

Руководства пользователей Goodfin

Руководства пользователей Goodfin

Exported on Oct 06, 2022

Table of Contents

1.1	Поиск документации	6
1.2	Избранные страницы.....	6
1.3	Недавно обновлённые страницы	7
2.1	Краткие сведения о возможностях системы Goodfin.....	8
2.2	Оглавление	8
2.3	Для клиентов/агентов, работающих в Goodfin	9
2.3.1	Программные требования для работы в системе Goodfin	9
2.3.2	Перед началом работы с системой пользователь (клиент, агент) должен проверить соответствие своего рабочего места следующим требованиям:.....	9
3.1	Список определений и сокращений, используемых в системе Goodfin	10
3.2	Список определений и сокращений, используемых в документации для разработчиков.....	16
4.1	1. Вариант по умолчанию: "Goodfin" <-> "Адаптер Goodfin" <-> "Банковский сервис"	17
4.2	2. Вариант с использованием мощностей Goodfin на транспортном уровне: "Внешняя система с интерфейсом для клиентов/агентов, схожая по целевому назначению с Goodfin" <-> "Адаптер Goodfin" <-> "Банковский сервис"	17
4.3	Порядок обмена интеграционными сообщениями в рамках бизнес-процесса получения, обработки заявки и выдачи Продукта по схеме интеграции "Goodfin" <-> "Адаптер Goodfin" <-> "Банковский сервис"	18
4.3.1	1. Порядок обмена интеграционными сообщениями с банковскими сервисами в нотации BPMN без привязки к особенностям банковского сервиса	19
4.3.2	2. Порядок обмена интеграционными сообщениями с банковскими сервисами на базе платформы IT-Finance в нотации UML (sequence diagram)	20
4.4	Вариант с использованием мощностей Goodfin на транспортном уровне: "Внешняя система с интерфейсом для клиентов/агентов, схожая по целевому назначению с Goodfin" <-> "Адаптер Goodfin" <-> "Банковский сервис".....	24
4.4.1	1. Порядок обмена интеграционными сообщениями в рамках бизнес-процесса получения, обработки заявки и выдачи Продукта.....	24
4.4.2	2. Плюсы и минусы схемы интеграционного взаимодействия для владельцев HUB	25
4.4.3	3. Укрупненная схема компонентов	26
4.4.4	4. Что нужно сделать владельцам HUB?	26
4.5	Структура API методов для работы с внешними системами	32
4.5.1	Общий порядок взаимодействия систем.....	33
4.5.2	Получение списка обязательных полей	33
4.5.3	Поиск продуктов на основе имеющихся на текущий момент данных	35
4.5.4	Информация о подходящих продуктах и требования по дальнейшему заполнению	35
4.5.5	Отправка заявки в сервисы с указанием продуктов	37
4.5.6	Информация о результате принятия заявки в работу.....	38
4.5.7	Сообщения отправляемые при изменении статуса заявок.....	39
4.5.8	Подписание заявки на стороне внешней системы	40
5.1	Оглавление	44
5.2	Введение	44
5.2.1	Краткое описание стадий работ и возможностей	44
5.3	I. Подготовительные работы по интеграции с применением API.....	45
5.3.1	Доступ к описанию методов API.....	46
5.3.2	Концепция организации передаваемых данных, используемая в API.....	46
5.3.3	Примерный порядок работ для подготовки к интеграции с Goodfin с использованием песочницы.....	51
5.3.4	Quickstart. Примеры интеграции с Goodfin	51
5.3.5	Для начала взаимодействия с тестовым окружением необходимо зарегистрировать сервис. 51	
5.3.6	Требуется получить токен, чтобы сервис как система мог авторизоваться перед Goodfin как системой.....	53
5.3.7	Это точка, куда система Goodfin (или песочница как эмулятор системы) будет отправлять все данные и сообщения для сервиса.	54
5.3.8	Шаг 3.1 Объявление/обновление параметров статусной модели сервиса	57
5.3.9	Шаг 3.2 Объявление/обновление передаваемых кодов ошибок.....	63
5.3.10	ШАГ 4. Обработка сервисом получаемых заявок от Goodfin (AddDealApplication)	66
5.3.11	ШАГ 4.1. Скачивание документов по заявке	68

5.3.12	После получения заявки и скачивания документов, банковский сервис должен оповестить систему Goodfin об успешном получении заявки или ошибке.	70
5.3.13	После получения заявки, банковский сервис может оповещать систему Goodfin об изменении статусов полученных и обрабатываемых заявок.	72
5.3.14	После получения заявки, банковский сервис может оповестить систему Goodfin о присвоении в сервисе внутреннего номера заявке. Информация о номере заявки, присвоенном в сервисе, может помочь клиентам при обращении в службу поддержки, если возникают проблемы с заявкой у клиента.	76
5.3.15	В системе Goodfin пользователь в любой момент, пока не получен финансовый продукт по отправленной заявке, может отозвать заявку. Соответственно, в сервис в точку интеграции, объявленную на шаге 3, будет отправлено сообщение типа CancelDealApplication. В ответ сервис должен научиться переводить отозванную заявку в соответствующий статус, указывающий, что заявка больше недействительна. Далее, по результате смены статуса сервиса должен отправить сообщение типа ChangeDealApplicationStatusEvent (см. шаг 4.3).	77
5.3.16	Goodfin шлет сообщения типа CommonResponse только с BASIC авторизацией.	78
5.3.17	Пример реализации "заглушки" средствами SoapUI, чтобы проверить получение заявок.	79
5.4	II.1. Создание продукта в виде конфигурационного файла yaml.	86
5.5	II.2. Создание тарификатора в виде конфигурационного файла yaml.	99
5.6	II.3. Тестирование и доработка банковского продукта.	115
5.7	III. Получение доступа в личный кабинет поставщика сервиса действующей системы Goodfin.	131
5.8	IV. Импорт подготовленных продуктов в действующую систему Goodfin.	131
5.9	Справочники, используемые при настройке yaml продукта и тарификатора.	131
5.10	Справочники, используемые в отправляемых заявках.	132
6.1	Оглавление.	133
6.2	Введение и краткое описание стадий работ.	133
6.2.1	Доступ к документации с описанием типов и методов API.	133
6.2.2	Краткое описание возможностей.	133
6.2.3	Краткое описание стадий работ для интеграции с действующей системой Goodfin.	134
6.2.4	Краткое описание стадий работ для предварительного тестирования интеграции в песочнице.	135
6.3	Примеры структуры передаваемого лида.	136
6.3.1	Пример структуры передаваемого лида по типу продукта "Банковская гарантия на исполнение", если инициатором отправки был агент.	136
6.3.2	Пример почти минимальной структуры передаваемого лида по типу продукта "Банковская гарантия на исполнение", если инициатором отправки была сама система - поставщик лидов.	146
6.4	Варианты требований к наполненности лидов в зависимости от целей.	148
6.4.1	Вариант I: Хотим передавать минимум информации, чтобы клиент максимально быстро передавался в работу менеджеру Goodfin.	148
6.4.2	Вариант II: Хотим передавать максимум информации, чтобы по переданному лиду без задержек формировалась заявка для отправки в банк.	149
7.1	Поля карточки компании: company_fields.	159
7.2	Статусы подключаемого сервиса: dealApplicationStatuses.	165
7.3	Прикрепляемые документы: docs_types.	166
7.4	Электронные торговые площадки: etp_types.	171
7.5	Федеральные законы: federallaw_types.	171
7.6	Типы прикрепляемых файлов: file_types.	172
7.7	Поля финансовых показателей: finance_fields.	172
7.8	Формы бухгалтерской и финансовой отчетности: finance_forms.	176
7.9	Документы, удостоверяющие личность: identity_doc_types.	177
7.10	Формы предпринимательской деятельности: legal_types.	177
7.11	Отчётные периоды: period_types.	178
7.12	Способы размещения заказа: placingway_types.	178
7.13	Типы продуктов: product_types.	179
7.14	Поля в заявках по продуктам: product_types_fields.	179
7.15	Поставщики сервиса: services.	203
7.16	Типы тарифов: tariff_cell_types.	204
7.17	Типы расчётных счетов: types_banking_details.	204
7.18	Пользовательские действия: userActions.	205
7.19	Типы валидаторов: validators.	209
7.20	Правила применимости: productApplicabilityRule, applicabilityRule.	211
7.20.1	Оглавление.	211
7.20.2	1 Примеры скриптов правил применимости.	211

7.20.3	2	Примеры получения значений полей по данным заявки для использования в правилах применимости.....	225
8.1		All Branches in this Space.....	229
8.2		Confirm bulk merging	229

Добро пожаловать!

Это главная страница для пространства документации системы Goodfin. Цель пространства - помочь освоить систему и получить максимум пользы от работы в системе.

База знаний для пользователей системы (клиентов, агентов) ведется по адресу <https://help.goodfin.ru/>.

1 Краткие сведения о возможностях системы

1. Единая точка входа для пользователя системы, возможность работы в различных кабинетах, как физическое лицо, как сотрудник компании-клиента, как сотрудник компании-агента.
2. Возможность подключения внешних сервисов, предоставляющих продукты и услуги как для компаний (ЮЛ, ИП), так и для физических лиц.
3. Поставщики сервисов могут гибко настроить список требуемых данных и документов по каждому продукту, которые должен предоставить клиент.
4. Для каждой компании (ЮЛ, ИП) в системе заводится только одна карточка компании-клиента. Работа с карточкой клиента осуществляется по принципам:
 - постоянного накопления и актуализации данных и документов, чтобы сократить время на оформление заявок на получение различных продуктов и услуг;
 - отслеживания всех изменений, сделанных пользователями системы в карточке клиента, чтобы сократить вероятность сокрытия данных недобросовестными менеджерами и агентами;
 - к карточке клиента получают доступ пользователи системы в соответствии с правами доступа;
 - минимальные системные требования к обязательности заполнения данных компании;
 - пользователь может ориентироваться на требования сервисов (банков) к обязательным данным по разным типам продуктов и заполнять только требуемые данные.
5. Возможность отслеживания всех выполненных изменений при совместной работе в системе. Для этого в системе предусмотрены компоненты: режим «Черновик», панель «Навигатор», «Лента активностей», «История».
6. Управление контактами, связанными с различными компаниями, с возможностью накопления данных обо всех средствах связи с ними и поддержки их в актуальном состоянии. Исключение возможности сокрытия контактных данных и «увода клиентской базы» при увольнении менеджера по продажам, при отключении агента.
7. Возможность работы нескольких агентов с одним клиентом по разным типам продуктов, сервисам.
8. Единое хранилище документов в системе по клиентам, агентам, сделкам, взаиморасчетам между участниками системы с обеспечением версионности документов (на любую точку во времени будут сохранены актуальные версии документов, поддержка историчности).

1.1 Поиск документации

Поиск

1.2 Избранные страницы

Содержимое по меткам

Содержимое с указанными метками не найдено.

1.3 Недавно обновлённые страницы

- [Поставщики сервиса: services](#) авг 31, 2022 • обновлено [Боровков Андрей Владимирович](#) • [изменения](#)
- [Структура API методов для работы с внешними системами](#) дек 16, 2021 • обновлено [Пучков Александр Андреевич \[X\]](#) • [изменения](#)
- [Типы валидаторов: validators](#) ноя 15, 2021 • обновлено [Алексеева Алена Андреевна](#) • [изменения](#)
- [Правила применимости: productApplicabilityRule, applicabilityRule](#) ноя 12, 2021 • обновлено [Пучков Александр Андреевич \[X\]](#) • [изменения](#)
- [Правила применимости: productApplicabilityRule, applicabilityRule](#) окт 25, 2021 • обновлено [Алексеева Алена Андреевна](#) • [изменения](#)

2 Общие сведения

2.1 Краткие сведения о возможностях системы Goodfin

1. Единая точка входа для пользователя системы, возможность работы в различных кабинетах, как физическое лицо, как сотрудник компании-клиента, как сотрудник компании-агента.
2. Возможность подключения внешних сервисов, предоставляющих продукты и услуги как для компаний (ЮЛ, ИП), так и для физических лиц.
3. Поставщики сервисов могут гибко настроить список требуемых данных и документов по каждому продукту, которые должен предоставить клиент.
4. Для каждой компании (ЮЛ, ИП) в системе заводится только одна карточка компании-клиента. Работа с карточкой клиента осуществляется по принципам:
 - постоянного накопления и актуализации данных и документов, чтобы сократить время на оформление заявок на получение различных продуктов и услуг;
 - отслеживания всех изменений, сделанных пользователями системы в карточке клиента, чтобы сократить вероятность сокрытия данных недобросовестными менеджерами и агентами;
 - к карточке клиента получают доступ пользователи системы в соответствии с правами доступа;
 - минимальные системные требования к обязательности заполнения данных компании;
 - пользователь может ориентироваться на требования сервисов (банков) к обязательным данным по разным типам продуктов и заполнять только требуемые данные.
5. Возможность отслеживания всех выполненных изменений при совместной работе в системе. Для этого в системе предусмотрены компоненты: режим «Черновик», панель «Навигатор», «Лента активностей», «История».
6. Управление контактами, связанными с различными компаниями, с возможностью накопления данных обо всех средствах связи с ними и поддержки их в актуальном состоянии. Исключение возможности сокрытия контактных данных и «увода клиентской базы» при увольнении менеджера по продажам, при отключении агента.
7. Возможность работы нескольких агентов с одним клиентом по разным типам продуктов, сервисам.
8. Единое хранилище документов в системе по клиентам, агентам, сделкам, взаиморасчетам между участниками системы с обеспечением версионности документов (на любую точку во времени будут сохранены актуальные версии документов, поддержка историчности).

2.2 Оглавление

2.3 Для клиентов/агентов, работающих в Goodfin

База знаний для пользователей системы (клиентов, агентов) ведется по адресу <https://help.goodfin.ru/>.

2.3.1 Программные требования для работы в системе Goodfin

Наименование	Версия	Официальный сайт продукта
КриптоПро CSP	3.6 и выше	http://www.cryptopro.ru/
КриптоПро ЭЦП Browser plug-in	2.0 и выше	http://www.cryptopro.ru/
Браузеры (любой из):		
Firefox	60.x и выше	https://www.mozilla.org/
Google Chrome	67.x и выше	https://www.google.ru/chrome/

2.3.2 Перед началом работы с системой пользователь (клиент, агент) должен проверить соответствие своего рабочего места следующим требованиям:

1. Наличие ЭЦП (усиленной квалифицированной электронной цифровой подписи).
2. Браузер Mozilla Firefox версии 60 и выше, Google Chrome версии 67 и выше, Internet Explorer версии 11 и выше.
3. Для использования ЭЦП в работе Программы необходимо иметь установленный плагин в используемом браузере. Для установки плагина следуйте инструкции <https://www.cryptopro.ru/products/cades/plugin>.
4. Доступ в сеть Интернет.

3 Список определений и сокращений

Оглавление

- [Список определений и сокращений, используемых в системе Goodfin](#)
- [Список определений и сокращений, используемых в документации для разработчиков](#)

3.1 Список определений и сокращений, используемых в системе Goodfin

Термин или сокращение	Определение или расшифровка
Авторизация	Это предоставленное какому-либо лицу право на совершение определенных действий в конкретной системе – на сайте или терминале. Авторизация позволяет исключить доступ к информации третьих лиц.
Агент	Внешний по отношению к Системе участник бизнес-процесса (юридическое лицо или индивидуальный предприниматель), который ведет работу с клиентами - соискателями банковской гарантии или другого продукта, доступного к получению в Системе, и/или другими агентами (субагентами), которые также ведут работу со своими клиентами.
Банковский сервис	см. "Сервис".
Банковский продукт	см. "Продукт".
ББ	Бухгалтерский баланс.
БГ	Банковская гарантия. Может быть нескольких типов: "банковская гарантия на исполнение контракта", "банковская гарантия на возврат аванса", "банковская гарантия на участие", "банковская гарантий обеспечения гарантийных обязательств".
БИК	Банковский идентификационный код.
Веб-страница	Экранная форма, отображаемая в браузере, представленная в виде совокупности объектов, структурированных для выполнения определенной задачи в удобной для пользователя форме.
Вкладка	Часть экранной формы Программы, отображаемая в браузере, служит для структурирования информации и набора определенных действий для пользователя.
ДОРС	Дистанционное открытие расчетного счета.
ЕГРИП	Единый государственный реестр индивидуальных предпринимателей.
ЕГРЮЛ	Единый государственный реестр юридических лиц.
ЕНВД	Единый налог на вмененный доход.
ЕСХН	Единый сельскохозяйственный налог.

