«СAB. Универсальная модульная платформа
обработки сервисных запросов»

Екатеринбург, 2020

Термины и определения	3
Назначение системы	4
Функциональные возможности	4
Сервис проверки персональных данных по «черным спискам»	6
Сервис интеграции с веб-сервисом MAIL.RU	12
Сервис проверки данных по базе исполнительных производств.....	14
Сервис интеграции с почтовым шлюзом MS Exchange	17
Сервис интеграции с API Контур.Фокус	19
Сервис работы с POS-терминалами INPAS	20
Сервис проверки инцидентов НСПК	22
Сервис взаимодействия с системой Моё Дело	23
Сервис определения оператора сотовой связи.....	25
Приложение № 1. Структура таблиц базы данных.	27
Приложение № 2. Описание API.....	30

Термин	Определение
ИС	Информационная система
Платформа	САБ. Универсальная модульная платформа обработки сервисных запросов
JMS	стандарт программного обеспечения для рассылки сообщений, позволяющий приложениям, выполненным на платформе Java EE создавать, посылать, получать и читать сообщения
JSON	(англ. <i>JavaScript Object Notation</i> , обычно) — текстовый формат обмена данными, основанный на JavaScript
REST	(от англ. <i>Representational State Transfer</i> - передача состояния представления) архитектурный стиль взаимодействия компонентов распределенного приложения в сети
URL	Адрес (ссылка), указывающий точное местоположение веб-ресурса
XML	eXtensible Markup Language — расширяемый язык разметки — рекомендованный Консорциумом всемирной сети (W3C) язык разметки, представляющий собой свод общих синтаксических правил. XML — текстовый формат, предназначенный для хранения структурированных данных, для обмена информацией между программами.

САБ. Универсальная модульная платформа обработки сервисных запросов (далее по тексту Платформа) предоставляет доступ к данным сторонних информационных систем по различным протоколам взаимодействия (REST, SOAP, JMS и т.д.), а так же позволяет совершать запросы к базе данных Платформы.

Для подключения к ряду сторонних ИС требуется предварительно заключить договоры с компаниями-разработчиками этих систем и получить необходимые параметры подключения к их API.

Для демонстрации работы сервисов на платформе настроены тестовые примеры, которые либо подключаются к тестовым контурам внешних систем, либо эмулируют их работу.

Функциональные возможности

Начать работу с Платформой можно перейдя по ссылке <http://bus-op01-t:9100>

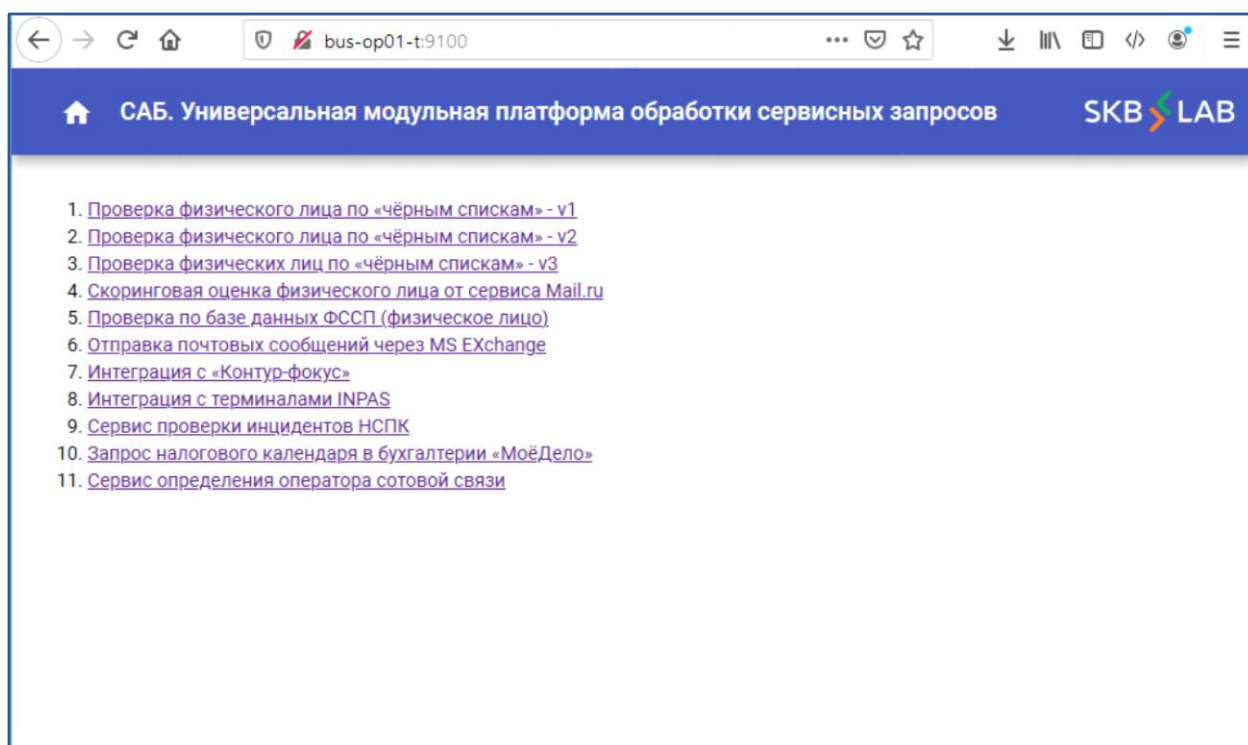


Рисунок 1 Список сервисов Платформы

Выбрав интересующий сервис необходимо заполнить предлагаемую форму и нажать кнопку «Отправить запрос»:

← → ↻ 🏠 bus-op01-t:9100/phone-operator ... 📄 ⌵ 🗨️ ⌂

🏠 САБ. Универсальная модульная платформа обработки сервисных запросов SKB LAB

Сервис определения оператора сотовой связи

Номер телефона *
+79122425567

Формат телефона +71234567889 или 81234567889

Рисунок 2 Заполнение формы данных для запроса

После обработки запроса и обращения к сторонней информационной системе или базе данных платформы на экран будет выведено сообщение с ответом, полученным от сервиса.