Термин или сокращение	Определение или расшифровка
Заявка	Основной объект системы, состоящий из определенной совокупности представлений необходимой информации (параметров заявки), касающихся бизнес-процесса выдачи банковской гарантии или иного продукта и обладающий определенными состоянием (статусами). Заявки автоматически генерируются при создании и заполнении сделки (см. "Сделка"). В рамках одной сделки можно отправить заявки во все доступные банковские сервисы для повышения вероятности получения требуемого банковского продукта.
Заявка на продукт	см. "Заявка".
ИП	Индивидуальный предприниматель.
Кабинет	см. "Личный кабинет".
Карточка клиента	Основной объект Системы, в котором собирается, структурируется, актуализируется вся информация о клиенте. Служит для сокращения времени по заполнению данных в сделках и сокращению числа ошибок в данных, передаваемых в составе заявки на продукт в банковские и финансовые сервисы. В качестве клиента могут выступать компании/ЮЛ или ИП.
Карточка компании	см. "Карточка клиента".
Карточка компании/ИП	см. "Карточка клиента".
КБК	Код бюджетной классификации
КИК	Кредит на исполнение контракта
КЛАДР	Классификатор адресов Российской Федерации
Клиент	Юридическое лицо, индивидуальный предприниматель, организация без прав юридического лица, физическое лицо, приобретающий самостоятельно или через агента необходимые продукты посредством Системы (см. также "Контрагент").
Компании	Различные объединения, товарищества, хозяйственные общества, фирмы, корпорации, предприятия, имеющие разные организационно-правовые формы (формы деловой организации работы, деятельности), которые являются по отношению к Системе клиентами (контрагентами). В Системе условно индивидуальные предприниматели также относятся к сущности Компании.
Контакт	Физическое лицо как представитель контрагента в Системе, может являться или не являться пользователем Системы. Для компаний и агентов в качестве контактов могут выступать сотрудники/ доверенные партнеры.
Контрагент	Участник бизнес-процесса, одна из сторон договорных отношений в Системе, см. также "Клиент".
КПП	Код причины постановки на учет. Код, состоящий из 9 цифр, в которых зашифрована информация о постановке организации на учет в Налоговой службе.

Термин или сокращение	Определение или расшифровка
Лид	Это набор структурированных данных различной степени наполненности, полученный из внешней системы от Поставщика лидов. Содержит сведения о потенциальном клиенте и его потребности в банковском или финансовом продукте. Лид может стать основанием сделки в Системе.
Личный кабинет	Именованная и сгруппированная совокупность допустимых действий над данными, относящаяся к определенной группе ролей. По сути, автоматизированное рабочее место Пользователя Системы. Например, существуют ЛК клиента, ЛК агента, ЛК владельца системы, ЛК поставщика сервисов, ЛК поставщика лидов, ЛК УЦ.
ЛК	Личный кабинет.
Менеджер агента	см. "Оператор агента".
Менеджер по работе с клиентами	см. "Оператор агента".
Навигатор	Панель вывода контекстной помощи и другой информации, упрощающей работу пользователя, расположенная справа от основной рабочей области на веб-страницах Системы. Используется для быстрого перемещения по полям, в которых необходимо внести исправления, отката к предыдущему состоянию вводимых данных и пр.
Объект системы	Именованная и сгруппированная совокупность данных и допустимых действий над данными. К объектам системы относятся: обращение, сделка, сервис, продукт, мероприятие, лид, документ, статья базы знаний, задача, заявка, клиент, контакт, сотрудник, очередь задач, E-mail, рассылка, звонок.
ОГРН	Основной государственный регистрационный номер записи о создании юридического лица либо записи о первом представлении в соответствии с Федеральным законом Российской Федерации «О государственной регистрации юридических лиц» сведений о юридическом лице, зарегистрированном до введения в действие указанного Закона.
ОГРНИП	Основной государственный регистрационный номер индивидуального предпринимателя — номер присваивается в налоговой инспекции при государственной регистрации индивидуального предпринимателя и внесении в Единый государственный реестр индивидуальных предпринимателей.
ОКАТО	Общероссийский классификатор объектов административно-территориального деления — классификатор объектов административно-территориального деления Российской Федерации, входит в состав «Единой системы классификации и кодирования технико-экономической и социальной информации Российской Федерации».
ОКВЭД	Общероссийский классификатор видов экономической деятельности.
Окно	См. «Веб-страница».

Термин или сокращение	Определение или расшифровка
ОКОГУ	Общероссийский классификатор органов государственной власти и управления.
ОКПО	Общероссийский классификатор предприятий и организаций.
ОКСМ	Общероссийский классификатор стран мира.
ОКТМО	Общероссийский классификатор территорий муниципальных образований.
ОКФС	Общероссийский классификатор форм собственности.
Оператор агента	Сотрудник агента, который ведет работу с клиентами от имени агента (см. "Агент") в рамках установленных полномочий (назначенной роли).
Организация	см. "Компания".
ОСГЗ	Обеспечение сопровождения государственного заказа.
ОСНО	Общая система налогообложения.
Пакет документов к продукту	Запрашиваемый набор обязательных документов, которые клиент должен предоставить при формировании заявки на определенный продукт в ходе сделки.
Панель "Категории"	Панель меню, отражающего категории полей, сгруппированных определенным образом. Используется для быстрого перемещения по группам полей, а также для информирования о категориях, в которых имеются поля с ошибками заполнения. Панель располагается слева от основной рабочей области на веб-страницах Системы и отображается в режиме работы с карточкой клиента (контрагента).
Панель навигатора	см. "Навигатор".
ПДЛ	Публичное должностное лицо.
Плагин ЭЦП	Это устанавливаемый программный модуль, динамически подключаемый к основной программе, в данном случае к браузеру, и предназначенный для расширения и/или использования её возможностей, в данном случае для возможности подписания документов Вашим ЭЦП в Программе.
Пользователь	Авторизованное в Системе физическое лицо, которое может работать в Системе в соответствии с назначенной ролью и представлять как собственные интересы, так и интересы компаний, агентов и т.п.
Поставщик продуктов	ЮЛ/Пользователь системы, имеющий возможность посредством Системы и/или Сервиса предоставления Продуктов, интегрированного с Системой, предоставлять Продукты на определяемых им условиях заинтересованным в получении Продуктов лицам, в том числе Пользователям Системы.
Поставщик сервиса	см. "Поставщик продуктов".
Поставщик лидов	ЮЛ/Пользователь системы, имеющий возможность посредством Собственной системы и/или Сервиса,

Термин или сокращение	Определение или расшифровка
	интегрированного с Системой, передавать лиды в Систему для последующей их обработки.
Прикрепленный документ	Файл по запросу одного из участников бизнес-процесса, содержащий необходимую информацию и прикрепляемый пользователем в систему.
Продукт	Конкретный банковский документ (или другой документ, или другой результат сделки в Системе), который производится банком (или другим типом организации, предоставляющей сервис) для обслуживания клиента. Может быть получен клиентом в результате совершения сделки в Системе. Продукт имеет определенные правила и ограничения по предоставлению клиентам, например, ограничения по суммам, по типу клиента и т.п. Примером типа продукта может быть "Банковская гарантия", а продуктом "Банковская гарантия до 1 млн. руб. от такого-то сервиса".
ПСН	Патентная система налогообложения.
Региональный агент	Внешний по отношению к Системе участник бизнес-процесса (юридическое лицо), который ведет работу с агентами (см. "Агент"), а также ведут работу со своими клиентами.
РКО	Расчетно-кассовое обслуживание.
Роль	Именованная совокупность прав доступа к функциональным возможностям Системы.
Сделка	Это основной объект системы, позволяющий клиенту получить от банковского или иного сервиса требуемый банковский или финансовый продукт. Клиент создает сделку в Системе на определенный тип продукта и может отправлять в рамках сделки заявки на получение продукта во все доступные сервисы. Сделка считается успешной, если клиент получил желаемый продукт, а банковский или финансовый сервис - комиссию за выданный Продукт.
Сервис	Внешний по отношению к Системе программный комплекс (веб-сервис), в котором выполняется основной бизнес-процесс по предоставлению услуги или продукта клиенту. Между сервисом и Системой устанавливается взаимодействие: по передаче заявок из Системы в сервис на приобретение продукта; по обмену статусами прохождения заявок до успешного завершения или отказа; по обмену дополнительными запросами и пр. необходимой информацией. Сервис при подключении к Системе может предложить один или несколько продуктов, которые могут быть интересны клиентам Системы.
Сервис предоставления Продуктов	см. "Сервис".
Сертификат подписи	Это электронный документ, позволяющий проверить подлинность электронной подписи, он подтверждает принадлежность ключа проверки электронной подписи владельцу сертификата.
Сертификат электронной подписи	см. "Сертификат подписи".

Термин или сокращение	Определение или расшифровка
Сертификат электронной цифровой подписи	см. "Сертификат подписи".
Система	Информационная система Goodfin, доступ к которой осуществляется через сеть Интернет, в том числе путем обращения к сайту, расположенному по адресу: https://goodfin.ru/
СНИЛС	Уникальный номер индивидуального лицевого счёта застрахованного лица в системе обязательного пенсионного страхования.
Статус документа "Актуальный"	Статус, указывающий, что прикрепленный документ клиента считается действующим и может рассматриваться в составе заявки как подтверждающий документ. Большинство банковских сервисов рассматривают только действующие документы.
Статус документа "Архивный"	Статус, указывающий, что прикрепленный документ клиента более недействителен. Статус выставляется автоматически, если у прикрепленного документа установили дату окончания действия и срок действия истек. Некоторые банковские сервисы могут запрашивать и рассматривать всю цепочку одного типа документа (например, договор аренды помещения). Кроме того, автоматически в архивный переводится версия документа, если клиент/агент загружает поверх новую версию документа.
Статус заявки	Состояние заявки, определяющее допустимость тех или иных действий пользователя системы над заявкой.
Статус сделки	Состояние сделки, определяющее допустимость тех или иных действий пользователя системы со сделкой. Статусы сделок: создана, в работе, успешно завершена, отменена, закрыта системой.
Сущность системы	см. "Объект системы"
Тип продукта	Категория продуктов, предлагаемых Системой клиентам, например, "Банковская гарантия на исполнение контракта", "ОСГЗ", "ЭЦП", "ДОРС для РКО" и т.п.
ТЗ	Тендерный займ.
УСН	Упрощенная система налогообложения.
УФК	Управление Федерального казначейства.
УЦ	Удостоверяющий центр - доверенная организация, которая имеет право выпускать сертификаты электронной подписи юридическим и физическим лицам.
ФИАС	Федеральная информационная адресная система — единый российский государственный адресный реестр. В ФИАС хранятся данные об адресных объектах, расположенных на территории России: субъектах федерации, районах, населенных пунктах, улицах, домах и др.
Экранная форма	см. "Веб-страница".

Термин или сокращение	Определение или расшифровка
ЭЦП	Электронная цифровая подпись.
ЮЛ	Юридическое лицо.

3.2 Список определений и сокращений, используемых в документации для разработчиков

Термин или сокращение	Определение или расшифровка
endpoint	Это место, обозначаемое в API как URL + наименование и тип запроса, куда с помощью API отправляются эти запросы и место, где располагаются ресурсы для выполнения запроса.
эндпоинт	см. Endpoint

4 Схемы интеграционного взаимодействия с участием системы Goodfin

Оглавление

- [1. Вариант по умолчанию: "Goodfin" <-> "Адаптер Goodfin" <-> "Банковский сервис"](#)
- [2. Вариант с использованием мощностей Goodfin на транспортном уровне: "Внешняя система с интерфейсом для клиентов/агентов, схожая по целевому назначению с Goodfin" <-> "Адаптер Goodfin" <-> "Банковский сервис"](#)

4.1 1. Вариант по умолчанию: "Goodfin" <-> "Адаптер Goodfin" <-> "Банковский сервис"

Для кого подходит такой вариант интеграции?

- для поставщиков банковских сервисов (для банков, имеющих свою автоматизированную систему с бизнес-логикой обработки заявок от клиентов), готовых подключиться к системе Goodfin.

Что получит поставщик банковского сервиса?

- клиенты/агенты, работающие через Goodfin, смогут отправлять свои заявки в подключенный банковский сервис.

Как работает такая схема интеграции?

- см. подробнее статью [Порядок обмена интеграционными сообщениями в рамках бизнес-процесса получения, обработки заявки и выдачи Продукта по схеме интеграции "Goodfin" <-> "Адаптер Goodfin" <-> "Банковский сервис"](#)

Что нужно сделать поставщикам банковского сервиса?

- "научить" свою автоматизированную систему работать с API Goodfin. См. подробнее [1. Подготовительные работы по интеграции с применением API](#)

4.2 2. Вариант с использованием мощностей Goodfin на транспортном уровне: "Внешняя система с интерфейсом для клиентов/агентов, схожая по целевому назначению с Goodfin" <-> "Адаптер Goodfin" <-> "Банковский сервис"

"Внешняя автоматизированная система с пользовательским интерфейсом для клиентов/агентов, схожая по целевому назначению с Goodfin" далее по тексту будет обозначаться как **HUB**.

Для кого подходит такой вариант интеграции?

- для владельцев HUB, которые хотят для своих клиентов/агентов расширить спектр банковских услуг и предложений от банков, а также предоставить возможность получать банковские продукты клиентам, продолжая работать только в HUB.

Что получит владелец HUB?

- доступ ко всем подключенным банковским сервисам, с которыми сейчас взаимодействует Goodfin, а также ко всем вновь подключаемым банковским сервисам.

Как работает такая схема интеграции?

- см. подробнее статью [Вариант с использованием мощностей Goodfin на транспортном уровне: "Внешняя система с интерфейсом для клиентов/агентов, схожая по целевому назначению с Goodfin" <-> "Адаптер Goodfin" <-> "Банковский сервис"](#)

Что нужно сделать владельцам HUB?

- см. подробнее статью [Вариант с использованием мощностей Goodfin на транспортном уровне: "Внешняя система с интерфейсом для клиентов/агентов, схожая по целевому назначению с Goodfin" <-> "Адаптер Goodfin" <-> "Банковский сервис"](#)

4.3 Порядок обмена интеграционными сообщениями в рамках бизнес-процесса получения, обработки заявки и выдачи Продукта по схеме интеграции "Goodfin" <-> "Адаптер Goodfin" <-> "Банковский сервис"

Оглавление

- [1. Порядок обмена интеграционными сообщениями с банковскими сервисами в нотации BPMN без привязки к особенностям банковского сервиса](#)
- [2. Порядок обмена интеграционными сообщениями с банковскими сервисами на базе платформы IT-Finance в нотации UML \(sequence diagram\)](#)
 - [2.1. Пример интеграционных сообщений с момента отправки заявки в сервис до момента успешного завершения выдачи БГ \(статусы настраиваемые, могут быть любые и для каждого сервиса свои, на диаграмме пример сообщений с сервисом "Открытие" на базе IT-Finance\)](#)
 - [2.2. Пример интеграционных сообщений с момента когда заявка в сервисе и клиент отзывает заявку \(статусы настраиваемые, могут быть любые и для каждого сервиса свои, на диаграмме пример сообщений с сервисом "Открытие" на базе IT-Finance\)](#)
 - [2.3. Пример интеграционных сообщений с момента когда заявка в сервисе и заявка отзывается системой \(статусы настраиваемые, могут быть любые и для каждого сервиса свои, на диаграмме пример сообщений с сервисом "Открытие" на базе IT-Finance\)](#)
 - [2.4. Пример интеграционных сообщений с момента когда заявка находится на рассмотрении в сервисе и сервис отклоняет заявку \(статусы настраиваемые, могут быть любые и для каждого сервиса свои, на диаграмме пример сообщений с сервисом "Открытие" на базе IT-Finance\)](#)

Для кого предназначена данная статья?

- для поставщиков банковских сервисов (для банков, имеющих свою автоматизированную систему с бизнес-логикой обработки заявок от клиентов), готовых подключиться к системе Goodfin.

Что узнаете из статьи?

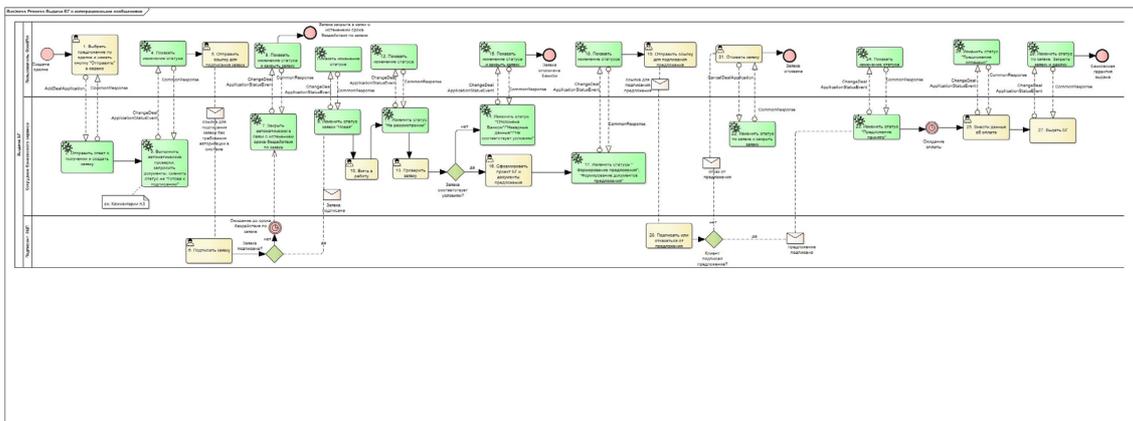
- как взаимодействуют банковские сервисы с Goodfin на базе API Goodfin в ходе получения и обработки заявок, отправленных клиентами/агентами Goodfin в банковский сервис.

Что нужно сделать поставщикам банковского сервиса, если принимают решение о подключении к Goodfin?

- "научить" свою автоматизированную систему работать с API Goodfin. См. подробнее [I. Подготовительные работы по интеграции с применением API](#)

4.3.1 1. Порядок обмена интеграционными сообщениями с банковскими сервисами в нотации BPMN без привязки к особенностям банковского сервиса

Для просмотра в увеличенном масштабе нажмите на картинку левой кнопкой мыши.



Комментарии к рис. выше:

1. Для интеграционных сообщений, отправляемых из Goodfin, типа AddDealApplication, CommonResponse, CancelDealApplication см. API документацию <https://sandbox.goodfin.ru/docs/bl-open-api/v1/index.html>, POST-запрос /goodfin/in.

2. Для интеграционных сообщений, принимаемых системой Goodfin из сервиса, см. документацию <https://sandbox.goodfin.ru/docs/shb-open-api/v1/index.html>:

POST-запрос /ad/v1/external/openapi/in для сообщений типа ChangeDealApplicationStatusEvent, UpdateApplicationParams и CommonResponse;

GET-запрос /api/v1/document/downloadaddocumentfile.

3. Шаг №3 показан укрупненно и состоит из более мелких шагов: автопроверка (после которой заявка может вернуться); запрос документов /api/v1/document/downloadaddocumentfile и, если все автоматические проверки пройдены, происходит смена статуса в "Готова к подписанию".

4. Пользователь Goodfin может отправить запрос на отзыв заявки на любом шаге бизнес процесса. Сервис может отказать в отзыве заявки или вернуть статус "Отозвана клиентом".

5. В процессе рассмотрения заявки в сервисе возможны различные шаги. В зависимости от жизненного цикла заявки с сервиса могут приходить статусы типа:

- "На рассмотрении";
- "Направлен запрос клиенту";
- "Формируется предложение";

"Формирование документов предложения";
"Проверка документов";
"Повторная проверка документов";
"Проект на согласовании";
"Запрос от клиента на доработку";
"Предложение принято";
"Предложение оплачено";
"Поиск предложений";
"Предварительное предложение принято"

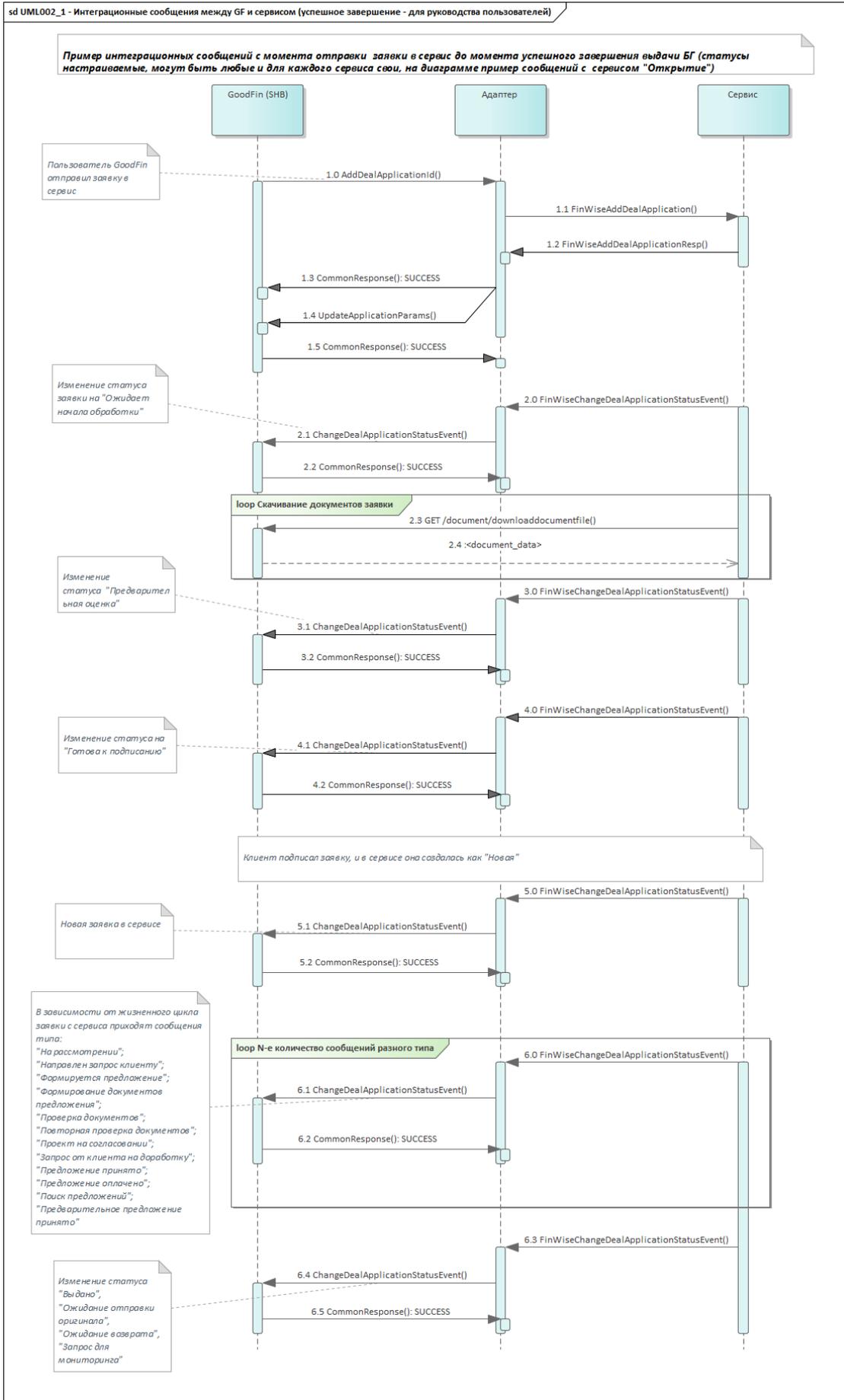
и др.

6. На рис. желтыми прямоугольниками обозначены задачи, которые выполняет пользователь, а зелёными - сервисные задачи, выполняемые системой.

4.3.2 2. Порядок обмена интеграционными сообщениями с банковскими сервисами на базе платформы IT-Finance в нотации UML (sequence diagram)

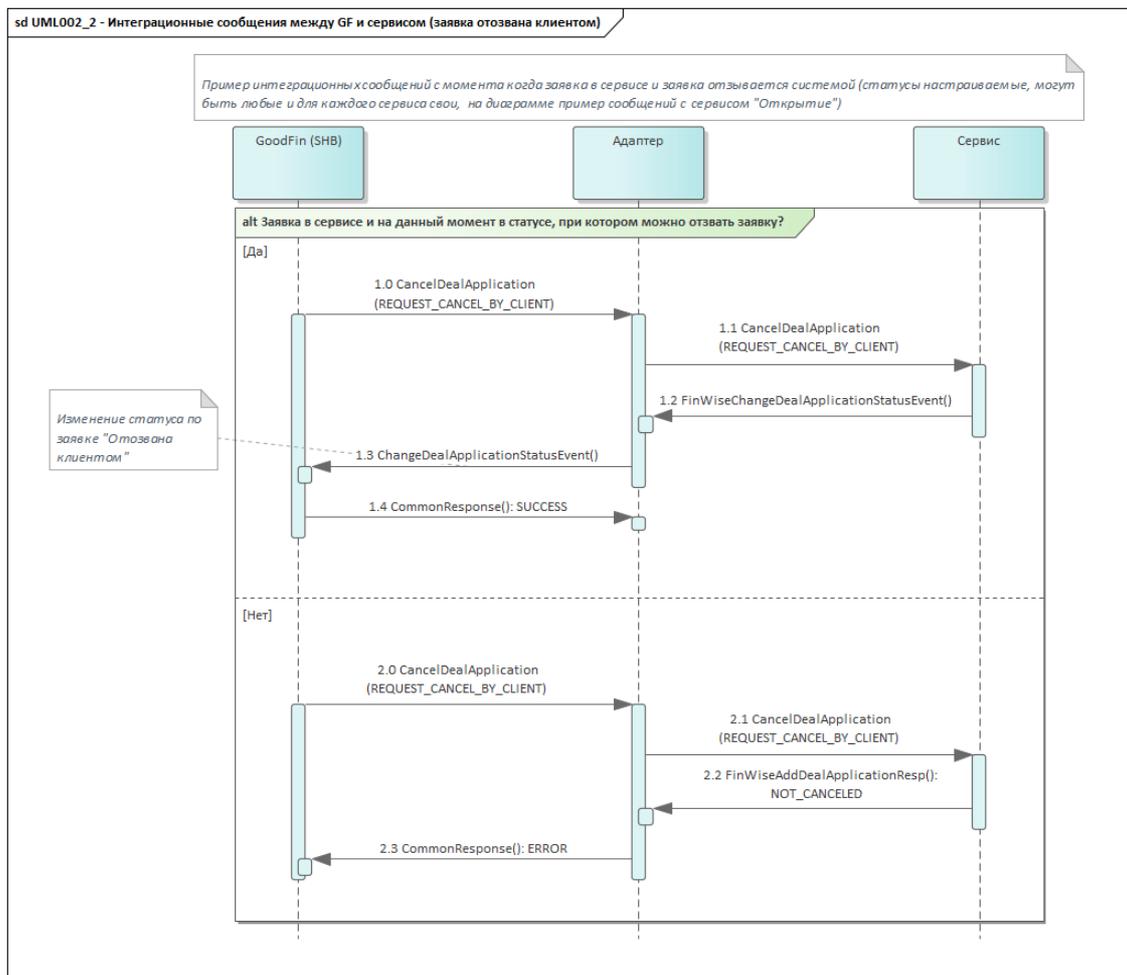
4.3.2.1 2.1. Пример интеграционных сообщений с момента отправки заявки в сервис до момента успешного завершения выдачи БГ (статусы настраиваемые, могут быть любые и для каждого сервиса свои, на диаграмме пример сообщений с сервисом "Открытие" на базе IT-Finance)

Интеграционные сообщения с пометкой FinWise... - это преобразованные под требования сервисов на базе платформы IT-Finance сообщения с учетом условных обозначений справочных данных, принятых в IT-Finance. Для прочих сервисов, если не оговорено иное, будут передаваться сообщения в том виде, в котором отправлены платформой Goodfin. Либо по договоренности выполняется доработка адаптера с учетом требований к данным подключаемого сервиса.



4.3.2.2 2.2. Пример интеграционных сообщений с момента когда заявка в сервисе и клиент отзывает заявку (статусы настраиваемые, могут быть любые и для каждого сервиса свои, на диаграмме пример сообщений с сервисом "Открытие" на базе IT-Finance)

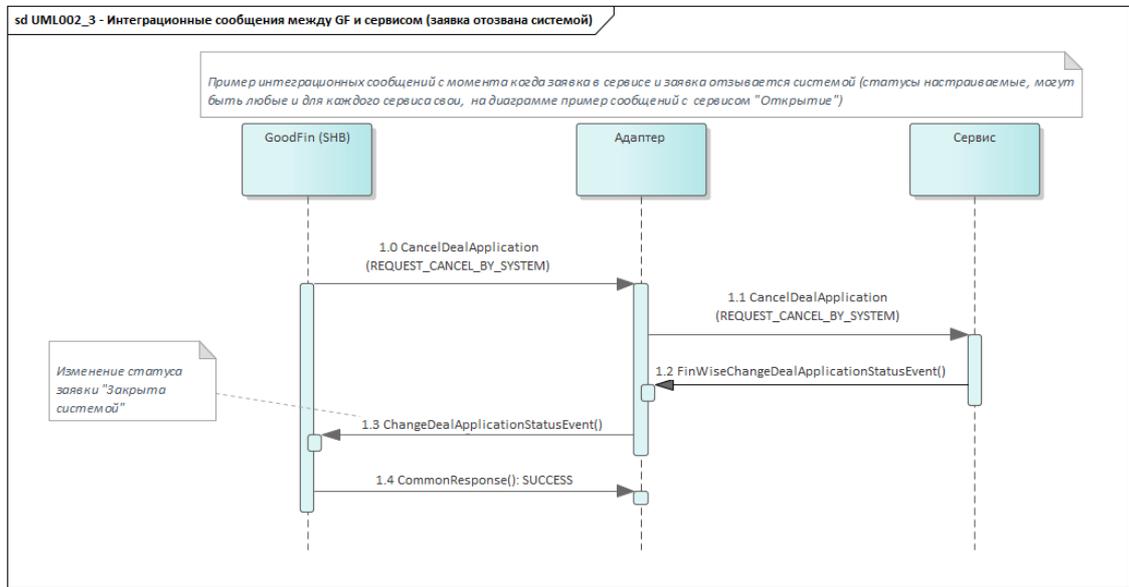
Интеграционные сообщения с пометкой FinWise... - это преобразованные под требования сервисов на базе платформы IT-Finance сообщения с учетом условных обозначений справочных данных, принятых в IT-Finance. Для прочих сервисов, если не оговорено иное, будут передаваться сообщения в том виде, в котором отправлены платформой Goodfin. Либо по договоренности выполняется доработка адаптера с учетом требований к данным подключаемого сервиса.



4.3.2.3 2.3. Пример интеграционных сообщений с момента когда заявка в сервисе и заявка отзывается системой (статусы настраиваемые, могут быть любые и для каждого сервиса свои, на диаграмме пример сообщений с сервисом "Открытие" на базе IT-Finance)

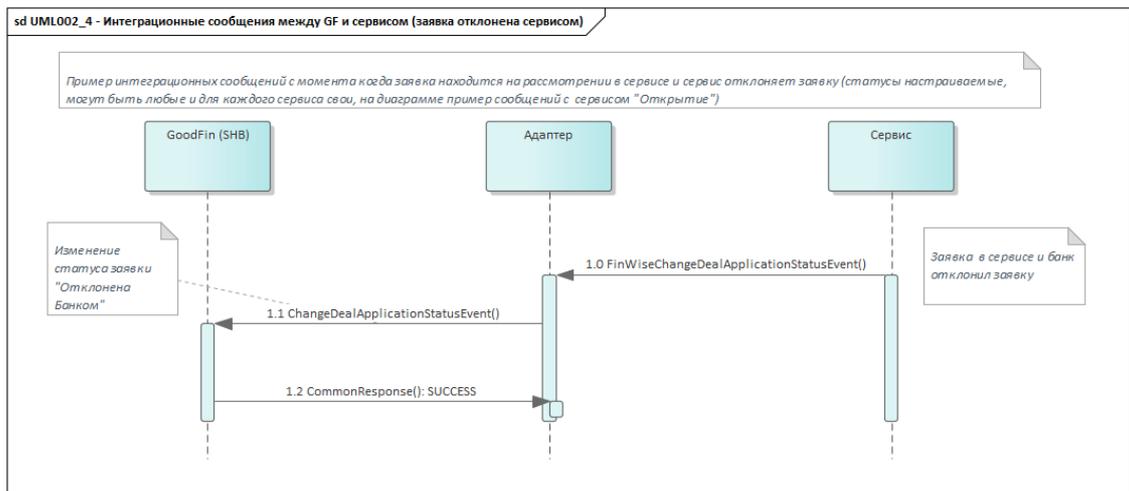
Интеграционные сообщения с пометкой FinWise... - это преобразованные под требования сервисов на базе платформы IT-Finance сообщения с учетом условных обозначений справочных данных, принятых в IT-Finance. Для прочих сервисов, если не оговорено иное, будут передаваться сообщения в том виде, в котором

отправлены платформой Goodfin. Либо по договоренности выполняется доработка адаптера с учетом требований к данным подключаемого сервиса.