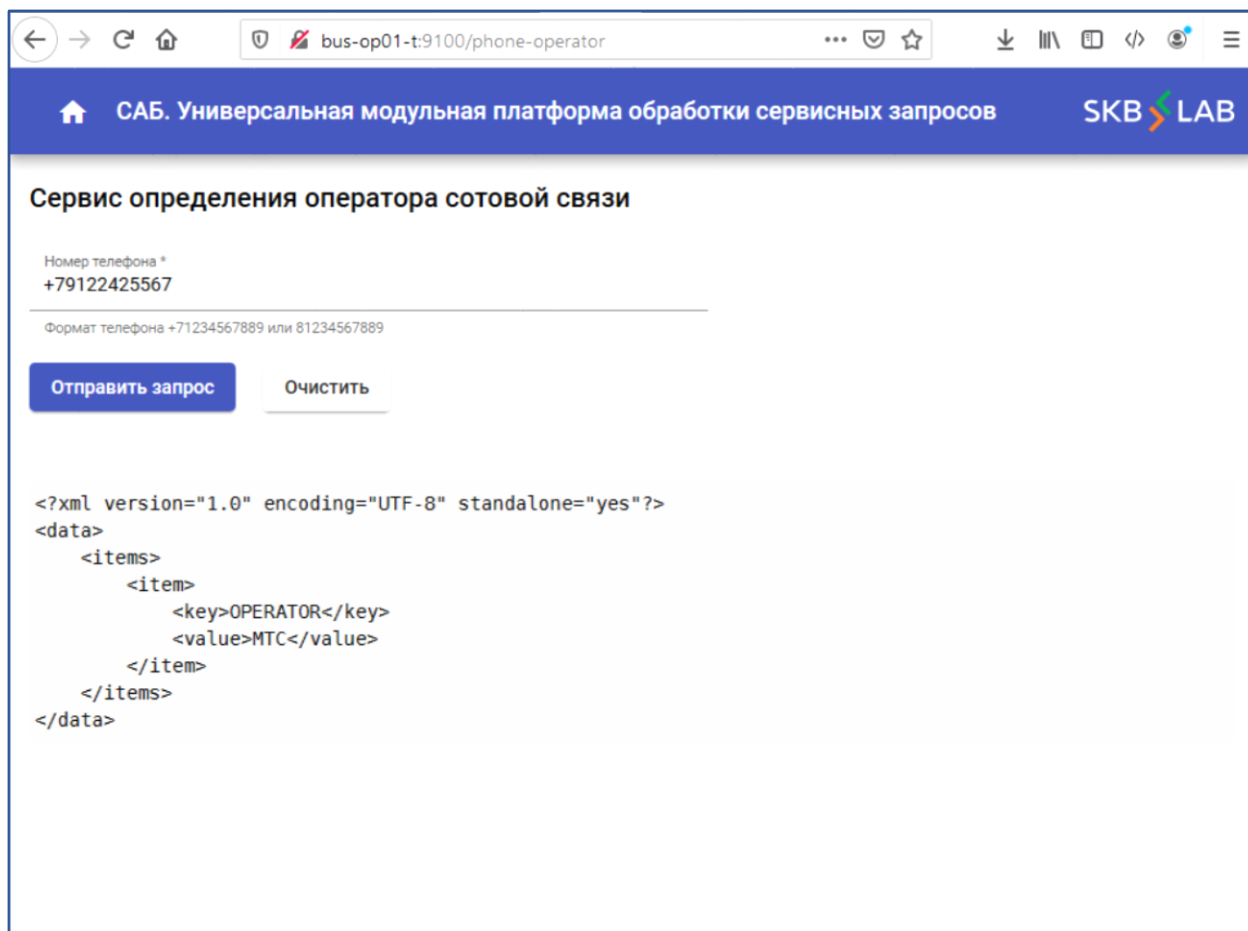


Рисунок 3. Ответ сервиса

Прохождение запросов на всех этапах логируется в базе данных Платформы (таблицы message_log и message_detail_log) и в текстовых файлах.

Для взаимодействия на программном уровне платформа имеет API. Описание API приведено в Приложении № 2.

Сервис проверки персональных данных по «черным спискам»

Сервис предоставляет возможность проверки персональных данных физических лиц по загруженной в базу данных платформы информации о различных негативных стоп-факторах, определяемых в отношении этих лиц.

Для проверки используются следующие таблицы базы данных:

- BL_PASSPORT – списки скомпрометированных паспортов
- BL_ADR_MASS_REG – списки адресов массовой регистрации
- BL_SIGNAL – список стоп-листов организации
- BL_VKI – данные о внутренней кредитной истории физических лиц
- CCL_EFRSB_DECISION_NAMES – список физических лиц - банкротов

Описание структуры таблиц приведено в Приложении № 1.

Проверка может быть произведена в трех вариантах:

- **Проверка физического лица по «Чёрным спискам» -v1:** позволяет осуществить проверку персональных данных физического лица по одному или нескольким факторам, выбранным из списка:
 - проверка по списку скомпрометированных паспортов
 - проверка наличия внутренней негативной кредитной истории и возврат ассоциативных массивов с данными в случае обнаружения
 - проверка наличия адреса регистрации физического лица в списке адресов массовой регистрации по нормализованной таблице
 - проверка наличия адреса проживания физического лица в списке адресов массовой регистрации по нормализованной таблице
 - проверка наличия данных физического лица в списке банкротов
 - проверка данных физического лица на наличие в стоп-листах и возврат балла негатива в случае обнаружения
- **Проверка физического лица по «Чёрным спискам» -v2:** позволяет осуществить проверку персональных данных физического лица по пакету факторов одновременно:
 - проверка по списку скомпрометированных паспортов
 - проверка данных физического лица на наличие внутренней негативной кредитной истории
 - проверка адреса регистрации/проживания физического лица по списку адресов массовой регистрации
 - проверка наличия данных физического лица в списке банкротов
 - проверка данных физического лица на наличие в стоп-листах
 - проверка данных физического лица на наличие в стоп-листах и возврат балла негатива в случае обнаружения
- **Проверка физического лица по «Чёрным спискам» -v3:** позволяет осуществить проверку персональных данных по списку физических лиц по пакету факторов:
 - проверка по списку скомпрометированных паспортов
 - проверка данных физического лица на наличие внутренней негативной кредитной истории
 - проверка адреса регистрации/проживания физического лица по списку адресов массовой регистрации
 - проверка наличия данных физического лица в списке банкротов
 - проверка данных физического лица на наличие в стоп-листах
 - проверка данных физического лица на наличие в стоп-листах и возврат балла негатива в случае обнаружения

Для демонстрации работы сервисов на платформе предусмотрены тестовые примеры. Данные лиц, по которым осуществляется тестовая проверка заполнены как значения по умолчанию в веб-интерфейсе платформы.

Проверка физического лица по «чёрным спискам» - v1.

Для проведения проверки по этому варианту, нужно заполнить данные физического лица и выбрать факторы, по которым должна быть проведена

проверка. Состав данных зависит от факторов, по которым требуется провести проверку. Т.е. в том случае если требуется проверка только по списку скомпрометированных паспортов, то достаточно ввести только ФИО и паспортные данные клиента.

Состав данных физического лица для проверки:

Поле	Примечание
Дата рождения	формат "dd.mm.yyyy"
Имя	
Фамилия	
Отчество	
Номер паспорта	
Серия паспорта	
Область/регион адреса регистрации	Только наименование
Район адреса регистрации	Только наименование
Город адреса регистрации	Только наименование
Населенный пункт адреса регистрации	Только наименование
Улица адреса регистрации	Только наименование
Дом адреса регистрации	
Квартира адреса регистрации	
Регион адреса проживания	Только наименование
Область адреса проживания	Только наименование
Город адреса проживания	Только наименование
Населенный пункт адреса проживания	Только наименование
Улица адреса проживания	Только наименование
Дом адреса проживания	
Квартира адреса проживания	
Факторы проверки	Выбор из списка доступных методов, один или несколько

Ответное сообщение приходит в формате JSON и содержит следующие переменные:

Выполняемая проверка	Возвращаемая переменная
проверка по списку скомпрометированных паспортов	CompPass - целое значение, 0 (не обнаружен) или 1 (найден)
проверка наличия внутренней негативной кредитной истории	VKI - целое значение, 0 или 1 REPORTDATE - максимальная дата из обнаруженных записей в таблице в формате dd.MM.yyyy CODE_ID - перечисление найденных кодов через ','

проверка адреса регистрации физического лица по списку адресов массовой регистрации	MassRegAdr - целое значение, 0 (не обнаружен) или 1 (найден)
проверка адреса проживания физического лица по списку адресов массовой регистрации	MassRegAdr - целое значение, 0 (не обнаружен) или 1 (найден)
проверка наличия данных физического лица в списке банкротов	BANKRUPT_CLIENT - целое значение, 0 (не обнаружен) или 1 (найден)
проверка данных физического лица на наличие в стоп-листах и возврат балла негатива в случае обнаружения	StopListCrit - целое значение, 0 (не обнаружен) или 1 (найден) WEIGHTNEGATIVE - сумма всех найденных полей в списке стоп-листов NEGATIVE_AND_WEIGHTH - перечисление полей значение <NEGATIVE>=<WEIGHTNEGATIVE> через ','

Состав переменных, передаваемых в ответе варьируется в зависимости от методов, по которым требуется осуществить проверку.