4.3.2.4 2.4. Пример интеграционных сообщений с момента когда заявка находится на рассмотрении в сервисе и сервис отклоняет заявку (статусы настраиваемые, могут быть любые и для каждого сервиса свои, на диаграмме пример сообщений с сервисом "Открытие" на базе IT-Finance)

Интеграционные сообщения с пометкой FinWise... - это преобразованные под требования сервисов на базе платформы IT-Finance сообщения с учетом условных обозначений справочных данных, принятых в IT-Finance. Для прочих сервисов, если не оговорено иное, будут передаваться сообщения в том виде, в котором отправлены платформой Goodfin. Либо по договоренности выполняется доработка адаптера с учетом требований к данным подключаемого сервиса.



4.4 Вариант с использованием мощностей Goodfin на транспортном уровне: "Внешняя система с интерфейсом для клиентов/агентов, схожая по целевому назначению с Goodfin" <-> "Адаптер Goodfin" <-> "Банковский сервис"

Оглавление

- [1. Порядок обмена интеграционными сообщениями в рамках бизнес-процесса получения, обработки заявки и выдачи Продукта](#)
- [2. Плюсы и минусы схемы интеграционного взаимодействия для владельцев HUB](#)
- [3. Укрупненная схема компонентов](#)
- [4. Что нужно сделать владельцам HUB?](#)

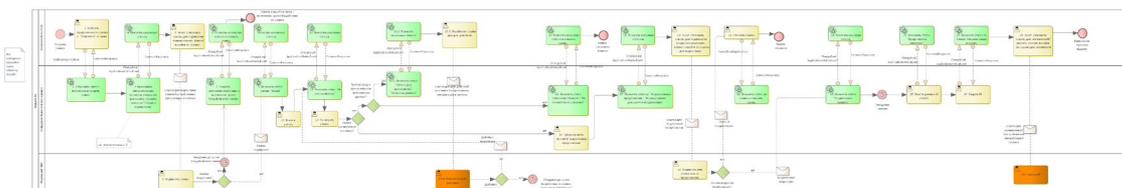
"Внешняя автоматизированная система с пользовательским интерфейсом для клиентов/агентов, схожая по целевому назначению с Goodfin" далее по тексту будет обозначаться как **HUB**.

Для кого предназначена данная статья?

- для владельцев HUB, которые хотят для своих клиентов/агентов расширить спектр банковских услуг и предложений от банков, а также предоставить возможность получать банковские продукты клиентам, продолжая работать только в HUB.

4.4.1 1. Порядок обмена интеграционными сообщениями в рамках бизнес-процесса получения, обработки заявки и выдачи Продукта

Для просмотра в увеличенном масштабе нажмите на картинку левой кнопкой мыши.



Комментарии к рис. выше:

1. На стороне HUB должны быть определены endpoints, описанные в <https://sandbox.goodfin.ru/docs/shb-open-api/v1/index.html>:

для получения информации по заявке сервисом:

GET /api/v1/document/getdocumentfile

POST /ad/v1/external/dealapplication/byid (наименование эндпоинта может быть иным, формироваться по правилам HUB. Сотрудники HUB должны передать наименование для регистрации в адаптере Goodfin)

для получения сообщений вида CommonResponse, ChangeDealAppStatEvent от сервисов в точку интеграции HUB:

POST /ad/v1/external/openapi/in (наименование эндпоинта может быть иным, формироваться по правилам HUB. Сотрудники HUB должны передать наименование для регистрации в адаптере Goodfin)

2. HUB должен уметь передавать заявки в установленном формате, концепция которого описана в I. Подготовительные работы по интеграции с применением API (см. раздел "Концепция организации передаваемых данных, используемая в API").
3. Шаг №3 показан укрупненно и состоит из более мелких шагов: автопроверка (после которой заявка может вернуться); запрос на скачивание документов в составе заявки и, если все автоматические проверки пройдены, происходит смена статуса на "Готова к подписанию" или др. (зависит от конкретного банковского сервиса).
4. Если пользователям HUB по аналогии с пользователями Goodfin будет предоставлена возможность отправить запрос на отзыв заявки на любом шаге бизнес-процесса, то сервис может отказать в отзыве заявки или вернуть статус "Отозвана клиентом".
5. В процессе рассмотрения заявки в сервисе возможны различные шаги и статусы. В зависимости от жизненного цикла заявки с сервиса могут приходиться статусы типа:
"На рассмотрении";
"Направлен запрос клиенту";
"Формируется предложение";
"Формирование документов предложения";
"Проверка документов";
"Повторная проверка документов";
"Проект на согласовании";
"Запрос от клиента на доработку";
"Предложение принято";
"Предложение оплачено";
"Поиск предложений";
"Предварительное предложение принято"
и др.
6. На рис. желтыми прямоугольниками обозначены задачи, которые выполняет пользователь; зелёными - сервисные задачи, выполняемые системами; оранжевыми - задачи, которые на текущий момент выполняет пользователь с применением сквозной авторизации и переходом в банковский сервис (будут вести работы, чтобы уйти от сквозной авторизации).
7. Все интеграционные взаимодействия между HUB и банковскими сервисами проходят через адаптер Goodfin, который выполняет необходимые преобразования сообщений под требования определенных банковских сервисов, тем самым обеспечивая для HUB единообразие в формате общения с различными банковскими сервисами.

4.4.2 2. Плюсы и минусы схемы интеграционного взаимодействия для владельцев HUB

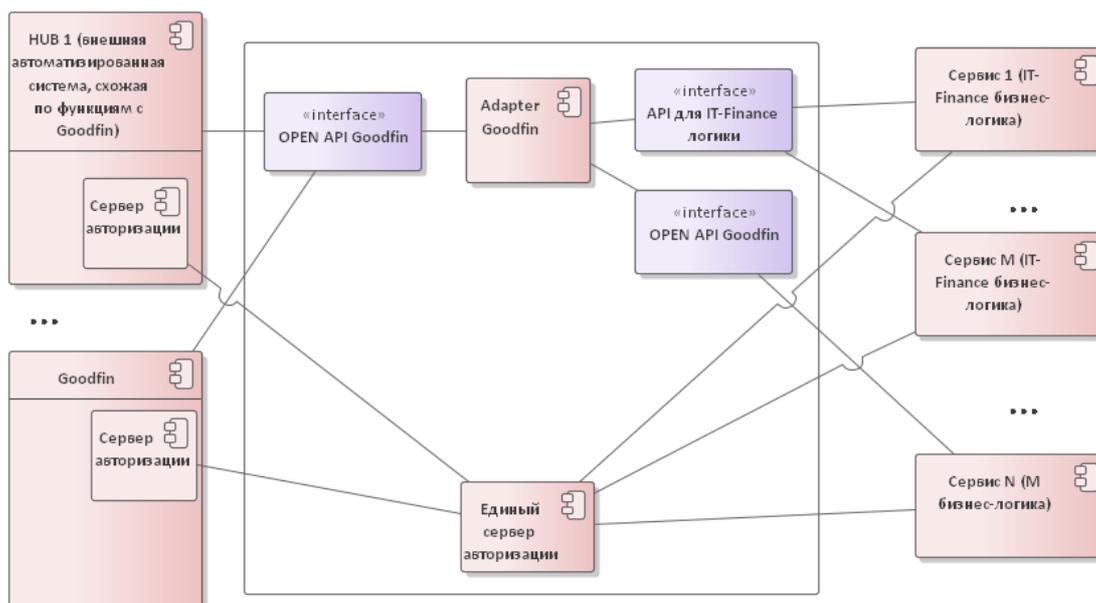
Плюсы

1. Клиенты/агенты HUB получают доступ ко всем банковским сервисам и их предложениям по продуктам, которые подключены к Goodfin.
2. Владельцам HUB достаточно поддерживать только один формат API от Goodfin, прочие издержки по работе на уровне API с банковскими сервисами берет на себя "Адаптер Goodfin" и команда поддержки Goodfin.
3. Данные клиентов/агентов остаются у владельца HUB и "не перетекают" в Goodfin.

Минусы

1. Владельцам HUB необходимо выполнить доработки пользовательского интерфейса: добавить возможность показывать клиентам предложения от нескольких банков; показывать текущее состояние заявок по нескольким банкам; в случае, если клиент подал несколько заявок в разные банки и в одном из них получил продукт, то автоматически отзываться прочие заявки; добавить возможность отправки заявки с исправленными данными, если по предыдущей получен отказ; выполнять и поддерживать настройки обязательности полей для каждого сервиса; фильтровать предложения от банковских сервисов, чтобы снизить для клиента "стресс" от получения отказов от банков, если заведомо клиент не подходит под требования банка.
2. Многие банковские сервисы требуют выполнения операций по подписанию заявки, принятию предложения, докреплению доп. документов и пр. на стороне интерфейса банка. Следовательно, владельцам HUB может потребоваться реализация сервера авторизации на базе OAuth для организации доступа клиентов со сквозной авторизацией к интерфейсу банков.

4.4.3 3. Укрупненная схема компонентов



4.4.4 4. Что нужно сделать владельцам HUB?

Примечание:

1. На текущий момент средствами OPEN API Goodfin отсутствует возможность запросить: "предварительный тариф по выбранному банковскому продукту(ам)", "получить список недостающих данных (данные карточки клиента, документы, финансовые показатели), которые обязательны в составе отправляемой заявки для выбранного банковского продукта(ов)"; "получить актуальный на момент обращения список подключенных банковских сервисов к Goodfin". Данные функции реализованы только для заявок, которые были сформированы из данных, хранящихся в самой системе Goodfin.
2. Банковские сервисы на базе IT-Finance (а также ITCredit) на текущий момент должны будут выполнить настройки по возможности приема заявок от HUB, отличного от Goodfin. Чтобы исключить необходимость регистрации HUB во всех подключенных сервисах

командой Goodfin должна быть проведена доработка единого сервера авторизации.
 3. Прочие банковские сервисы могут потребовать/ не потребовать другие спец. настройки или доработки. Предсказать сложно, так как процесс подключения новых систем динамичный и может быть растянут во времени по независящим от команды Goodfin причинам.

Виды работ	Требования к HUB	Перечень доработок на адаптере Goodfin	Перечень доработок в банковских сервисах, которые подключены к Goodfin	Прочие организационные требования к команде Goodfin
Доработка тестового окружения в рамках sandbox.goodfin		<p>Проверить возможности sandbox.goodfin адаптера принимать сообщения от HUB, отличного от Goodfin (готовность его работы с несколькими HUB).</p> <p>Для проверки подготовить postman - проект или soapUI - проект, имитирующий отправку заявок от имени HUB.</p>	<p>Подключить в ограниченном режиме тестовую инстанцию без прав доступа со стороны банковских работников и регистрации извне, чтобы проверить в последующем, что:</p> <ul style="list-style-type: none"> - заявка дошла; - заявка провалидирована; - документы скачаны; - статус заявки перешел в "Готова к подписанию"; - клиент HUB может перейти по ссылке в сервис (при этом пользователь создан в сервисе на основании данных сервера 	<p>Дописать статью с правилами подключения к тестовой среде или организовать передачу необходимых данных для работы с тестовой средой по закрытому каналу.</p>

Виды работ	Требования к HUB	Перечень доработок на адаптере Goodfin	Перечень доработок в банковских сервисах, которые подключены к Goodfin	Прочие организационные требования к команде Goodfin
			авторизации HUB).	
Поддержка в HUB формата заявки согласно требованиям OPEN API Goodfin	Согласно описанию концепции передаваемых данных в 1. Подготовительные работы по интеграции с применением API реализовать метод(ы) маппинга данных из заявки в формате HUB в формат Goodfin..			
Реализация возможности поддержки OPEN API Goodfin	<p>На стороне HUB должны быть определены эндпоинты, описанные в https://sandbox.goodfin.ru/docs/shb-open-api/v1/index.html:</p> <p>1. Для получения информации по заявке сервисом:</p> <p>1.1. GET /api/v1/document/getdocumentfile</p> <p>1.2. POST /ad/v1/external/dealapplication/byid</p> <p>Примечание: наименование эндпоинта может быть иным, формироваться по правилам HUB. Сотрудники HUB должны передать наименование для регистрации в адаптере Goodfin.</p> <p>2. Для получения сообщений вида CommonResponse, ChangeDealAppStateEvent от сервисов в точку интеграции HUB:</p> <p>2.1. POST /ad/v1/external/openapi/in</p>			

Виды работ	Требования к HUB	Перечень доработок на адаптере Goodfin	Перечень доработок в банковских сервисах, которые подключены к Goodfin	Прочие организационные требования к команде Goodfin
	Примечание: наименование эндпоинта может быть иным, формироваться по правилам HUB. Сотрудники HUB должны передать наименование для регистрации в адаптере Goodfin.			
Унификация порядка скачивания документов в под требования API Goodfin	Предоставить эндпоинт с правами доступа, по которому сервисы будут обращаться за скачиванием документов в составе заявки в рамках поддержки метода getdocumentfile.	Опубликовать требования к скачиванию документов (эндпоинты скачивания файлов, получения информации о заявке). В составе заявки передавать информацию о path к скачиванию каждого документа.	Использовать пути из заявки для скачивания документов вместо текущего пути скачивания по умолчанию, соответствующего Goodfin.	
Возможность клиентов HUB переходить в сервис по ссылкам с авторизацией через HUB, для выполнения операций	Реализовать сервер авторизации на базе OAuth, выполняющий укрупненно следующие операции: - клиент HUB обращается к странице сервиса, которая требует авторизации; - сервис перенаправляет запрос в единый сервер авторизации (ЕСА); - ЕСА отправляет запрос к серверу авторизации HUB; - после успешной авторизации HUB выпускает токен для	Доработка единого сервера авторизации.		

Виды работ	Требования к HUB	Перечень доработок на адаптере Goodfin	Перечень доработок в банковских сервисах, которые подключены к Goodfin	Прочие организационные требования к команде Goodfin
	доступа клиенту в банковский сервис.			
Интеграционное тестирование в рамках sandbox.goodfin	<p>Подготовить генератор тестовых заявок средствами функциональности HUB для проверки качества проведенных подготовительных работ на HUB.</p> <p>Сообщить данные для адаптера Goodfin.</p>	<p>На адаптере провести работы по подключению.</p> <p>Написать интеграционный тест, который получает сгенерированные заявки от HUB и конвертирует их в логику сервисов с отправкой в тестовый сервис, а также пробросом ответов в HUB.</p>		
Возможность клиентам HUB переходить по ссылкам в сервисы с авторизацией и без для выполнения различных действий: подписание заявки в сервисе; ответ на доп.запрос(ы) документо в или исправление данных заявки; принятие предложения; прикрепление	Поддерживать различные шаблоны формирования внешних ссылок, которые реализованы в банковских сервисах.		На базе сервисов, отличных от IT-Finance могут потребоваться доработки по предоставлению внешней ссылки для подписания, формат которой надо будет передать разработчикам HUB для поддержки.	Организовать передачу актуальных данных по правилам формирования внешних ссылок каждого сервиса.