Пример ответного сообщения:

```
{
  "valueList": [ {
    "key": "SERVICE_MESSAGE",
    "value": "SUCCESSFULL_BUS_RESPONSE"
  },
  {
    "key": "ReturnText",
    "value": "REQUEST_1"
  },
  {
    "key": "BANKRUPT_CLIENT",
    "value": "0"
  },
  {
    "key": "CompPass",
    "value": "1"
  },
  {
    "key": "MassRegAdr",
    "value": "1"
  },
  {
    "key": "MassResAdr",
    "value": "1"
  },
  {
    "key": "StopListCrit",
    "value": "0"
  },
  {
    "key": "WEIGHTNEGATIVE",
    "value": "0"
  },
  {
    "key": "NEGATIVE AND WEIGHTH",
    "value": ""
  },
  {
    "key": "VKI",
    "value": "0"
  },
  {
    "key": "REPORTDATE",
    "value": "01.03.2020"
  },
  {
    "key": "CODE_ID",
    "value": "4"
  }
]
```

```
}
]
}
```

Проверка физического лица по «чёрным спискам» - v2

Для проведения проверки по этому варианту, нужно заполнить данные одного физического лица:

Поле	Примечание
Имя	
Фамилия	
Отчество	
Номер паспорта	
Серия паспорта	
Регион адреса регистрации	Только наименование
Город адреса регистрации	Только наименование
Улица адреса регистрации	Только наименование
Дом адреса регистрации	
Дата рождения	формат "dd.mm.yyyy"

Ответное сообщение приходит в формате JSON и содержит следующие переменные:

Выполняемая проверка	Возвращаемая переменная
проверка по списку скомпрометированных паспортов	CompPass - целое значение, 0 (не обнаружен) или 1 (найден)
проверка данных физического лица на наличие внутренней негативной кредитной истории	VKI - целое значение, 0 (не обнаружен) или 1 (найден)
проверка адреса регистрации/проживания физического лица по списку адресов массовой регистрации	MassRegAdr - целое значение, 0 (не обнаружен) или 1 (найден)
проверка наличия данных физического лица в списке банкротов	BANKRUPT_CLIENT - целое значение, 0 (не обнаружен) или 1 (найден)
проверка данных физического лица на наличие в стоп-листах	StopListCrit - целое значение, 0 (не обнаружен) или 1 (найден)
проверка данных физического лица на наличие в стоп-листах и возврат балла негатива в случае обнаружения	WEIGHTNEGATIVE - сумма всех найденных полей в стоп-листах NEGATIVE_AND_WEIGHT - перечисление полей значение <NEGATIVE>=<WEIGHTNEGATIVE> через ','

Пример ответного сообщения:

```
{ "valueList": [{ "key": "CompPass", "value": "0" }, { "key": "VKI", "value": "0" }, { "key": "MassRegAd  
r", "value": "0" }, { "key": "StopListCrit", "value": "0" }, { "key": "BANKRUPT_CLIENT", "value": "0"  
}, { "key": "WEIGHTNEGATIVE", "value": "0" }, { "key": "NEGATIVE_AND_WEIGHTH", "value": "  
"" }, { "key": "SERVICE_MESSAGE", "value": "SUCCESSFULL_BUS_RESPONSE" } ] }
```

Проверка физического лица по «чёрным спискам» - v3

Проверка может быть выполнена по списку данных нескольких физических лиц по пакету факторов. Для проверки для каждого лица нужно заполнить данные:

Поле	Примечание
Идентификатор клиента	Порядковый номер
Имя	
Фамилия	
Отчество	
Номер паспорта	
Серия паспорта	
Кем выдан паспорт	
Страна адреса регистрации	В формате ФИАС
Регион адреса регистрации	В формате ФИАС
Область адреса регистрации	В формате ФИАС
Город адреса регистрации	В формате ФИАС
Район адреса регистрации	В формате ФИАС
Улица адреса регистрации	В формате ФИАС
Дом адреса регистрации	
Литера	
Номер блока	
Квартира	
Дата рождения	формат "dd.mm.yyyy"

Ответное сообщение формируется в виде массивов Object (в соответствии с идентификаторами клиентов), в которых возвращаются данные проверки по каждому из физических лиц. Состав переменных ответного сообщения:

Выполняемая проверка	Возвращаемая переменная
проверка по списку скомпрометированных паспортов	CompPass - целое значение, 0 (не обнаружен) или 1 (найден)
проверка данных физического лица на наличие внутренней негативной кредитной истории	VKI - целое значение, 0 (не обнаружен) или 1 (найден)

проверка адреса регистрации/проживания физического лица по списку адресов массовой регистрации	MassRegAdr - целое значение, 0 (не обнаружен) или 1 (найден)
проверка наличия данных физического лица в списке банкротов	BANKRUPT_CLIENT - целое значение, 0 (не обнаружен) или 1 (найден)
проверка данных физического лица на наличие в стоп-листах	StopListCrit - целое значение, 0 (не обнаружен) или 1 (найден)
проверка данных физического лица на наличие в стоп-листах и возврат балла негатива в случае обнаружения	WEIGHTNEGATIVE - сумма всех найденных полей в стоп-листах NEGATIVE_AND_WEIGHT - перечисление полей значение <NEGATIVE>=<WEIGHTNEGATIVE> через ';'.

Пример ответного сообщения:

```
<Object id="deadcode-beda-beda-beda-0cc47a78b9cd">
  <item>
    <key>CompPass</key>
    <value></value>
  </item>
  <item>
    <key>MassRegAdr</key>
    <value>0</value>
  </item>
  <item>
    <key>StopListCrit</key>
    <value>0</value>
  </item>
  <item>
    <key>VKI</key>
    <value>0</value>
  </item>
  <item>
    <key>BANKRUPTCY</key>
    <value>0</value>
  </item>
</Object>
<Object id="deadcode-a6e2-11e6-80cb-0cc47a78b9cd">
  <item>
    <key>CompPass</key>
    .....
  </item>
</Object>
```

Сервис интеграции с веб-сервисом MAIL.RU

Сервис предоставляет возможность получить скоринговую оценку физического лица от веб-сервиса «Mail.ru».

Для доступа к API сервиса Mail.ru требуется авторизация. Авторизация производится по ключу (токену) доступа, передаваемому в каждом запросе. Данные для подключения пользователь API получает от компании «Mail.ru Group» при регистрации. Полученные данные записываются в конфигурационный файл микросервиса SRV.MAILRU_API

```
mailru.url=https://XXX
SKBLAB.mailru.Login=XXX
SKBLAB.mailru.Token=XXX
SKBLAB.mailru.ApplicationName=XXX
SKBLAB.mailru.ProductName=XXX
```

Сервис реализует кэширование данных, все ответы, полученные от mail.ru сохраняются в БД в таблице "CHECK_SCORE"¹. При выполнении повторного запроса в течение одной недели сервис вернет инициатору взаимодействия значения, сохраненные в БД.

Для проведения проверки нужно заполнить данные физического лица:

Поле	Примечание
Имя	
Фамилия	
Отчество	
Дата рождения	формат "dd.mm.yyyy"
Пол	Мужской/женский
Модель скоринговой проверки	Модель сервиса mail.ru
Номер мобильного телефона	
Электронная почта	
Тип ответного сообщения	XML/JSON

В ответном сообщении сервис возвращает скоринговый балл, полученный от сервиса Mail.ru в переменной (key) SCORE.