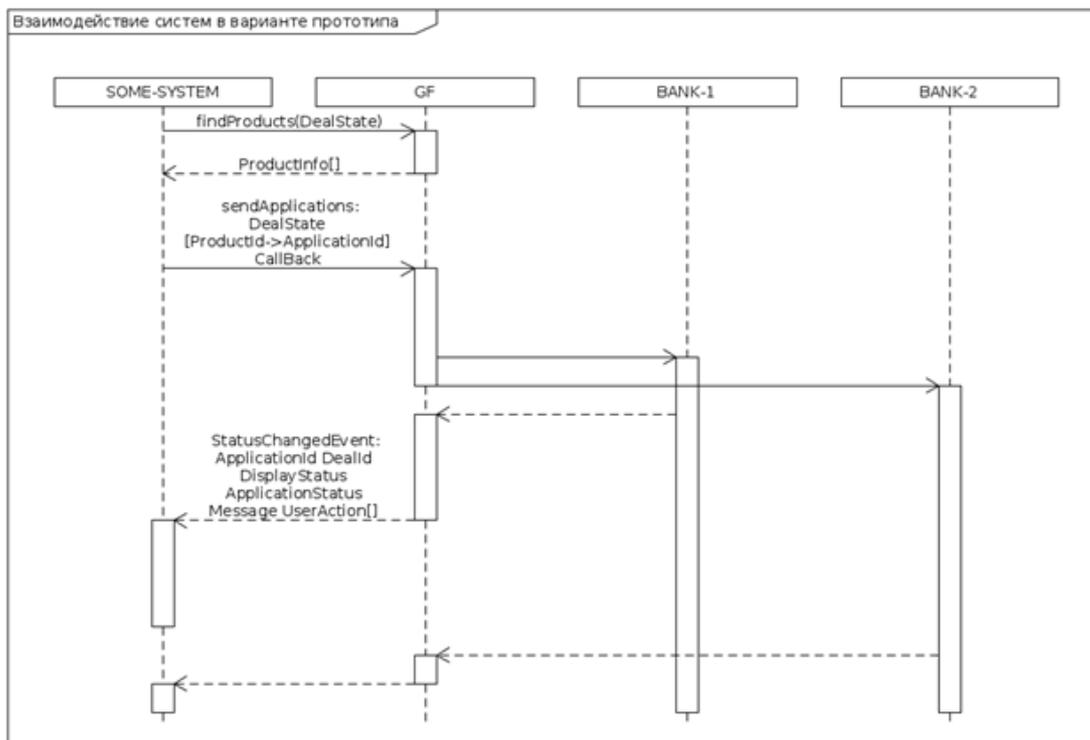
Виды работ	Требования к HUB	Перечень доработок на адаптере Goodfin	Перечень доработок в банковских сервисах, которые подключены к Goodfin	Прочие организационные требования к команде Goodfin
платежного документа; скачивание готовых документов в при выпуске банковского продукта.				
Добавление возможности клиентам на уровне UI HUB работать с сервисами	<p>Выполнить доработки пользовательского интерфейса:</p> <ol style="list-style-type: none"> 1. добавить возможность показывать клиентам предложения от нескольких банков; показывать текущее состояние заявок по нескольким банкам; 2. в случае, если клиент подал несколько заявок в разные банки и в одном из них получил продукт, то автоматически отзываться прочие заявки; добавить возможность отправки заявки с исправленными данными, если по предыдущей получен отказ; 3. выполнять и поддерживать настройки обязательности полей для каждого сервиса; 4. фильтровать предложения от банковских сервисов, если клиент не подходит под требования банковского сервиса. 			<p>Организовать передачу актуальных данных:</p> <ul style="list-style-type: none"> - список активных сервисов; - список активных продуктов; - маппинг статусов сервисов; - маппинг ошибок сервисов.
Поддержка в актуальном состоянии				<p>При "ручном" способе поддержки обновлять раздел "Единый"</p>

Виды работ	Требования к HUB	Перечень доработок на адаптере Goodfin	Перечень доработок в банковских сервисах, которые подключены к Goodfin	Прочие организационные требования к команде Goodfin
справочников со списком подключенных сервисов и продуктов				<p>указатель по справочникам" раз в месяц.</p> <p>Разовые работы: доопределить все справочные данные, которые Goodfin должен предоставить владельцам HUB для организации процесса интеграции; выложить/актуализировать в разделе "Единый указатель по справочникам".</p>

4.5 Структура API методов для работы с внешними системами

- [Общий порядок взаимодействия систем](#)
- [Получение списка обязательных полей](#)
 - [Получение обязательных полей по типу клиента](#)
 - [Получение обязательных полей по типу продукта](#)
- [Поиск продуктов на основе имеющихся на текущий момент данных](#)
- [Информация о подходящих продуктах и требования по дальнейшему заполнению](#)
 - [Описание полей ответа](#)
- [Отправка заявки в сервисы с указанием продуктов](#)
- [Информация о результате принятия заявки в работу](#)
 - [Структура сообщения об успешном принятии заявки в работу](#)
 - [Структура сообщения об ошибке](#)
- [Сообщения отправляемые при изменении статуса заявок](#)
 - [Возможные значения applicationStatus:](#)
- [Подписание заявки на стороне внешней системы](#)
 - [Список возможных алгоритмов](#)

4.5.1 Общий порядок взаимодействия систем



4.5.2 Получение списка обязательных полей

Для получения списка первичных обязательных полей без привязки к конкретным экземплярам используются следующие API-методы:

4.5.2.1 Получение обязательных полей по типу клиента

POST /api/v1/service/profilebasefield/list

Используется для получения первичных обязательных полей зависимых от типа клиента (ФЛ или ИП/ЮЛ).

Нажмите, чтобы посмотреть/скрыть пример

Пример тела запроса:

```
{
  "clientType": "COMPANY"
}
```

где clientType может иметь следующие значения:

- "INDIVIDUAL" - физическое лицо;
- "COMPANY" - юр.лицо/ИП.

Пример получаемого ответа:

```

{
  "result": [
    "regDate",
    "baseOkved",
    "okopf",
    "taxSystem",
    "legalType",
    "legalAddress",
    "legalAddress.city",
    "legalAddress.house",
    "legalAddress.isOwned",
    "legalAddress.postCode",
    "legalAddress.region",
    "legalAddress.street",
    "factAddress",
    "factAddress.city",
    "factAddress.house",
    "factAddress.isOwned",
    "factAddress.postCode",
    "factAddress.region",
    "factAddress.street",
    "sourceInfo",
    "sourceInfo.headCompany",
    "sourceInfo.hasBranches"
  ],
  "fetchFields": {}
}

```

4.5.2.2 Получение обязательных полей по типу продукта

POST /api/v1/service/productbasefield/list

Используется для получения первичных обязательных полей зависимых от типа продукта.

Нажмите, чтобы посмотреть/скрыть пример

Пример тела запроса:

```

{
  "productType": "BG"
}

```

Возможные значения productType см. в справочнике [product types](#).

Пример получаемого ответа:

```
{
  "result": [
    "BG:productDealState.amount",
    "BG:productDealState.tender.federalLaw",
    "BG:productDealState.customer.region",
    "BG:productDealState.startDate",
    "BG:productDealState.endDate",
    "BG:productDealState.deliveryIsRequired",
    "BG:productDealState.termDays"
  ],
  "fetchFields": {}
}
```

4.5.3 Поиск продуктов на основе имеющихся на текущий момент данных

POST /api/v1/deal/findProducts

Пример сообщения: [findProducts.json](#)

4.5.4 Информация о подходящих продуктах и требования по дальнейшему заполнению

Ответ, получаемый при выполнении **findProducts**, содержит список доступных продуктов

Нажмите, чтобы посмотреть/скрыть пример

```

{
  "result": [
    {
      "product": "7ab44246-4607-41f4-8272-8b2ed290e210:beff7718-c72b-43be-b7c6-aa58b1df3cca",
      "serviceCompanyErrors": [],
      "serviceProductErrors": [
        {
          "validator": {
            "field": "BG:productDealState.tender.enforceAmount",
            "type": "PRODUCT_FIELD_REQUIRED"
          },
          "count": 1
        }
      ],
      "serviceMissingClientDocTypes": [
        {
          "docType": "EXTRACT_FROM_LEGAL_PERSON_REGISTRY",
          "paramGroups": []
        }
      ],
      "serviceMissingDealDocTypes": [],
      "serviceMissingIndicators": [],
      "documentTooltips": [
        {
          "tooltip": null,
          "docTypes": [
            "HEAD_PASSPORT_COPY",
            "EXTRACT_FROM_LEGAL_PERSON_REGISTRY",
            "CHARTER",
            "HEAD_APPOINTMENT_PROTOCOL",
            "FINANCE_REPORT_LAST_QUARTER",
            "ANALYTICAL_BALANCE_LAST_REPORTING_YEAR"
          ]
        }
      ],
      "financeIndicatorTooltips": [],
      "tariffCompanyErrors": [],
      "tariffProductErrors": []
    }
  ],
  "fetchFields": {
    "Product": {
      "7ab44246-4607-41f4-8272-8b2ed290e210:beff7718-c72b-43be-b7c6-aa58b1df3cca": {
        "id": "7ab44246-4607-41f4-8272-8b2ed290e210:beff7718-c72b-43be-b7c6-aa58b1df3cca",
        "name": "Банковская гарантия на исполнение для ЮЛ банка ДЖОЛВЖД (ред.10.12.20)",
        "shortName": "БГ на исполнение, ЮЛ (ред.10.12.20, ДЖОЛВЖД)",
        "productType": "BG",
        "description": null,
        "state": "1",
        "author": "ServiceContactParticipant:806b9fea-0d07-4f18-bd9d-0ca3c1e9a346",
        "createDateTime": "2020-12-10T05:59:47.746426",
        "modifier": "ServiceContactParticipant:806b9fea-0d07-4f18-bd9d-0ca3c1e9a346",
        "modifiedDateTime": "2020-12-28T06:27:48.044206"
      }
    }
  }
}

```

4.5.4.1 Описание полей ответа

Ключ	Описание	Прим.
product	Содержит ссылку на fetchField Product	
serviceCompanyErrors	Содержит информацию о необходимых для заполнения полях продукта. Перечень возможных значений см. в справочнике company_fields	
serviceProductErrors	Содержит информацию о необходимых для заполнения полях продукта. Перечень возможных значений см. в справочнике product_types_fields	
serviceMissingClientDocTypes	Содержит список недостающих документов клиента. Перечень возможных значений см. в справочнике docs_types	
serviceMissingDealDocTypes	Содержит список недостающих обязательных документов о сделке.	
serviceMissingIndicators	Содержит список достающих обязательных фин.показателей. Перечень возможных значений см. в справочнике finance_fields	
documentTooltips	Содержит список документов необходимых при заполнении "либо документы, либо фин.показатели"	
financeIndicatorTooltips	Содержит список фин.показателей необходимых при заполнении "либо документы, либо фин.показатели"	
tariffCompanyErrors		
tariffProductErrors		

4.5.5 Отправка заявки в сервисы с указанием продуктов

POST /api/v1/deal/send_applications

При отправке заявок указывается список целевых продуктов, и идентификаторы будущих заявок для соответствующего продукта. В dealState передается та же структура, что и ранее передавалась в /api/v1/deal/findProducts

Нажмите, чтобы посмотреть/скрыть пример

```

{
  "toProducts": [
    {
      "productId": "7ab44246-4607-41f4-8272-8b2ed290e444:179f3310-9827-44a2-913a-515cf2c43721",
      "applicationId": "de74bb16-7d03-420f-9c84-f034f249a1b0"
    }
  ],
  "cbUrl": "https://some_host.com/cb"
  "dealState": <...>
}

```

В ближайшем будущем планируется вместо productId принимать токен, который будет возвращаться из findProducts. Так же вероятно замена передачи данных в dealState на указание URI, по которому можно получить данные, с поддержкой указания требуемых секций данных. Эти преобразования дадут возможность применить стиль REST с применением HATEOAS.

4.5.6 Информация о результате принятия заявки в работу

Система банка, получив заявку, делает первичную ее валидацию, по результатам которой она отправляет сообщение о успешном или провалившемся результате принятия заявки в обработку.

В атрибуте targetObjectId приходит идентификатор заявки, по которой пришел результат.

Нажмите, чтобы посмотреть/скрыть пример

4.5.6.1 Структура сообщения об успешном принятии заявки в работу

```

{
  "originatorMsgType": "AddDealApplication",
  "originatorMsgId": "a45f8520-b132-4ab4-82ce-562affdbed43",
  "targetObjectId": "333a7c64-973d-40ce-8fb5-18f228d89b8f",
  "result": "SUCCESS",
  "error": null
}

```

4.5.6.2 Структура сообщения об ошибке

```
{
  "originatorMsgType": "AddDealApplication",
  "originatorMsgId": "e8eb698b-0dea-4fd7-a705-b254eb53e06f",
  "targetObjectId": "333a7c64-973d-40ce-8fb5-18f228d89b8f",
  "result": "ERROR",
  "error": {
    "id": "adbab53c-11aa-43c8-b9e1-8960a03d8ee0",
    "errorCode": "MY_ERROR_STATUS_1",
    "message": "Some message",
    "params": {
      "MY_PARAM_3": "VALUE3"
    }
  }
}
```

4.5.7 Сообщения отправляемые при изменении статуса заявок

Сообщение будет приходить на ранее указанный адрес обратного вызова.

В сообщении присутствует идентификатор заявки, который был ранее указан при "applicationId".

Для повышения уровня безопасности коммуникации, и минимизации согласования авторизационных данных, данное сообщение планируется передавать в виде подписанного JWT, чтобы принимающая сторона смогла проверять достоверность полученных данных без необходимости какой-либо дополнительной аутентификации при обратном вызове.

Нажмите, чтобы посмотреть/скрыть пример

```
{
  "applicationId": "de74bb16-7d03-420f-9c84-f034f249a1b0",
  "displayStatus": "Готова к подписанию",
  "applicationStatus": "SENT_TO_SERVICE",
  "message": null,
  "userActions": [
    {
      "actionType": "CLIENT_INTERNAL_SIGN",
      "label": "Подписать",
      "description": "Подписать заявку",
      "documentsToSignUrl": "/deal/documents_to_sign/de74bb16-7d03-420f-9c84-f034f249a1b0"
    },
    {
      "actionType": "AGENT_GET_INTERNAL_SIGN_URL",
      "label": "Получить ссылку",
      "description": "Получить ссылку для подписания клиентом",
      "documentsToSignUrl": "/deal/documents_to_sign/de74bb16-7d03-420f-9c84-f034f249a1b0"
    }
  ]
}
```

Перечень возможных значений dealApplicationStatus см. в справочнике [dealApplicationStatuses](#).

Перечень возможных userActions см. в справочнике [userActions](#).

На текущий момент в прототипе не реализован проброс API для подписания документов, чтобы заявка могла считаться подписанной. Необходимые данные для получения и подписания документов предполагается указывать в виде соответствующих URI внутри каждого элемента userActions. Так же возможна реализация ресурса, для получения текущего состояния заявки, чтобы внешняя система могла в любой момент получить актуальное состояние заявки, и возможные действия по ней.

Не все требования к полям сервисы декларируют в своих продуктах, некоторые проверки они выполняют у себя самостоятельно, поэтому возможны статусы отклонения заявки.

Нажмите, чтобы посмотреть/скрыть пример

Пример отклонения заявки

```
{
  "applicationId": "33c9e43e-82fd-43ea-8fbf-4d54fa4fa97f",
  "displayStatus": "Ошибка отправки",
  "applicationStatus": "REJECTED_BY_SERVICE",
  "message": "У компании-агента не заполнен ОКПО в карточке компании.\r\nУ
компании-агента не заполнен базовый ОКВЭД в карточке компании.\r\nУ
компании-агента не заполнен юридический адрес в карточке компании.\r\nУ
компании-агента не заполнен тип организации в карточке компании.\r\nУ
компании-агента не заполнен фактический адрес в карточке компании.",
  "userActions": []
}
```

4.5.7.1 Возможные значения applicationStatus:

Значение	Описание
SENT_TO_SERVICE	Отправлена в сервис
CLOSED_BY_SYSTEM	Закрыта системой. Терминальный статус
REJECTED_BY_SERVICE	Отклонена сервисом (банком). Терминальный статус
PRODUCT_ISSUED	Выпущен продукт. Терминальный статус

4.5.8 Подписание заявки на стороне внешней системы

При получении события о смене статуса заявки, если в данном сообщении так же присутствует UserAction с типом CLIENT_INTERNAL_SIGN, система-отправитель должна выполнить действие подписания заявки. Процесс подписания заявки организован через подписание ее документов, куда входят как документы, которые были ранее присланы при подаче заявки, так и сгенерированные на стороне банка документы (например, анкета клиента).

Получить список документов, требуемых на подпись можно вызвав URL, указанный в атрибуте userActions[].documentsToSignUrl. В результате данного вызова будет получен список документов, с их хешами разных алгоритмов, для возможности подписания, и URL для скачивания (если пользователю или системе отправителю, потребуется содержимое документов).

4.5.8.1 Список возможных алгоритмов

- GOST-3411,
- GOST-3411-2012-512,
- GOST-3411-2012-256,
- SHA-256.

Так же в теле ответа приходит URL, на который необходимо отправлять результаты подписи. Данный URL определен в атрибуте result.signUrl.