Для демонстрации работы сервиса интеграции настроена эмуляция ответов со стороны Mail.ru, настройки подключения к эмулятору записаны в конфигурационном файле базовой поставки платформы.

Пример ответного сообщения в формате XML:

```
<data>
<items>
  <item>
    <key>SCORE</key>
    <value>0.44080663</value>
  </item>
  <item>
    <key>DATE</key>
    <value>07.10.2020</value>
  </item>
  <item>
    <key>MATCHINGLEVEL</key>
    <value>3</value>
  </item>
</items>
</data>
```

Пример ответного сообщения в формате JSON:

```
{"valueList":[{"key":"SCORE","value":"0.7007928"},
{"key":"DATE","value":"06.10.2020"},
{"key":"MATCHINGLEVEL","value":"3"},
```

¹ Структура таблицы в Приложении № 1

Сервис проверки данных по базе исполнительных производств

Сервис позволяет взаимодействовать с внешним веб-сервисом API FSSP <https://api-ip.fssprus.ru/> с целью получения общедоступной информации об исполнительных производствах контрагента – физического лица.

Для доступа к API FSSP требуется авторизация. Авторизация производится по ключу доступа (токену), передаваемому в каждом запросе и получаемому пользователем API при регистрации.

API FSSP выполнен в архитектурном стиле REST. Параметры подключения к API настраиваются в конфигурационном файле микросервиса SRV.CHECKDEBT.FSSP.API, там же хранится токен доступа:

```
fssp.api.tokens=XXX
api.client.connectionTimeout=4000
api.client.readTimeout=4000
api.client.url=https://api-ip.fssprus.ru/api/v1.0
api.client.enableSSL=true
```

API FSSP возвращает ответы на запросы в асинхронном режиме. Так как API FSSP в часы пиковых нагрузок может не обрабатывать запросы на протяжении нескольких часов, то в сервисе реализован механизм перезапросов и установления тайм-аутов. Параметры настраиваются в конфигурационном файле микросервиса SRV.CHECKDEBT.FSSP.API:

```
api.time.per.task = 1800000 ограничение в мс (тайм-аут) на выполнение одной задачи
store.intervals = 1sx3,3sx5,5sx5,30sx10,1mx10,5mx10,10mx10,30m – интервалы выполнения перезапросов в миллисекундах
throttle.capacity = максимальное количество запросов к внешнему сервису за 30 секунд
```

API FSSP по одному запросу осуществляет поиск исполнительных производств в одном регионе. Сервис имеет механизм дубликации запросов, который позволяет производить поиск данных из одного запроса по нескольким регионам одновременно. В режиме дубликации единичный запрос будет размножен, т.е. для каждого региона, указанного в параметре *regions.duplicate.list* будет создан отдельный запрос, который обрабатывается как независимый, а результаты объединяются воедино, прежде чем быть помещенными в ответное сообщение. Данный режим используется для запросов по всем регионам.

Политика дубликации запроса определяется параметром сервиса *regions.duplicate.policy*:

- *REQUIRED* – дубликация осуществляется всегда, независимо от переданного региона;
- *DEMANDED* - по требованию, если код региона равен -1, в ином случае одиночный запрос;
- *NONE* - никогда, запрос всегда будет одиночным

Параметры механизма дубликации задаются в конфигурационном файле микросервиса SRV.CHECKDEBT.FSSP.API:

aggregator.timeout = 3600000 - Время на объединение результата, используется в режиме дубликации, когда запрос совершается по всем регионам, время в мс
regions.duplicate.policy – политика дубликации запроса по всем регионам
regions.duplicate.list - Список кодов регионов через запятую, используется только для дубликации.

Для проверки физического лица нужно заполнить данные:

Поле	Примечание
Дата рождения	формат “dd.mm.yyyy”
Имя	
Отчество	
Фамилия	
Код региона для поиска ИП	Номер региона или -1 для поиска по всем регионам
Идентификационный номер клиента	

Параметры ответного сообщения:

Переменная	Значение
Name	ФИО
BirthDate	Дата рождения
Address	Фрагмент адреса
IPNum	Номер исполнительного производства
IPDate	Дата исполнительного производства
IDRequisites	Данные исполнительного листа
IPCloseDate	Дата завершения исполнительного производства
IPCloseReason	Основание завершения исполнительного производства
Service	Сервисное сообщение
IDSubjName	Основание для возбуждения исполнительного производства
IPDebt	Сумма взыскания
OspProperty	Информация о подразделении судебных приставов, в котором находится исполнительное производство
SPI	Информация о судебном приставе

Пример ответного сообщения:

```
<data>
  <items>
    <item>
      <key>From</key>
      <value>API</value>
      <type>0</type>
    </item>
```

```

<item>
  <key>Name</key>
  <value>СВЕТЛАНА ВИКТОРОВНА НОЖКИНА</value>
  <type>0</type>
</item>
<item>
  <key>BirthDate</key>
  <value>20.01.1983</value>
  <type>0</type>
</item>
<item>
  <key>Address</key>
  <value>Н ТАГИЛ</value>
  <type>0</type>
</item>
<item>
  <key>IPNum</key>
  <value>52810/18/66008-ИП</value>
  <type>0</type>
</item>
<item>
  <key>IPDate</key>
  <value>30.05.2018</value>
  <type>0</type>
</item>
<item>
  <key>IDRequisites</key>
  <value>Исполнительный лист от 31.01.2017 № 010965867
Постановление о взыскании исполнительского сбора ДЗЕРЖИНСКИЙ РАЙОННЫЙ СУД Г.
НИЖНИЙ ТАГИЛ</value>
  <type>0</type>
</item>
<item>
  <key>IPCloseDate</key>
  <value></value>
  <type>0</type>
</item>
<item>
  <key>IPCloseReason</key>
  <value></value>
  <type>0</type>
</item>
<item>
  <key>Service</key>
  <value></value>
  <type>0</type>
</item>
<item>
  <key>IDSubjName</key>
  <value>Задолженность по кредитным платежам (кроме ипотеки):
311116.48 руб. Исполнительский сбор: 22818.08 руб.</value>
  <type>0</type>
</item>
<item>
  <key>IPDebt</key>
  <value>311116.48</value>
  <type>0</type>
</item>
<item>
  <key>OspProperty</key>

```



```
<value>Дзержинский РОСП г. Нижнего Тагила 622051, Россия,  
Свердловская обл., , г. Нижний Тагил, , ул. Правды, д. 13, , </value>  
<type>0</type>  
</item>  
<item>  
<key>SPI</key>  
<value>ВЕРШИН И. А. +7(3435)33-40-31 </value>  
<type>0</type>  
</item>  
</items>  
</data>
```

Сервис интеграции с почтовым шлюзом MS Exchange

Сервис позволяет взаимодействовать с почтовым сервером Microsoft Exchange для отправки почтовых сообщений и получения информации о жизненном цикле обработки сообщения (обработке, формировании, отправке, доставке, прочтении).

Взаимодействие с Microsoft Exchange осуществляется по протоколу Exchange Web Services (EWS), разработанному компанией Microsoft и предназначенному для управления почтой и другими компонентами составляющими MS Exchange.

Для доступа к MS Exchange по протоколу EWS требуется авторизация с использованием сертификатов безопасности.

Настройки параметров подключения и авторизации хранятся в конфигурационном файле микросервиса SRV.MSEXCH.

```
exchange.URL=XXX – параметры подключения  
trustStore.file=XXX – место хранения сертификата  
trustStore.password=XXX – пароль сертификата  
kerberos.fileconf=XXX - место хранения файла с конфигурацией авторизации через  
kerberos  
SKB.mail.login=XXX – логин входя в учетную запись  
SKB.mail.password=XXX – пароль вода в учетную запись  
mail.SenderEmailAddress=XXX – адрес, указанный в качестве отправителя письма  
exchange.ReceiveTimeout=60000  
exchange.ConnectionTimeout=30000  
exchange.MajorVersion=14
```

Сервис позволяет не только отправить письмо через почтовый сервер, но и получить информацию об этапах прохождения письма – обработке, доставке, прочтении.

По умолчанию сервис формирует ответные сообщения на каждом из этапов прохождения письма. Время формирования уведомлений ограничено тайм-аутом, параметры которого настраиваются в конфигурационном файле микросервиса SRV.MSEXCH:

```
scheduler.period=180s – периодичность запуска шедулера
scheduler.maxPoolSize=1
scheduler.readTimeout=180 – таймаут для уведомления о чтении в минутах
scheduler.deliverTimeout=180 – таймаут для уведомления о доставке в минутах
```

В уведомлении передаются коды результатов успешной обработки сообщения или коды ошибок.

Коды результатов обработки:

Код обработки	Описание
1001	Processing message – входящее сообщение обработано
1002	Exchange CreateItem response – почтовое отправление создано
1003	Exchange SendItem response - отправлено

Коды ошибок, возвращаемых сервисом:

Код ошибки	Описание
1010	Ошибка отправки, ошибка получения из сервиса
1002	Ошибка валидации сообщения
-1	Прочие исключения

Для отправки письма необходимо заполнить данные:

Значение	Примечание
Адрес отправителя письма	Электронная почта
Адрес получателя письма	Электронная почта
Тема письма	В формате txt
Текст письма	Блок текста

Без реального подключения к Microsoft Exchange сервис возвращает уведомления о приеме сообщений, но дальнейшая отправка почтового сообщения не осуществляется.

Пример уведомления:

```
<ns2:EXCH_A_EMAILMESSAGE>
  <ns3:Sender>
    <EmailAddress>sender@skblab.ru</EmailAddress>
  </ns3:Sender>
  <ns3:OfReceiver/>
  <ns2:Response>
    <ns2:Success>1003-Exchange SendItem response:
    NO_ERROR</ns2:Success>
  </ns2:Response>
</ns2:EXCH_A_EMAILMESSAGE>
```

Сервис интеграции с API Контур.Фокус

Сервис позволяет взаимодействовать с API Контур.Фокус (<https://focus.kontur.ru/site/capabilities/api>) с целью получения различных данных о контрагентах по ИНН или ОГРН.

Взаимодействие с Контур.Фокусом осуществляется по протоколу HTTPS с использованием метода get – сервис вызывает определенные методы и получает ответы с информацией, соответствующей типу запроса. Взаимодействие с API Контур.Фокус осуществляется в формате xml.

Параметры подключения и общие таймауты соединения (в миллисекундах) настраиваются в конфигурационном файле микросервиса SRV.FOCUS:

```
#url for https:  
#focus.url=https://focus-api.kontur.ru/  
#url for http4:  
focus.url=focus-api.kontur.ru/  
focus.connectionTimeout=30000  
focus.soTimeout=30000  
focus.maxConnectionsPerHost=10  
focus.retryCount=3
```

Список реализованных методов API «Контур.Фокус»:

- **api3/req** – получение основных реквизитов ЕГРЮЛ/ЕРГИП – метод позволяет получать текущие и исторические реквизиты юридических лиц или предпринимателей;
- **api3/req/mon** - получение списка ИНН/ОГРН лиц, установленных под наблюдение, по которым были изменения в реквизитах на указанную дату;
- **api3/briefReport** – экспресс-отчет по контрагенту (светофор);
- **api3/analytics** – получение списка маркеров и числовых индикаторов для построения аналитики;
- **api3/contacts** - получение сведений о контактных телефонах;
- **api3/egrDetails** – получение расширенных сведений из ЕГРЮЛ/ЕРГИП;
- **api3/licences**- получение информации по лицензиям;
- **api3/buh** - получение показателей бухгалтерской отчетности;
- **api3/fssp** -получение информации об исполнительных производствах;
- **api3/govPurchasesOfParticipant** - получение информации о гос.закупках, в которых участвовала организация;
- **api3/govPurchasesOfCustomer** - получение информации о гос.закупках, в которых организация участвовала как заказчик;
- **api3/stat** – получение информации об использовании ключа;
- **api3/personaffiliates/req** – получение данных по компаниям, связанным с указанным физическим лицом (руководитель и/или учредитель и/или ИП);
- **api3/checkPassport** - проверка недействительных паспортов;

- **api3/companyAffiliates/req** - реквизиты организаций, связанных с компанией;
- **agent/briefreportssummary** - получение дополнительного экспресс-отчета по контрагенту (светофор);

Доступ к API Контур.Фокус осуществляется с использованием токенов доступа. Токены доступа хранятся в таблице базы данных SYS_PROPERTIES².

В таблице заполняются параметры:

Поле	Значение
SYS_ID	SRV.FOCUS
PROPERTY_NAME	KEY
PROPERTY_VALUE	Наименование бизнес-процесса, для которого будет использоваться токен доступа (SKBLAB.FOCUS)
KEY1	Токен доступа

Для проведения проверки нужно заполнить данные:

Поле	примечание
ОГРН организации	
ИНН организации или физического лица/ИП	10 или 12 цифр
Метод	Выбор метода API Контур.Фокус для проверки

Для демонстрации работы на платформе настроено подключение к тестовому контуру системы <https://developer.kontur.ru/doc/focus?about=2>, данные тестового клиента заполнены как значение по умолчанию в интерфейсе платформы. Для тестового клиента доступны не все методы проведения проверки.

Ответное сообщение соответствует формату ответного сообщения от Контур.Фокус, описание форматов - <https://focus-api.kontur.ru/api3/index.html>

Сервис работы с POS-терминалами INPAS

Сервис позволяет взаимодействовать с API INPAS (<https://www.inpas.ru/>) для передачи новых и изменения существующих мерчантов и терминалов POS в базе данных INPAS.

Взаимодействие осуществляется с двумя версиями API, имеющими две разные точки входа. Каждой точке входа присваивается свой псевдоним (alias). Доступ к API требует авторизации.

Параметры авторизации для каждого из alias, тайм-ауты и адреса точек входа настраиваются в конфигурационном файле микросервиса SRV.INPAS.SEND:

² Структура таблицы представлена в Приложении № 1

```
serverAliases=inpasSrv1,inpasSrv2

inpasSrv1.login=XXX
inpasSrv1.password=XXX
inpasSrv1.url=XXX
inpasSrv1.connectionTimeout= 30000
inpasSrv1.soTimeout=30000

inpasSrv2.login=XXX
inpasSrv2.password=XXX
inpasSrv2.url=XXX
inpasSrv2.connectionTimeout=30000
inpasSrv2.soTimeout=30000
```

Для отправки сообщения нужно заполнить данные:

Поле	Примечание
Тип терминала	В соответствии с кодировкой INPAS
Идентификатор мерчанта	Идентификатор в INPAS
Имя	
Фамилия	
Телефон	
Почтовый адрес	
Наименование мерчанта	
Тип мерчанта	В соответствии с классификацией INPAS
ИНН мерчанта	10 или 12 цифр
МСС терминала	Код МСС
Номер контракта	Номер контракта с INPAS
Идентификатор терминала	Идентификатор терминала в INPAS
Фактическое имя	
Фактический адрес	
Время работы	В формате пн,вт,ср,чт,пт 00:00-24:00; сб,вс 00:00-24:00
Модель терминала	
Имя владельца терминала	
Фамилия владельца терминала	
Телефон владельца терминала	
комментарий	

В ответ приходит статус сообщения (принят, ошибки).