Нажмите, чтобы посмотреть/скрыть пример списка документов на подписание

```

{
  "result": {
    "signUrl": "/deal/servicedocument/quick/sign",
    "documents": [
      {
        "id": "684369",
        "dealApplicationId": "333a7c64-973d-40ce-8fb5-18f228d89b8f",
        "documentTypeName": "Копии паспортов Бенефициарных владельцев",
        "description": null,
        "files": [
          {
            "id": "NzY5Mjk1OkF0dGFjaG1lbnRz",
            "filename": "mtsbank_logored_ru_rgb.jpg",
            "url": "http://192.168.1.138:8080/bidders/dg/remote_download/Attachments/769295/7b2def6f-ecb9-49ad-a32a-e5327e889f6d/",
            "comment": null,
            "hashes": [
              {
                "algorithm": "GOST-3411-2012-512",
                "hash": "{
          "result": {
            "signUrl": "/deal/servicedocument/quick/sign",
            "documents": [
              {
                "id": "684369",
                "dealApplicationId": "333a7c64-973d-40ce-8fb5-18f228d89b8f",
                "documentTypeName": "Копии паспортов Бенефициарных владельцев",
                "description": null,
                "files": [
                  {
                    "id": "NzY5Mjk1OkF0dGFjaG1lbnRz",
                    "filename": "mtsbank_logored_ru_rgb.jpg",
                    "url": "http://192.168.1.138:8080/bidders/dg/remote_download/Attachments/769295/7b2def6f-ecb9-49ad-a32a-e5327e889f6d/",
                    "comment": null,
                    "hashes": [
                      {
                        "algorithm": "GOST-3411-2012-512",
                        "hash": "<OMMITED>"
                      },
                      {
                        "algorithm": "GOST-3411-2012-256",
                        "hash": "<OMMITED>"
                      }
                    ]
                  }
                ],
              },
              {
                "id": "NzY5Mjk2OkF0dGFjaG1lbnRz",
                "filename": "test.docx",
                "url": "http://192.168.1.138:8080/bidders/dg/remote_download/Attachments/769296/7b2def6f-ecb9-49ad-a32a-e5327e889f6d/",
                "comment": null,
                "hashes": [
                  {
                    "algorithm": "GOST-3411-2012-512",
                    "hash": "<OMMITED>"
                  },
                  {
                    "algorithm": "GOST-3411-2012-256",

```

```

        "hash": "<OMMITED>"
      }
    ]
  },
  "fetchFields": {}
}

```

Нажмите, чтобы посмотреть/скрыть пример сообщения с результатом подписания

```

{
  "signatures": [
    {
      "dealApplicationId": "333a7c64-973d-40ce-8fb5-18f228d89b8f",
      "documentSignatures": [
        {
          "serviceDocumentFileId": "Mjk4NjgxOkdlbmVyYXRlZERvY3VtZW50cw",
          "signature": "<OMMITED>"
        },
        {
          "serviceDocumentFileId": "Mjk4NjgyOkdlbmVyYXRlZERvY3VtZW50cw",
          "signature": "<OMMITED>"
        },
        {
          "serviceDocumentFileId": "MTEzMDC0MTpBdHRhY2htZW50cw",
          "signature": "<OMMITED>"
        }
      ]
    }
  ]
}

```

5 Для банковских сервисов, подключаемых к Goodfin

5.1 Оглавление

5.2 Введение

Настоящий документ представляет собой руководство для разработчиков со стороны банковского сервиса, ответственных за подготовку банковской системы к интеграции с системой Goodfin.

Также документ содержит руководство для специалистов по банковским продуктам, описывающее порядок и особенности добавления/изменения банковских продуктов в системе Goodfin.

5.2.1 Краткое описание стадий работ и возможностей

I. Для начала интеграционного взаимодействия банковского сервиса с системой Goodfin проводятся работы силами разработчиков банковского сервиса по поддержке API, предоставленного системой Goodfin.

На базе "песочницы" (тестовое окружение <https://sandbox.goodfin.ru/>) разработчики банковского сервиса реализуют и тестируют интеграцию с системой Goodfin.

Подробнее см. [I. Подготовительные работы по интеграции с применением API](#)

Также вы можете воспользоваться одним из готовых проектов для знакомства с процессом интеграции. См. [Quickstart. Примеры интеграции с Goodfin](#)

Также вы можете предварительно ознакомиться с порядком обмена интеграционными сообщениями в процессе отправки заявки из Goodfin и обработки заявки в банковском сервисе до момента выпуска банковского продукта.

См. [Порядок обмена интеграционными сообщениями в рамках бизнес-процесса получения, обработки заявки и выдачи Продукта по схеме интеграции "Goodfin" <-> "Адаптер Goodfin" <-> "Банковский сервис"](#)

Базовое описание см. в:

- методы и типы регистрации/обновления сведений о подключении сервиса: <https://sandbox.goodfin.ru/docs/shb-open-api/v1/index.html>

- методы и типы описания бизнес-процесса отправки заявок в сервисы и обмена сообщениями с сервисом: <https://sandbox.goodfin.ru/docs/bl-open-api/v1/index.html>

II. Разработка и тестирование продуктов на базе тестового окружения Goodfin.

Используя тестовое окружение с UI и данными тестовых учетных записей разработчики банковского сервиса реализуют и тестируют банковские продукты для последующего импорта их в действующую систему Goodfin.

Подробнее см. [II.1. Создание продукта в виде конфигурационного файла yam1](#), также см. [II.2. Создание тарификатора в виде конфигурационного файла yam1](#), далее см. .

III. Получение доступа в личный кабинет поставщика сервиса действующей системы Goodfin.

Регистрация и первичная настройка выполняется сотрудниками владельца системы, данные для входа в кабинет передается представителю поставщика.

Подробнее см. [III. Получение доступа в личный кабинет поставщика сервиса действующей системы Goodfin](#)

IV. Регистрация продукта(ов).

Протестированные на базе тестового окружения Goodfin продукты импортируются в действующую систему Goodfin.

Подробнее см. [IV. Импорт подготовленных продуктов в действующую систему Goodfin](#)

Кроме того, вы можете:

1. Запрос на добавление данных (новых полей, типов документов) в систему, которых нет в списке системы.
2. Оперативное включение/ отключение доступности продукта для выбора участниками системы в рамках сделок.
3. Изменение условий тарифов по предоставляемым продуктам.
4. Аналитика по востребованности продуктов и эффективности работы агентов (в разработке).

Принятые соглашения по оформлению Руководства

1. Кавычками выделяются названия полей и других объектов экранных форм и пр. Наименования разделов и пунктов меню отделяются друг от друга символом →.
2. Для акцентирования внимания на действиях, которые обязательны для выполнения, используются рамки и стрелки красного цвета.
3. Для выделения блоков текста используются специальные средства оформления, представленные ниже:

ПРИМЕЧАНИЕ

Служит для выделения дополнительной информации, в том числе ссылок на фрагменты документации. Следует непосредственно за элементом, к которому оно относится.

ВНИМАНИЕ!

Служит для описания действий, обязательных для выполнения, либо критичных моментов в работе Программы, несоблюдение которых может привести к сбою.

5.3 I. Подготовительные работы по интеграции с применением API

- [Доступ к описанию методов API](#)
- [Концепция организации передаваемых данных, используемая в API](#)
 - [1. Нормализованная форма данных](#)

- [2. Концепция абстрактной ссылки](#)
- [3. Концепция идентификации - композитный идентификатор](#)
- [4. Пример](#)
- [Примерный порядок работ для подготовки к интеграции с Goodfin с использованием песочницы](#)

5.3.1 Доступ к описанию методов API

Базовое описание см. в:

- методы и типы регистрации/обновления сведений о подключении сервиса: <https://sandbox.goodfin.ru/docs/shb-open-api/v1/index.html>

- методы и типы описания бизнес-процесса отправки заявок в сервисы и обмена сообщениями с сервисом: <https://sandbox.goodfin.ru/docs/bl-open-api/v1/index.html>

5.3.2 Концепция организации передаваемых данных, используемая в API

Примеры сформированных тестовых заявок на дату 30/12/2020 со структурой, актуальной для версии 1.19 и выше, по которым можно посмотреть структуру данных:

- для типа продукта Банковская гарантия на исполнение от клиента ЮЛ - [БГ\(И\) ЮЛ.json](#)
- для типа продукта Банковская гарантия на исполнение от клиента ИП - [БГ\(И\) ИП.json](#)
- для типа продукта Банковская гарантия на возврат аванса от клиента ЮЛ - [БГ\(А\) ЮЛ.json](#)
- для типа продукта Банковская гарантия обеспечения гарантийных обязательств от клиента ЮЛ - [БГ\(О\) ЮЛ.json](#)
- для типа продукта Банковская гарантия на участие от клиента ЮЛ - [БГ\(У\) ЮЛ.json](#)
- для типа продукта Тендерный займ от клиента ЮЛ - [ТЗ ЮЛ.json](#)
- для типа продукта Кредит на исполнение контракта от клиента ЮЛ - [КиК ЮЛ.json](#)

5.3.2.1 1. Нормализованная форма данных

В некоторых методах API применена нормализация данных.

В нормализованном виде взаимосвязь между объектами определяется явными ссылками, а сами объекты с данными лежат в специальной области структуры - `fetchFields`.

В структуре блока `fetchFields` объекты распределены по их типу. Общий формат передаваемых данных можно определить следующим образом:

```
{
  "result": {
    "referenceToObject1": "obj1",
    ...
  },
  "fetchFields": {
    "Object1": {
      "obj1": {
        "id": "obj1",
```

```

        "referenceToObject2": "obj2",
        ...
    }
},
"Object2": {
    "obj2": {
        "id": "obj2",
        ...
    }
},
...
}
}

```

5.3.2.2 2. Концепция абстрактной ссылки

Если на уровне структуры данных поле ссылается на объект некой абстракции, то данное поле применяет концепцию абстрактной ссылки. Сама ссылка определяет какого типа объект по его идентификатору, указанному после разделителя.

В следующем примере концепция абстрактной ссылки была применена к полю `"result.referenceToAbstractObject"`. Его содержимое - `"ConcreteObject1:obj1"` указывает на то, что данные объекта необходимо искать по следующему пути: `"fetchFields.ConcreteObject1.obj1"`. Варианты типов ссылки ограничены типом самой ссылки.

```

{
  "result": {
    "referenceToAbstractObject": ["ConcreteObject1:obj1",
"ConcreteObject2:obj2"]
  },
  "fetchFields": {
    "ConcreteObject1": {
      "obj1": {
        "id": "obj1",
        "dataField1": "data111"
      }
    },
    "ConcreteObject2": {
      "obj2": {
        "id": "obj2",
        "dataField2": "data222"
      }
    }
  }
}
}

```

Концепция абстрактной ссылки также применяется для обобщающих проекций. В примере ниже такой вариант показан в `"AbstractProjection"`.

У данного вида объектов идентифицирующим полем является абстрактная ссылка. В остальном обобщающая проекция ничем не отличается от других объектов.

```

{
  "result": {
    "referenceToAbstractObject": ["ConcreteObject1:obj1",
"ConcreteObject2:obj2"]
  },
  "fetchFields": {
    "AbstractProjection": {
      "ConcreteObject1:obj1": {
        "id": "ConcreteObject1:obj1",
        "fld": "43434035848"
      }
    },
  },
}

```

```

    "ConcreteObject2:obj2": {
      "id": "ConcreteObject2:obj2",
      "fld": "54545467676"
    }
  }
}

```

В случаях, когда у объекта есть данные, которые требуется подгружать опционально, используются дополнительные объекты-проекции с дополнительными данными, при этом идентификатор равен идентификатору основного объекта.

```

{
  "result": {
    "referenceToConcreteObject": "obj1"
  },
  "fetchFields": {
    "ConcreteObject1": {
      "obj1": {
        "id": "obj1",
        "dataField1": "data111"
      }
    },
    "AdditionalDataForConcreteObject1": {
      "obj1": {
        "id": "obj1",
        "additionalData": "vfgsdljkg987ihdfj"
      }
    }
  }
}

```

5.3.2.3 3. Концепция идентификации - композитный идентификатор

Для композиции объектов используется также отдельная концепция идентификации - композитный идентификатор.

Объекты, которые являются частью композиции, в своем идентификаторе всегда имеют идентификатор своего композита. Таким образом, в примере ниже идентификатор `"comp1:item1"` имеет в своем содержимом и идентификатор своего композита `"comp1"`.

Это дает гарантию, что при пакетной обработке объектов, которые являются частью композита, можно всегда выйти на сам объект композиции. Это дает возможность выбирать все части композиции, фильтруя объекты по их идентификатору функцией `startWith("comp1")`.

Несмотря на то, что сейчас для многих идентификаторов, в том числе и композитных, используется `uuid`, уникальность идентификатора `"CompositeObjectItem"` гарантируется только в пределах его композита(`"comp1"`).

```

{
  "result": {
    "referenceToObject": "comp1"
  },
  "fetchFields": {
    "CompositeObject": {
      "comp1": {
        "id": "comp1",
        "dataField3": "data123"
      }
    },
    "CompositeObjectItem": {
      "comp1:item1": {

```

```

    "id": "comp1:item1",
    "dataField4": "data321"
  },
  "comp1:item2": {
    "id": "comp1:item2",
    "dataField4": "321data"
  }
}
}
}

```

В **"fetchFields"** так же могут возвращаться объекты объединенные общей абстракцией. В этом случае их конкретный тип зависит от одного из полей, который присутствует во всех конкретных типах. На примере ниже таким полем является **"type"**. В зависимости от значения данного поля, будет определено, какие дополнительные поля применимы к данному объекту.

```

{
  "result": {
    "referenceToObject": "obj1"
  },
  "fetchFields": {
    "AbstractObject": {
      "obj1": {
        "id": "obj1",
        "type": "type1",
        "fldType1": 2323
      },
      "obj2": {
        "id": "obj2",
        "type": "type2",
        "fldType2": "djfklsfhjf"
      },
      "obj3": {
        "id": "obj3",
        "type": "type2",
        "fldType2": "ppisaiosifg"
      }
    }
  }
}

```

5.3.2.4 4. Пример

Ниже приведен пример данных заявки в нормализованном виде, часть данных опущена, чтобы сократить объем.

В данном примере поле **"result.client"** применяет концепцию абстрактной ссылки, при этом тип данной ссылки позволяет также ссылаться на объекты типа **"IndividualContactParticipant"**.

Объект **"Person"** является частью композита **"Company"**. Поля **"author"** и **"modifier"** объекта **"Company"** разыменовываются через проекцию **"ModifierParticipant"**.

```

{
  "result": {
    "id": "e452c6e2-449a-4f4b-8da2-dfaecc194e0b",
    "dealNumber": "БГ(И)-6648",
    "currencyType": "1",
    "income": 7206.00,
    "client": "Company:32a45baf-ef78-4459-96a8-1d133cf2b913"
  },
  "fetchFields": {

```

```

"Person": {
  "32a45baf-ef78-4459-96a8-1d133cf2b913:92bd540a-3ce5-4a51-b1a1-5e9bf085f959": {
    "id": "32a45baf-ef78-4459-96a8-1d133cf2b913:92bd540a-3ce5-4a51-b1a1-5e9bf085f959",
    "firstName": "Алексей",
    "middleName": "Владимирович",
    "docSeries": "4607",
    "docType": "PASSPORT",
  }
},
"OktmoCatalog": {
  "23129": {
    "id": "23129",
    "parentId": "23128",
    "code": "17701000001",
    "name": "г Владимир"
  }
},
"Company": {
  "32a45baf-ef78-4459-96a8-1d133cf2b913": {
    "id": "32a45baf-ef78-4459-96a8-1d133cf2b913",
    "oktmo": "23129",
    "legalAddress": "32a45baf-ef78-4459-96a8-1d133cf2b913:0c8ecd6f-d7f7-41d7-b15a-ed850dc24944",
    "author": "IndividualContactParticipant:961b4042-a55d-4b7e-8edf-33da2884e33c",
    "createDateTime": "2019-02-04T14:52:59.865",
    "modifier": "CompanyContactParticipant:29bb5e27-8f2f-4001-afa2-61292ab971eb",
    "modifiedDateTime": "2019-08-02T13:14:28.084"
  }
},
"PersonRole": {
  "32a45baf-ef78-4459-96a8-1d133cf2b913:92bd540a-3ce5-4a51-b1a1-5e9bf085f959:5f5869a4-e6b2-4a5f-b254-9583689954b3": {
    "id": "32a45baf-ef78-4459-96a8-1d133cf2b913:92bd540a-3ce5-4a51-b1a1-5e9bf085f959:5f5869a4-e6b2-4a5f-b254-9583689954b3",
    "roleType": "BENEFICIAR",
    "percent": 100.0000,
    "date": null
  },
  "32a45baf-ef78-4459-96a8-1d133cf2b913:92bd540a-3ce5-4a51-b1a1-5e9bf085f959:f708ce68-386a-445c-87ea-18c826ca05e6": {
    "id": "32a45baf-ef78-4459-96a8-1d133cf2b913:92bd540a-3ce5-4a51-b1a1-5e9bf085f959:f708ce68-386a-445c-87ea-18c826ca05e6",
    "roleType": "BOOKER",
    "fullPostName": "Главный бухгалтер",
    "postDate": null,
    "endDate": null,
    "workYears": null,
    "workPrev": null
  }
},
"CurrencyTypeCatalog": {
  "1": {
    "id": "1",
    "code": "RUB",
    "name": "Российский рубль",
    "codeNum": "643"
  }
}

```

```

    },
    "ModifierParticipant": {
      "CompanyContactParticipant:29bb5e27-8f2f-4001-afa2-61292ab971eb": {
        "id": "CompanyContactParticipant:29bb5e27-8f2f-4001-afa2-61292ab971eb",
        "firstName": "Алексей",
        "middleName": "Владимирович"
      },
      "IndividualContactParticipant:961b4042-a55d-4b7e-8edf-33da2884e33c": {
        "id": "IndividualContactParticipant:961b4042-a55d-4b7e-8edf-33da2884e33c",
        "firstName": "Алексей",
        "middleName": "Владимирович"
      }
    },
    "CompanyContactParticipant": {
      "29bb5e27-8f2f-4001-afa2-61292ab971eb": {
        "id": "29bb5e27-8f2f-4001-afa2-61292ab971eb",
        "firstName": "Алексей",
        "middleName": "Владимирович",
        "userEmail": "alex@example.com",
        "company": "32a45baf-ef78-4459-96a8-1d133cf2b913",
        "verificationType": "ЕСР"
      }
    }
  }
}

```

5.3.3 Примерный порядок работ для подготовки к интеграции с Goodfin с использованием песочницы

5.3.4 Quickstart. Примеры интеграции с Goodfin

Для начала знакомства с процессом интеграции Вы можете воспользоваться одним из проектов:

1. Пример интеграции с Goodfin на kotlin;
2. Пример интеграции с Goodfin средствами SOAP UI.