Для демонстрации работы сервиса на платформе настроен тестовый пример, данные для проведения теста приведены в интерфейсе платформы как значения по умолчанию.

Пример ответного сообщения:

```
<data>
  <type>ABS:RQ_INPAS_ADD</type>
  <Contract-Numb>49128101475</Contract-Numb>
  <status>success</status>
</data>
```

Сервис проверки инцидентов НСПК

Сервис позволяет взаимодействовать с «Национальной системой платежных карт» (НСПК) для получения данных об инцидентах, связанных с мошенническим поведением юридических лиц, данные о которых передаются банками-эквайерами платёжной системе «МИР».

Доступ к API НСПК (<https://www.nspk.ru/banks/developer/api-spp/>) осуществляется через REST-интерфейс с использованием токенов доступа.

Параметры подключения к API и тайм-ауты соединения настраиваются в конфигурационном файле микросервиса SRV.NSPK:

```
nspkRequest.url=spp-test.nspk.ru
nspkRequest.MethodAuth=tsp/v1/user/login
nspkRequest.username=XXX
nspkRequest.password=XXX
nspkRequest.MethodReq=tsp/v1/tsp

nspkRequest.connectionTimeout=60000
nspkRequest.soTimeout=60000
nspkRequest.retryCount=3

trustStore.file=resources/truststore-t.ks
trustStore.password=XXX
```

Токены доступа получают автоматически при подключении к API на основании логина/пароля и хранятся в таблице SYS_PROPERTIES базы данных. В таблице указываются параметры:

Поле	Значение
SYS_ID	SRV.NSPK
PROPERTY_NAME	TOKEN
PROPERTY_VALUE	Токен доступа

Поиск инцидентов осуществляется по ИНН и ОГРН организации.

Поле	Примечание
ОГРН	
ИНН	10 или 12 цифр

В ответ передаются коды обнаруженных в системе инцидентов (через запятую).

Для демонстрации работы сервиса интеграции настроена эмуляция ответов со стороны NSPK, настройки подключения к эмулятору записаны в конфигурационном файле базовой поставки платформы.

Пример ответного сообщения:

```
<spp_response>
  <status>Ok</status>
  <status_info></status_info>
```

```
<inn>6652018890</inn>
<ogrn1056602434140</ogrn>
<incident_codes>00,02,01</incident_codes>
</spp_response>
```

Сервис взаимодействия с системой Моё Дело

Сервис предназначен для взаимодействия с системой аутсорсинга бухгалтерского учёта «Моё Дело» <https://www.moedelo.org/buhgalterskie-uslugi/outsorsing> и позволяет получить налоговый календарь.

Доступ к API MOEDEL0 требует авторизации и осуществляется с использованием ключевых сертификатов, которые должны быть получены при заключении договора с компанией, предоставляющей услугу.

Параметры подключения к API MOEDEL0, место хранения ключевого сертификата и тайм-ауты соединения задаются в конфигурационных файлах микросервиса SRV.MOEDEL0_API

```
SKB.moedelo.url=XXX
SKB.moedelo.login=XXX
SKB.moedelo.password=XXX

moedelo.connectionTimeout=60000
moedelo.soTimeout=60000
moedelo.retryingCount=3

trustStore.file=XXX
trustStore.password=XXX
```

Запрос на получение налогового календаря может быть как в формате JSON так и XML. Для получения календаря нужно заполнить данные:

Значение	Примечание
Система налогообложения	В соответствии с классификацией сервиса «МоёДело»
Организационно-правовая форма бизнеса	В соответствии с классификацией сервиса «МоёДело»
Дата начала	формат “dd.mm.yyyy”
Дата окончания	формат “dd.mm.yyyy”

Авансовый налог на прибыль	Включать или не включать (false или true)
----------------------------	---

Ответное сообщение будет представлено в формате JSON или в формате XML в соответствии с форматом запроса. Состав переменных в сообщении соответствует описанному в документации API MOEDELO <https://restapi.moedelo.org/s/?url=/docs>.

Для демонстрации работы сервиса интеграции настроена эмуляция ответов со стороны API MoeDelo, настройки подключения к эмулятору записаны в конфигурационном файле базовой поставки платформы. В ответе эмулятора всегда приводится фрагмент налогового календаря на январь 2018 года.

Пример ответного сообщения (налоговый календарь)

В формате JSON:

```
[
  {
    "Id": 549,
    "Url": "/Requisites/FirmCalendar/549",
    "PeriodNumber": 4,
    "PeriodType": "quarter",
    "Year": 2017,
    "StartDate": "2018-02-01T00:00:00",
    "EndDate": "2018-04-01T00:00:00",
    "Title": "Сдать расчет 6-НДФЛ за 2017 год"
  },
  {
    "Id": 553,
    "Url": "/Requisites/FirmCalendar/553",
    "PeriodNumber": 12,
    "PeriodType": "month",
    "Year": 2017,
    "StartDate": "2018-01-01T00:00:00",
    "EndDate": "2018-01-15T00:00:00",
    "Title": "Отчет в ПФР: СЗВ-М за декабрь 2017 года"
  },
  .....
  {
    "Id": 645,
    "Url": "/Requisites/FirmCalendar/645",
    "PeriodNumber": 3,
    "PeriodType": "quarter",
    "Year": 2018,
    "StartDate": "2018-10-01T00:00:00",
    "EndDate": "2018-10-31T00:00:00",
    "Title": "Сдать расчет 6-НДФЛ за 9 месяцев 2018 года"
  }
]
```

В формате XML:

```
<data>
  <event>
    <EndDate>2018-04-01T00:00:00</EndDate>
    <Id>549</Id>
    <PeriodNumber>4</PeriodNumber>
```



```

<PeriodType>quarter</PeriodType>
<StartDate>2018-02-01T00:00:00</StartDate>
<Title>Сдать расчет 6-НДФЛ за 2017 год</Title>
<Url>/Requisites/FirmCalendar/549</Url>
<Year>2017</Year>
</event>
<event>
  <EndDate>2018-01-15T00:00:00</EndDate>
  <Id>553</Id>
  <PeriodNumber>12</PeriodNumber>
  <PeriodType>month</PeriodType>
  <StartDate>2018-01-01T00:00:00</StartDate>
  <Title>Отчет в ПФР: СЗВ-М за декабрь 2017 года</Title>
  <Url>/Requisites/FirmCalendar/553</Url>
  <Year>2017</Year>
</event>
.....
<event>
  <EndDate>2018-10-31T00:00:00</EndDate>
  <Id>645</Id>
  <PeriodNumber>3</PeriodNumber>
  <PeriodType>quarter</PeriodType>
  <StartDate>2018-10-01T00:00:00</StartDate>
  <Title>Сдать расчет 6-НДФЛ за 9 месяцев 2018 года</Title>
  <Url>/Requisites/FirmCalendar/645</Url>
  <Year>2018</Year>
</event>
</data>

```

Сервис определения оператора сотовой связи

Сервис предназначен для определения принадлежности номера мобильного телефона к тому или иному оператору сотовой связи.

При получении запроса сервис осуществляет поиск номера телефона в базе данных и возвращает инициатору запроса наименование оператора сотовой связи, к которому привязан номер телефона.

Сервис производит поиск мобильного оператора в базе данных перенесенных номеров (таблица PHONE базы данных платформы), если номер телефона не обнаружен, то осуществляет поиск по таблице соответствия индексов номеров телефонов сотовому оператору (таблица базы данных платформы PHONE_CODE). В ответном сообщении возвращается имя оператора, которому соответствует номер телефона, либо запись «Телефонный номер не найден», если принадлежность номера телефона не определена.

Данные в таблице PHONE_CODE заполняются на основании «Выписки из реестра российской системы и плана нумерации» Федерального агентства связи (<https://rossvyaz.gov.ru/devatelnost/resurs-numeracii/vypiska-iz-reestra-sistemy-i-plana-numeracii>). Структура таблицы приведена в Приложении №1.

Данные о перенесенных номерах ежедневно (в 22-00 мск) публикует ФГУП ЦНИИС в «Центральной базе перенесенных номеров» (ЦБДПН), доступ к которой осуществляется через сервер Numlex.

Сервис определения оператора сотовой связи ежедневно (расписание устанавливается в конфигурационном файле микросервиса SRV.PHONE.LOADER) создает подключение к удаленному SFTP серверу ЦБДПН и скачивает архив с перенесенными номерами. Данные из полученного архива загружаются в таблицу PHONE. Структура таблицы приведена в Приложении № 1.

Параметры подключения к SFTP-серверу и размещения файла с архивом задаются в конфигурационном файле микросервиса SRV.PHONE.LOADER:

```
localFolder=/tmp/PHONE/  
login=XXX  
password=XXX  
ipAdress=prod-sftp.numlex.ru  
port=3232  
sftpFolderGlobal=numlex/Port_All_New  
seconds=0  
minutes=0  
hours=4
```

Для проведения проверки нужно задать номер телефона:

Поле	Примечание
Номер телефона	

В ответе (формат XML) возвращается переменная (key) «OPERATOR», в значении которой (value) указано наименование оператора сотовой связи «МТС», либо переменная (key) «Error», в значении которой приводится расшифровка ошибки определения номера.

Пример ответа:

```
<data>  
  <item>  
    <key>OPERATOR</key>  
    <value>MTS</value>  
  </item>  
</data>
```

Приложение № 1. Структура таблиц базы данных.

Таблица BL_ADR_ALIASES - типы регионов

Поле	Тип данных	Примечание
TYPE	VARCHAR2(128 BYTE)	Тип региона
ALIAS	VARCHAR2(16 BYTE)	Сокращение

Таблица BL_ADRES_MASS_REG – адреса массовой регистрации

Поле	Тип данных	Примечание
COUNTRY	VARCHAR2(100 BYTE)	Страна
SUBJECT	VARCHAR2(100 BYTE)	Регион
AREA	VARCHAR2(100 BYTE)	Область
CITY	VARCHAR2(100 BYTE)	Город
SETTLEMENT	VARCHAR2(100 BYTE)	Район
STREET	VARCHAR2(100 BYTE)	Улица
HOUSE	VARCHAR2(100 BYTE)	Дом
LETTER	VARCHAR2(100 BYTE)	Корпус
FLAT	VARCHAR2(100 BYTE)	квартира

Таблица BL_ADR_MASS_REG_NRM – адреса массовой регистрации (нормализованная таблица)

Поле	Тип данных	Примечание
COUNTRY	VARCHAR2(100 BYTE)	Страна
AREA	VARCHAR2(100 BYTE)	Область
CITY	VARCHAR2(100 BYTE)	Город
SETTLEMENT	VARCHAR2(100 BYTE)	Район
STREET	VARCHAR2(100 BYTE)	Улица
HOUSE	VARCHAR2(100 BYTE)	Дом
FLAT	VARCHAR2(100 BYTE)	Квартира
SUBJECT_CODE	NUMBER	Код субъекта

Таблица BL_PASSPORT – список скомпрометированных паспортов

Поле	Тип данных	Примечание
LASTNAME	VARCHAR2(100 BYTE)	Фамилия
FIRSTNAME	VARCHAR2(100 BYTE)	Имя
MIDDLENAME	VARCHAR2(100 BYTE)	Отчество
BIRTHDAY	DATE	Дата рождения
SERIALPASS	VARCHAR2(50 BYTE)	Серия паспорта

NUMBERPASS	VARCHAR2(50 BYTE)	Номер паспорта
ISSUEDBY	VARCHAR2(500 BYTE)	Кем выдан паспорт

Таблица BL_SIGNAL – стоп-листы

Поле	Тип данных	Примечание
LASTNAME	VARCHAR2(200 BYTE)	Фамилия
FIRSTNAME	VARCHAR2(200 BYTE)	Имя
MIDDLENAME	VARCHAR2(200 BYTE)	Отчество
BIRTHDAY	DATE	Дата рождения
NEGATIVE	VARCHAR2(60 BYTE)	Наименование негативного фактора
WEIGHTNEGATIVE	VARCHAR2(50 BYTE)	Балл негатива

Таблица BL_VKI – внутренняя кредитная история

Поле	Тип данных	Примечание
LASTNAME	VARCHAR2(100 BYTE)	Фамилия
FIRSTNAME	VARCHAR2(100 BYTE)	Имя
MIDDLENAME	VARCHAR2(100 BYTE)	Отчество
BIRTHDAY	DATE	Дата рождения
CLIENT_ID	NUMBER	ИД клиента в системе 1
CLIENT_GID	NUMBER	ИД клиента в системе 2
CODE_ID	NUMBER(10,0)	Код кредитной истории
SOURCE_SYSTEM_ID	NUMBER	ИД системы источника данных
REPORTDATE	DATE	Дата

Таблица CCL_EFRSB_DECISION_NAMES – список физических лиц – банкротов

Поле	Тип данных	Примечание
IBSO_ID	VARCHAR2(100 BYTE)	Идентификатор физического лица
FIO	VARCHAR2(512 BYTE)	ФИО
BIRTHDATE	DATE	Дата рождения

Таблица CHECK_SCORE – скоринговый балл сервиса «mail.ru»

Поле	Тип данных	Примечание
NAME	VARCHAR2(128 BYTE)	Имя
SURNAME	VARCHAR2(128 BYTE)	Фамилия
PATRONYMIC	VARCHAR2(128 BYTE)	Отчество
BIRTHDATE	DATE	Дата рождения

SEX	NUMBER	Пол
MODEL	VARCHAR2(128 BYTE)	Имя модели
MOBILE	VARCHAR2(128 BYTE)	Мобильный номер телефона
EMAIL	VARCHAR2(128 BYTE)	Электронная почта
SCORE	NUMBER	Скоринговый балл
CREATE_DATE	DATE	Дата заполнения данных
MATCHINGLEVEL	NUMBER	Уровень совпадения

Таблица PHONE - список номеров телефонов, переведенных к другим операторам стовой связи

Поле	Тип данных	Примечание
TEL	VARCHAR2(30 BYTE)	Номер телефона
OPER	VARCHAR2(50 BYTE)	Оператор сотовой связи

Таблица PHONE_CODE – список диапазонов номеров, закрепленных за оператором сотовой связи

Поле	Тип данных	Примечание
OPER	VARCHAR2(200 BYTE)	Оператор сотовой связи
CODE	VARCHAR2(30 BYTE)	Код
FIRST_INDEX	VARCHAR2(30 BYTE)	Первый индекс диапазона номеров
LAST_INDEX	VARCHAR2(30 BYTE)	Последний индекс диапазона номеров

Таблица SYS_PROPERTIES

Поле	Тип данных	Примечание
SYS_ID	VARCHAR2(100 BYTE)	Идентификатор системы
PROPERTY_NAME	VARCHAR2(200 BYTE)	Параметр
PROPERTY_VALUE	VARCHAR2(1000 BYTE)	Значение параметра
DESCRIPTION	VARCHAR2(200 BYTE)	Описание
UPDATE_TIMESTAMP	TIMESTAMP(6)	Время модификации
KEY1	VARCHAR2(200 CHAR)	Дополнительный параметр 1
KEY2	VARCHAR2(200 CHAR)	Дополнительный параметр 2
KEY3	VARCHAR2(200 CHAR)	Дополнительный параметр 3

Приложение № 2. Описание API.