Для выбора проекта перейдите по [ссылке](#).

5.3.5 Для начала взаимодействия с тестовым окружением необходимо зарегистрировать сервис.

Для сервиса будет сгенерирована учетная запись (логин/пароль OAuth клиента) и идентификатор сервиса. Логин и идентификатор сервиса совпадают.

В песочнице метод `/api/v1/sandbox/register-service` эмулирует создание записи о новом подключаемом сервисе в действующей системе Goodfin.

Также после выполнения метода `/api/v1/sandbox/registerservice` в песочнице: создается пользователь для обслуживания сервиса и пользователь со стационарным клиентом для данного сервиса, которых можно использовать для тестирования и проверки настроек продуктов (согласно [II.3. Тестирование и доработка банковского продукта](#)).

Логины пользователей создаются по правилам:

`user_<serviceName>`

`ss_<serviceName>`

Пароль у всех пользователей одинаковый - `<password>` из ответа на вызов метода `/api/v1/sandbox/registerservice`.

Например, если получен в ответе:

```
BODY:
{
  "serviceName": "iniq8unrB1",
  "password": "h2KLv0UY4g"
}
```

Тогда логин `user_iniq8unrB1` и пароль `h2KLv0UY4g` будет использоваться для входа в кабинет поставщика сервиса.

Логин `ss_iniq8unrB1` и пароль `h2KLv0UY4g` будет использоваться для входа в кабинет клиента для тестирования настроек продукта.

В действующей системе данный метод не используется. Специалисты системы Goodfin самостоятельно добавляют запись о новом сервисе, генерируют логин и пароль. Передают их представителям сервиса.

Нажмите, чтобы посмотреть/скрыть пример

Пример запроса для песочницы:

```
Method: POST
URL: https://sandbox.goodfin.ru/api/v1/sandbox/registerservice
BODY:
{}
```

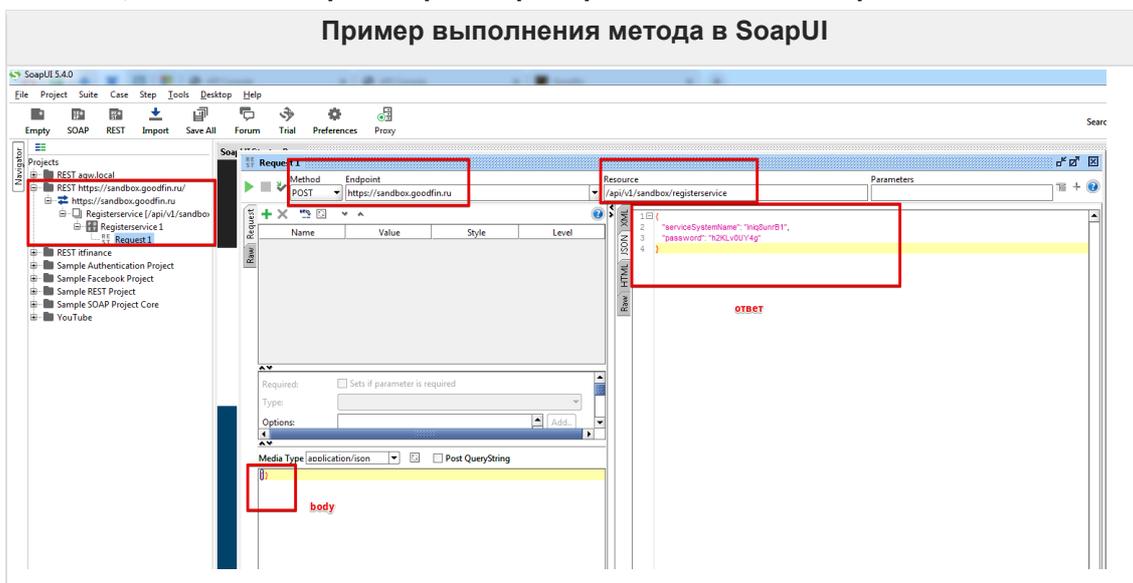
Тот же пример запроса для песочницы в формате HTTP/1.1:

```
POST /api/v1/sandbox/registerservice HTTP/1.1
Host: sandbox.goodfin.ru
Content-Type: application/json
Cache-Control: no-cache
{}
```

Пример получаемого ответа:

```
BODY:
{
  "serviceSystemName": "iniq8unrB1",
  "password": "h2KLV0UY4g"
}
```

Нажмите, чтобы посмотреть/скрыть пример выполнения в SoapUI



5.3.6 Требуется получить токен, чтобы сервис как система мог авторизоваться перед Goodfin как системой.

В песочнице метод `/api/v1/oauth/token`

В действующей системе токен получать аналогичным методом, используя выданные разработчиками Goodfin `serviceSystemName` и `password` после регистрации сервиса.

Подробнее см. стандарт jwt.io по правилам получения и "времени жизни" токена.

ВНИМАНИЕ!

Полученный токен используется далее во всех методах.

Нажмите, чтобы посмотреть/скрыть пример

Пример запроса для песочницы:

где Authorization - это параметр, полученный по данным логина и пароля (basic авторизация).

```
Method: POST
URL: https://sandbox.goodfin.ru/api/v1/oauth/token
HEADERS:
Authorization: Basic aW5pcTh1bnJCMTpoMktMdjBVWTRn
BODY:
grant_type=client_credentials
```


[/api/v1/service/errormapping/update](#) (см. пример в [Шаг 3.2](#)
[Объявление/обновление передаваемых кодов ошибок](#)).

5.3.8 Шаг 3.1 Объявление/обновление параметров статусной модели сервиса

Чтобы система Goodfin могла корректно выводить пользователям статус заявки в сервисе и для определенных статусов показывать действия, которые может выполнить пользователь с заявкой, используется метод [/api/v1/service/statusmapping/update](#)

При получении от сервиса CommonResponse (см. примеры на шаге 4.3) с информацией о измененном статусе заявки, Goodfin будет искать полученный статус в настройках и выводить пользователю: статус заявки в сервисе из значения displayStatus; возможные действия с заявкой согласно userActions; переводить заявку в соответствующий статус на Goodfin (по dealApplicationStatus).

Перечень возможных значений dealApplicationStatus см. в справочнике [dealApplicationStatuses](#).

Перечень возможных userActions см. в справочнике [userActions](#).

Нажмите, чтобы посмотреть/скрыть пример 1

Пример запроса для песочницы:

где Authorization - это значение параметра access_token из ответа, полученного на шаге 2.


```
    "actionType": "CLIENT_SIGN",
    "label": "Подписать",
    "description": "Подписать заявку",
    "template": "https://servicell.com/{APPLICATION_ID}"
  },
  {
    "actionType": "AGENT_GET_SIGN_URL",
    "label": "Получить ссылку",
    "description": "Получить ссылку для подписания клиентом",
    "template": "https://servicell.com/{APPLICATION_ID}"
  }
]
}
}
```

Пример запроса для песочницы в формате HTTP/1.1:


```

{
  "actionType": "CLIENT_SIGN",
  "label": "Подписать",
  "description": "Подписать заявку",
  "template": "https://service11.com/{APPLICATION_ID}"
},
{
  "actionType": "AGENT_GET_SIGN_URL",
  "label": "Получить ссылку",
  "description": "Получить ссылку для подписания клиентом",
  "template": "https://service11.com/{APPLICATION_ID}"
}
]
}
}
}

```

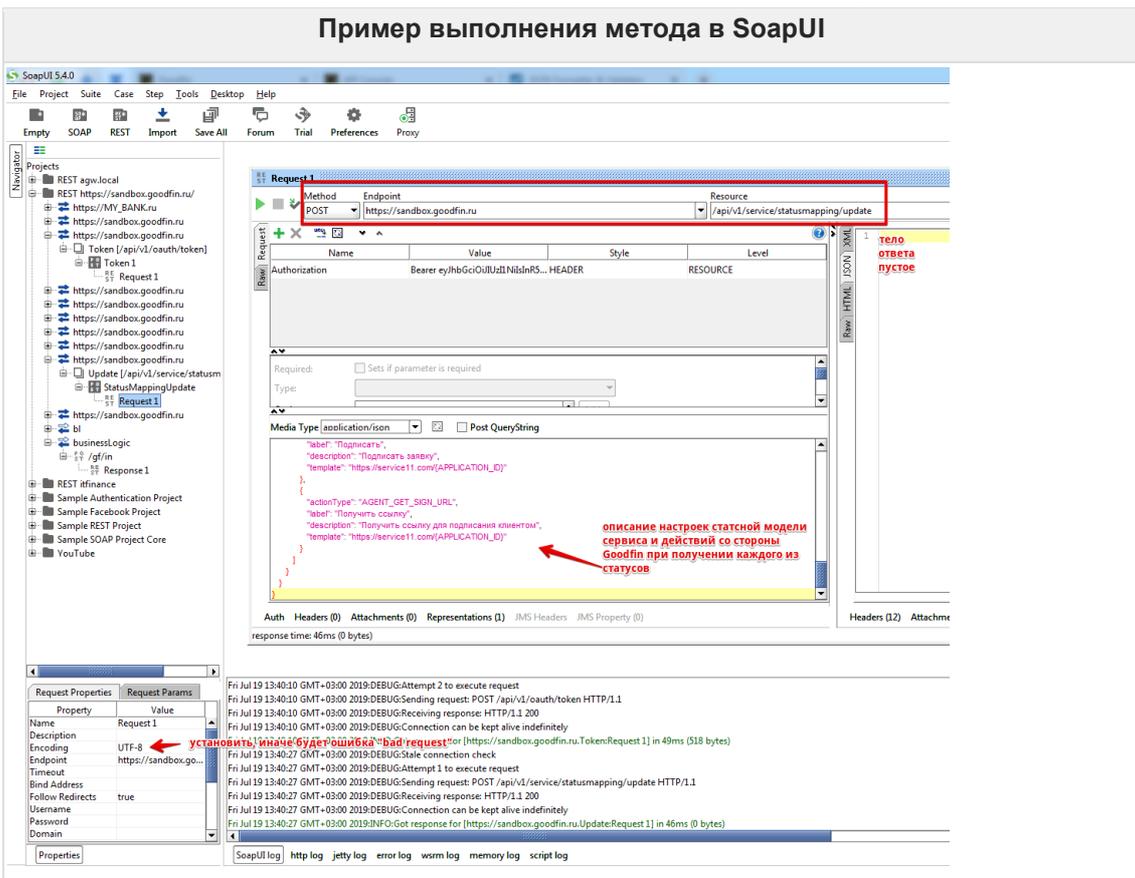
Пример получаемого ответа:

```

<Пусто>

```

Нажмите, чтобы посмотреть/скрыть пример выполнения в SoapUI



Нажмите, чтобы посмотреть/скрыть пример 2

Пример запроса для песочницы:

где Authorization - это значение параметра access_token из ответа, полученного на шаге 2.

Пример настроек через проброс кодов ошибок на статусы

```
Method: POST
URL: https://sandbox.goodfin.ru/api/v1/service/errormapping/update
HEADERS:
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOi0lsiyWRhcHRlciIsImluaXE4dW5yQjEiLCJzaGIiXSwic2NvcGUiOlsicmVhZCIiIndyaXRlIi10sImV4cCI6MTU2MDUyNTIiXNiwidXNlcjklIjoizjdjODNmZTMtOTgxYS00MTl1LTg1MjUtNjVlZWU1NTljNzYwIiwianRpIjoiodliN2U1MTgtN2M0Ny00NmFjLWJhMWMtODdkYWY3YjUzMzc3IiwiaW52xpZW50X2lkIjoiaW5pcTh1bnJCMSJ9.Zt_Idc-Ga3LDFhWdxY03toDuVJlAwoeB_u_zlWY7nNQ
BODY:
{"errorMapping":{"type":"IDENTITY"}}
```

Пример настроек через явное отображение ошибок

```
Method: POST
URL: https://sandbox.goodfin.ru/api/v1/service/errormapping/update
HEADERS:
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOi0lsiyWRhcHRlciIsImluaXE4dW5yQjEiLCJzaGIiXSwic2NvcGUiOlsicmVhZCIiIndyaXRlIi10sImV4cCI6MTU2MDUyNTIiXNiwidXNlcjklIjoizjdjODNmZTMtOTgxYS00MTl1LTg1MjUtNjVlZWU1NTljNzYwIiwianRpIjoiodliN2U1MTgtN2M0Ny00NmFjLWJhMWMtODdkYWY3YjUzMzc3IiwiaW52xpZW50X2lkIjoiaW5pcTh1bnJCMSJ9.Zt_Idc-Ga3LDFhWdxY03toDuVJlAwoeB_u_zlWY7nNQ
BODY:
{
  "errorMapping": {
    "type": "CONFIGURABLE_MAP",
    "errorCodeSysStatusMap": {
      "ERROR1": "ERR1"
    }
  }
}
```

Пример запроса для песочницы в формате HTTP/1.1:

Пример настроек через пропуск кодов ошибок на статусы

```
POST /api/v1/service/errormapping/update HTTP/1.1
Host: sandbox.goodfin.ru
Content-Type: application/json
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOi0lciYWRhcHRlciIsImluaXE4dW5yQjEiLCJzaGIiXSwic2NvcGUiOlsicmVhZCIiIndyaXRlIi0sImV4cCI6MTU2MDUyNTIwNiwiXNlclkiIjoizjdjODNmZTMtOTgxYS00MTl1LTg1MjU0MjU1NTljNzYwIiwianRpIjoiodliN2U1MTgtN2M0Ny00NmFjLWJhMWMtODdkYWZ3YjUzZmZ3IiwiaWF0IjoiY2xpZW50X2lkIjoiaW5pcThlbnJCMSJ9.Zt_Idc-Ga3LDFhWdxY03toDuVJlAwoeB_u_zlWY7nNQ
Cache-Control: no-cache

{"errorMapping":{"type":"IDENTITY"}}
```

Пример обрабатываемых по данной настройке ответов от инстанции

```
{
  "payloadType": "CommonResponse",
  "originator": "BG",
  "receiver": "Shb",
  "msgId": "40a98957-c077-4d1d-9d1f-5cc58dca1db7",
  "msgDateTime": "2019-06-27T03:33:17.429",
  "payload": {
    "originatorMsgType": "AddDealApplication",
    "originatorMsgId": "58e77408-1c33-4477-8ab9-154ee406a0cc",
    "targetObjectId": "7455db89-85a1-4fb6-afb7-20207e22c29d",
    "result": "ERROR",
    "error": {
      "id": "6236fbf9-8683-4db9-a41f-e9e38ef65d70",
      "errorCode": "ERR1",
      "message": "Test Error",
      "params": null
    }
  }
}
```

Пример настроек через явное отображение ошибок

```
POST /api/v1/service/errormapping/update HTTP/1.1
Host: sandbox.goodfin.ru
Content-Type: application/json
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOi0lciYWRhcHRlciIsImluaXE4dW5yQjEiLCJzaGIiXSwic2NvcGUiOlsicmVhZCIiIndyaXRlIi0sImV4cCI6MTU2MDUyNTIwNiwiXNlclkiIjoizjdjODNmZTMtOTgxYS00MTl1LTg1MjU0MjU1NTljNzYwIiwianRpIjoiodliN2U1MTgtN2M0Ny00NmFjLWJhMWMtODdkYWZ3YjUzZmZ3IiwiaWF0IjoiY2xpZW50X2lkIjoiaW5pcThlbnJCMSJ9.tjBOKtuRYptXV51UuBDPM7MIfeFS6Ct0uASdiFq5Mpg
Cache-Control: no-cache

{
  "errorMapping": {
    "type": "CONFIGURABLE_MAP",
    "errorCodeSysStatusMap": {
      "ERROR1": "ERR1"
    }
  }
}
```

Пример обрабатываемых по данной настройке ответов от инстанции

```
{
  "payloadType": "CommonResponse",
  "originator": "test1.service.local",
  "receiver": "Shb",
  "msgId": "f49ff225-55f3-451a-9a0a-512d9ed6b53a",
  "msgDateTime": "2019-06-27T03:51:16.008",
```

```

"payload": {
  "originatorMsgType": "AddDealApplication",
  "originatorMsgId": "a08d089e-4fd4-4800-90c0-b105f59cf61b",
  "targetObjectId": "b0187449-b76f-40b5-8618-76023c6c6980",
  "result": "ERROR",
  "error": {
    "id": "6a59c649-e7a4-4f1f-89fe-fc91f2a09ea7",
    "errorCode": "ERROR1",
    "message": "Test Error",
    "params": null
  }
}
}

```

Пример получаемого ответа:

<Пусто>

5.3.10 ШАГ 4. Обработка сервисом получаемых заявок от Goodfin (AddDealApplication)

В песочнице имеется метод `/api/v1/sandbox/generated deals`, который генерирует набор тестовых сделок/заявок по различным типам продуктов. Данный метод эмулирует отправку заявок в сервис в точку интеграции, объявленную на шаге 3.