Для взаимодействия на программном уровне на платформе разработан API.

API работает в асинхронном режиме.

Сервис публикует методы на <http://bus-op01-t:9100>, которые принимают параметры в заголовках. Содержимое заголовков должно быть завернуто в base64.

В ответ методы возвращают заголовок `HWarpId`, который является идентификатором для получения ответа.

Размещается ответ по адресу <http://bus-op01-t:9100/warp/response>, если ответ найден, то будет возвращен `HTTP_RESPONSE_CODE = 200` и тело ответа, если не найден, то будет возвращен `HTTP_RESPONSE_CODE = 204`.

Количество методов соответствует количеству сервисов, предоставляемых платформой. Форматы ответов приведены в разделах с описанием сервисов.

1. *blacklist_v1* - Проверка физического лица по «чёрным спискам»- v1

Заголовок	Описание
BirthDay	Дата рождения формат "dd.mm.yyyy"
FirstName	Имя
LastName	Фамилия
MiddleName	Отчество
NumberPass	Номер паспорта
SerialPass	Серия паспорта
RegionReg	Область адреса регистрации
AreaReg	Район адреса регистрации
CityReg	Город адреса регистрации
SettlementReg	Населенный пункт адреса регистрации
StreetReg	Улица адреса регистрации
HouseReg	Дом адреса регистрации
FlatReg	Квартира адреса регистрации
RegionRes	Область адреса проживания
AreaRes	Район адреса проживания
CityRes	Город адреса проживания
SettlementRes	Населенный пункт адреса проживания
StreetRes	Улица адреса проживания
HouseRes	Дом адреса проживания
FlatRes	Квартира адреса проживания
HCheck	Методы проверки

Список возможных методов проверки (один или несколько, указывается через запятую):

- *processPassport* – проверка по списку скомпрометированных паспортов
- *processVKI* – проверка наличия внутренней негативной кредитной истории и возврат ассоциативных массивов с данными в случае обнаружения

- *processRegistration* – проверка наличия адреса регистрации физического лица в списке адресов массовой регистрации по нормализованной таблице
- *processResidence* – проверка наличия адреса проживания физического лица в списке адресов массовой регистрации по нормализованной таблице
- *processPhysicalBankruptcy* – проверка наличия данных физического лица в списке банкротов
- *processStopList* - проверка данных физического лица на наличие в стоп-листах и возврат балла негатива в случае обнаружения

2. *blacklist_v2* - Проверка физического лица по «черным спискам» - v2

Заголовок	Описание
FirstName	Имя
LastName	Фамилия
MiddleName	Отчество
PassNum	Номер паспорта
PassSer	Серия паспорта
Region	Регион адреса регистрации
City	Город адреса регистрации
Street	Улица адреса регистрации
House	Дом адреса регистрации
BirthDate	Дата рождения формат “dd.mm.yyyy”

3. *blacklist_v3* Проверка физических лиц по «чёрным спискам» -v3

Заголовок	Описание
ObjectId1	Идентификатор клиента
FirstName1	Имя
LastName1	Фамилия
MiddleName1	Отчество
PassNum1	Номер паспорта
PassSer1	Серия паспорта
PassIss1	Кем выдан паспорт
Country1	Страна адреса регистрации
Region1	Область адреса регистрации
District1	Район адреса регистрации

City1	Город адреса регистрации
Locality1	Район адреса регистрации
Street1	Улица адреса регистрации
House1	Дом адреса регистрации
Letter1	Литера
Hblock1	Номер блока
Apartment1	Квартира
BirthDate1	Дата рождения формат "dd.mm.yyyy"

Наименования заголовков по остальным лицам сопровождаются индексом 2, 3... например ObjectId2, ObjectId3

4. *scoringMail* - Скоринговая оценка физического лица от сервиса Mail.ru

Заголовок	Описание
NAME	Имя
SURNAME	Фамилия
PATRONYMIC	Отчество
BIRTHDATE	Дата рождения
SEX	Пол (1-мужской, 2 – женский)
MODEL	Модель скоринговой проверки
MOBILE	Номер мобильного телефона
EMAIL	Электронная почта
FORMAT	Формат сообщения (JSON/XML)

5. *FSSP_FI* - Проверка по базе данных ФССП (физические лица)

Заголовок	Описание
BIRTHDATE	Дата рождения
NAME	Имя
PATRONYMIC	Отчество
SURNAME	Фамилия
REGION_CODE	Код региона для поиска ИП
CLIENTIDRUNBEGIN	Идентификационный номер клиента

6. *SendMail* - Отправка почтовых сообщений через MS Exchange

Заголовок	Описание
Sender	Адрес отправителя письма

Recipients	Адрес получателя письма
Subject	Тема письма
Body	Текст письма

7. *ConturFocus* - Интеграция с «Контур-Фокус»

Заголовок	Описание
CLOGRN	ОГРН организации
CLINN	ИНН организации или физического лица/ИП
Method	Метод из списка

Список возможных значений переменной *method* (может быть только один):

- **req** – получение основных реквизитов ЕГРЮЛ/ЕРГИП;
- **reqMon** - получение списка ИНН/ОГРН лиц, установленных под наблюдение
- **briefReport** – экспресс-отчет по контрагенту (светофор)
- **analytics** – получение списка маркеров и числовых индикаторов для построения аналитики
- **contacts** - получение сведений о контактных телефонах
- **egrDetails** – получение расширенных сведений из ЕГРЮЛ/ЕРГИП
- **licences**- получение информации по лицензиям
- **buh** - получение показателей бухгалтерской отчетности
- **fssp** -получение информации об исполнительных производствах
- **govPurchasesOfParticipant** - получение информации о гос.закупках, в которых участвовала организация
- **govPurchasesOfCustomer** - получение информации о гос.закупках, в которых организация участвовала как заказчик
- **stat** – получение информации об использовании ключа
- **personaffiliatesReq** – получение данных по компаниям, связанным с указанным физическим лицом (руководитель и/или учредитель и/или ИП)
- **checkPassport** - проверка недействительных паспортов
- **companyAffiliatesReq** - реквизиты организаций, связанных с компанией
- **briefreportssummary** - получение дополнительного экспресс-отчета по контрагенту

8. *inpas* - Сервис работы с POS-терминалами INPAS

Заголовок	Описание
Type	Тип терминала
merchantid	Идентификатор мерчанта
namecl	Имя

surnamecl	Фамилия
phonecl	Телефон
emailcl	Почтовый адрес
merchantname	Наименование мерчанта
merchanttype	Тип мерчанта
Inn	ИНН мерчанта
Mcc	МСС терминала
contractnumb	Номер контракта
terminalid	Идентификатор терминала
factname	Фактическое имя
factaddress	Фактический адрес
worktime	Время работы
termmodel	Модель терминала
namecltsp	Имя владельца терминала
surnamecltsp	Фамилия владельца терминала
phonecltsp	Телефон владельца терминала
comment	комментарий

9. *nspk* - Сервис проверки инцидентов НСПК

Заголовок	Описание
OGRN	ОГРН
INN	ИНН

10. *TaxCalendar* - Запрос налогового календаря в бухгалтерии «МоёДело»

Заголовок	Описание
TaxationSystem	Система налогообложения (число)
BusinessForm	Организационно-правовая форма бизнеса (число)
StartDate	Дата начала
EndDate	Дата окончания
IsPayProfitAdvancesTax	Авансовый налог на прибыль (true/false)

11. *PhoneOperator* - Сервис определения оператора сотовой связи

Заголовок	Описание
MOBILE	Номер телефона