Данный метод можно использовать для того, чтобы научиться принимать и отображать полученные заявки во внутренний формат сервиса. Описание типов данных, получаемых в составе заявок, см. в [API документации](#), варианты возможных значений справочных данных см. в [Справочники, используемые в отправляемых заявках](#).

Структура сообщения типа AddDealApplication

Примеры заявок см. в [1. Подготовительные работы по интеграции с применением API](#)

В песочнице для скачивания документов используется метод [/api/v1/document/downloaddocumentfile](https://sandbox.goodfin.ru/api/v1/document/downloaddocumentfile)

В действующей системе для скачивания документов используется метод [/api/v1/document/downloaddocumentfile](https://sandbox.goodfin.ru/api/v1/document/downloaddocumentfile)

Пример запроса для песочницы

```
Method: GET
URL: https://sandbox.goodfin.ru/api/v1/document/downloaddocumentfile
HEADERS:
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOi0lsiyWRhcHRlciIsImluaXE4dW5yQjEiLCJzaGIiXSwic2NvcGUiOlsicmVhZCIiIndyaXRlIl0sImV4cCI6MTU2MDUyNTIwNiwiXNlclklIjoizjZjdjODNmZTMtOTgxYS00MTl1LTg1MjU0NjVlZWU1NTljNzYwIiwianRpIjoiodliN2U1MTgtN2M0Ny00NmFjLWJhMWMtODdkYWY3YjUzZmZc3IiwiaW52xpZW50X2lkIjoiaW5pcTh1bnJCMSJ9.Zt_Idc-Ga3LDFhWdxY03toDuVJlAwoeB_u_zlWY7nNQ
QUERY_PARAMS:
documentFileId=0d372d43-57ee-4684-828e-8568b81d3c73
isInline=true
```

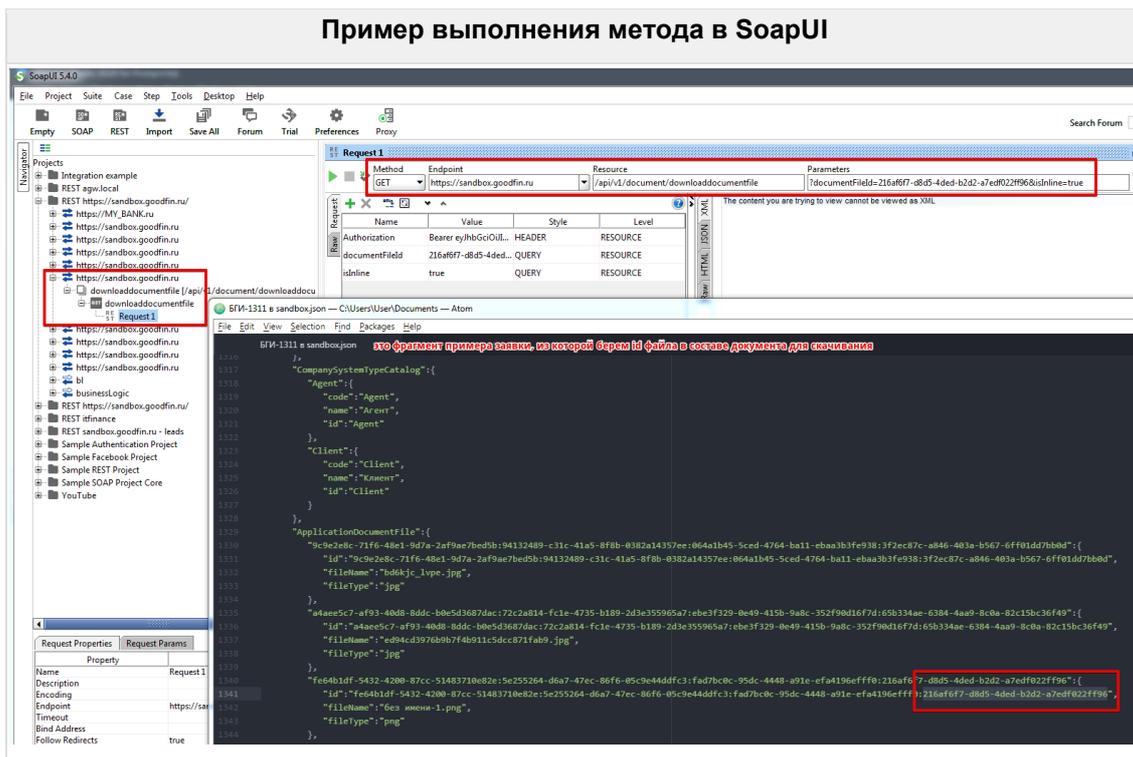
Пример запроса для песочницы в формате HTTP/1.1:

```
GET /api/v1/document/downloaddocumentfile?documentFileId=0d372d43-57ee-4684-828e-8568b81d3c73&isInline=true HTTP/1.1
Host: sandbox.goodfin.ru
Content-Type: application/json
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOi0lsiyWRhcHRlciIsImluaXE4dW5yQjEiLCJzaGIiXSwic2NvcGUiOlsicmVhZCIiIndyaXRlIl0sImV4cCI6MTU2MDUyNTIwNiwiXNlclklIjoizjZjdjODNmZTMtOTgxYS00MTl1LTg1MjU0NjVlZWU1NTljNzYwIiwianRpIjoiodliN2U1MTgtN2M0Ny00NmFjLWJhMWMtODdkYWY3YjUzZmZc3IiwiaW52xpZW50X2lkIjoiaW5pcTh1bnJCMSJ9.Zt_Idc-Ga3LDFhWdxY03toDuVJlAwoeB_u_zlWY7nNQ
```

Пример получаемого ответа:

<скачанный файл>

Нажмите, чтобы посмотреть/скрыть пример выполнения в SoapUI



5.3.12 После получения заявки и скачивания документов, банковский сервис должен оповестить систему Goodfin об успешном получении заявки или ошибке.

Чтобы Goodfin узнал о возможных кодах ошибок, которые будет слать сервис, нужно выполнить `/api/v1/service/errormapping/update` (см. описание в 3.2.)

В песочнице в точку доступа `/ad/v1/external/орепари/in` отправляется сообщение типа `CommonResponse`

В действующей системе в точку доступа `/ad/v1/external/орепари/in` отправляется сообщение типа `CommonResponse`

Нажмите, чтобы посмотреть/скрыть пример

Пример запроса для песочницы

где "receiver": "Shb" - это указание на систему Goodfin в качестве получателя сообщения;

"originator": "iniq8unrB1" - это указание на сервис - отправитель сообщения, при этом "iniq8unrB1" - логин/системное имя сервиса, которое получено на шаге 1.

```

Method: POST
URL: https://sandbox.goodfin.ru/ad/v1/external/openapi/in
HEADERS:
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOi0lsiyWRhcHRlciIsImpxMGp2TUUpYb
noiLCJzaGIiXSwic2NvcGUiOlsicmVhZCIiIndyaXRlIi0sImV4cCI6MTU2MDE1MjA2MiwidXN
lcklkIjoizjkkYmRmOTAtNzE1My00Mzk1LTg2NjEtYjZkODUyYzY4NmIwIiwianRpIjoianNTZkY
TAxNTYtODJkMy00MzZlLWJjNDMtNjFhNTkwMDFkMDY0IiwiaY2xpZW50X2lkIjoianEwanZNSlh
ueiJ9.GFN0Av8SrxfyKNiHPLJvW5NUIg59f97nbgU86Gr1nbQ
BODY:
{
  "payloadType": "CommonResponse",
  "originator": "iniq8unrB1",
  "receiver": "Shb",
  "msgId": "34433355-8018-43b4-bbbb-60f10f1db05c",
  "msgDateTime": "2019-06-18T05:37:52.812",
  "payload": {
    "originatorMsgType": "AddDealApplication",
    "originatorMsgId": "a45f8520-b132-4ab4-82ce-562affdbed43",
    "targetObjectId": "8a511353-bbbe-4ee0-8205-38816625ce07",
    "result": "SUCCESS",
    "error": null
  }
}

```

Пример запроса для песочницы в формате HTTP/1.1:

```

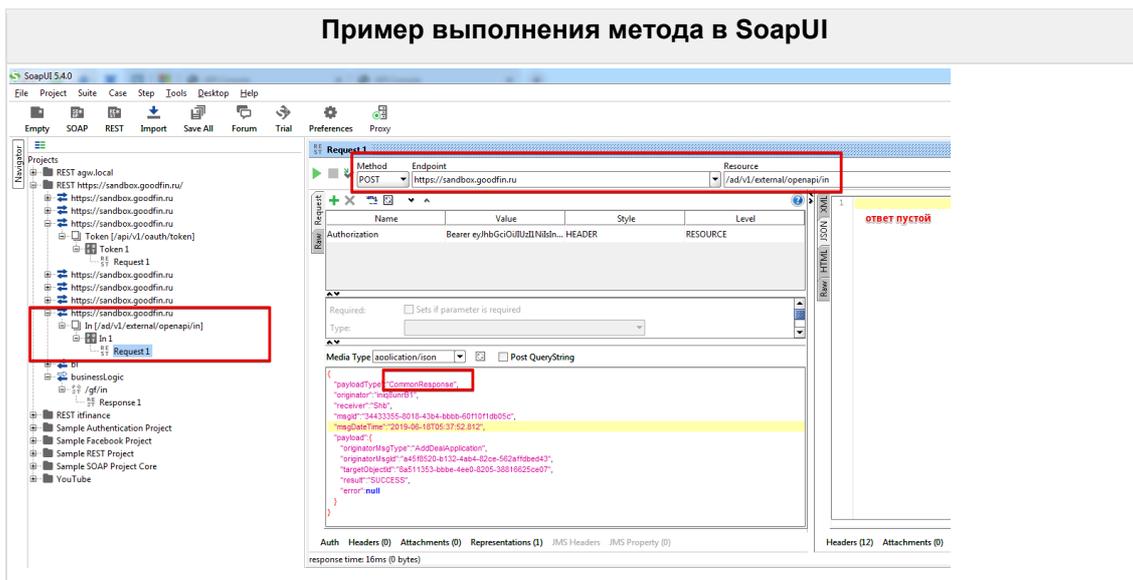
POST /ad/v1/external/openapi/in HTTP/1.1
Host: sandbox.goodfin.ru
Content-Type: application/json
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOi0lsiyWRhcHRlciIsImpxMGp2TUUpYb
noiLCJzaGIiXSwic2NvcGUiOlsicmVhZCIiIndyaXRlIi0sImV4cCI6MTU2MDE1MjA2MiwidXN
lcklkIjoizjkkYmRmOTAtNzE1My00Mzk1LTg2NjEtYjZkODUyYzY4NmIwIiwianRpIjoianNTZkY
TAxNTYtODJkMy00MzZlLWJjNDMtNjFhNTkwMDFkMDY0IiwiaY2xpZW50X2lkIjoianEwanZNSlh
ueiJ9.GFN0Av8SrxfyKNiHPLJvW5NUIg59f97nbgU86Gr1nbQ
Cache-Control: no-cache
{
  "payloadType": "CommonResponse",
  "originator": "iniq8unrB1",
  "receiver": "Shb",
  "msgId": "34433355-8018-43b4-bbbb-60f10f1db05c",
  "msgDateTime": "2019-06-18T05:37:52.812",
  "payload": {
    "originatorMsgType": "AddDealApplication",
    "originatorMsgId": "a45f8520-b132-4ab4-82ce-562affdbed43",
    "targetObjectId": "8a511353-bbbe-4ee0-8205-38816625ce07",
    "result": "SUCCESS",
    "error": null
  }
}

```

Пример получаемого ответа:

<Пусто>

Нажмите, чтобы посмотреть/скрыть пример выполнения в SoapUI



5.3.13 После получения заявки, банковский сервис может оповещать систему Goodfin об изменении статусов полученных и обрабатываемых заявок.

Чтобы Goodfin узнал о настройках статусной модели сервиса, нужно выполнить `/api/v1/service/statusmapping/update` (см. описание в 3.1.)

В песочнице в точку доступа `/ad/v1/external/openapi/in` отправляется сообщение типа `ChangeDealApplicationStatusEvent`

В действующей системе в точку доступа `/ad/v1/external/openapi/in` отправляется сообщение типа `ChangeDealApplicationStatusEvent`

Нажмите, чтобы посмотреть/скрыть пример

Пример запроса для песочницы

где "receiver": "Shb" - это указание на систему Goodfin в качестве получателя сообщения;

"originator": "iniq8unrB1" - это указание на сервис - отправитель сообщения, при этом "iniq8unrB1" - логин/системное имя сервиса, которое получено на шаге 1.

```

Method: POST
URL: https://sandbox.goodfin.ru/ad/v1/external/openapi/in
HEADERS:
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiI0IiwiaWF0IjoiYWRhcHRlciIsImpxMGp2TUUpYb
noilCJzaGIiXSwic2NvcGUiOlsicmVhZCIzIndyaXRlIl0sImV4cCI6MTU2MDE1MjA2MiwidXN
lcklkIjoiZjkkYmRmOTAtNzE1My00Mzk1LTg2NjEtYjZkODUyYzY4NmIwIiwianRpIjoianNTZkY
TAxNTYtODJkMy00MzZlLWJjNDMtNjFhNTkwMDFkMDY0IiwiaWF0IjoiY2xpZW50X2lkIjoianEwanZNSlh
ueiJ9.GFN0Av8SrxfyKNIHPLJvW5NUIg59f97nbgU86GrlnbQ
BODY:
{
    "payloadType": "ChangeDealApplicationStatusEvent",
    "originator": "iniq8unrB1",
    "receiver": "Shb",
    "msgId": "4a5f52d0-26af-4fa9-a506-40aa4aba81c5",
    "msgDateTime": "2019-06-05T11:40:59.614",
    "payload": {
        "dealApplicationId": "4a5f52d0-26af-4fa9-
a506-40aa4aba81c5",
        "status": "MY_SUPER_STATUS",
        "message": "Смена статуса",
        "params": null
    }
}

```

Пример запроса для песочницы в формате HTTP/1.1:

```

POST /ad/v1/external/openapi/in HTTP/1.1
Host: sandbox.goodfin.ru
Content-Type: application/json
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiI0IiwiaWF0IjoiYWRhcHRlciIsImpxMGp2TUUpYb
noilCJzaGIiXSwic2NvcGUiOlsicmVhZCIzIndyaXRlIl0sImV4cCI6MTU2MDE1MjA2MiwidXN
lcklkIjoiZjkkYmRmOTAtNzE1My00Mzk1LTg2NjEtYjZkODUyYzY4NmIwIiwianRpIjoianNTZkY
TAxNTYtODJkMy00MzZlLWJjNDMtNjFhNTkwMDFkMDY0IiwiaWF0IjoiY2xpZW50X2lkIjoianEwanZNSlh
ueiJ9.GFN0Av8SrxfyKNIHPLJvW5NUIg59f97nbgU86GrlnbQ
Cache-Control: no-cache

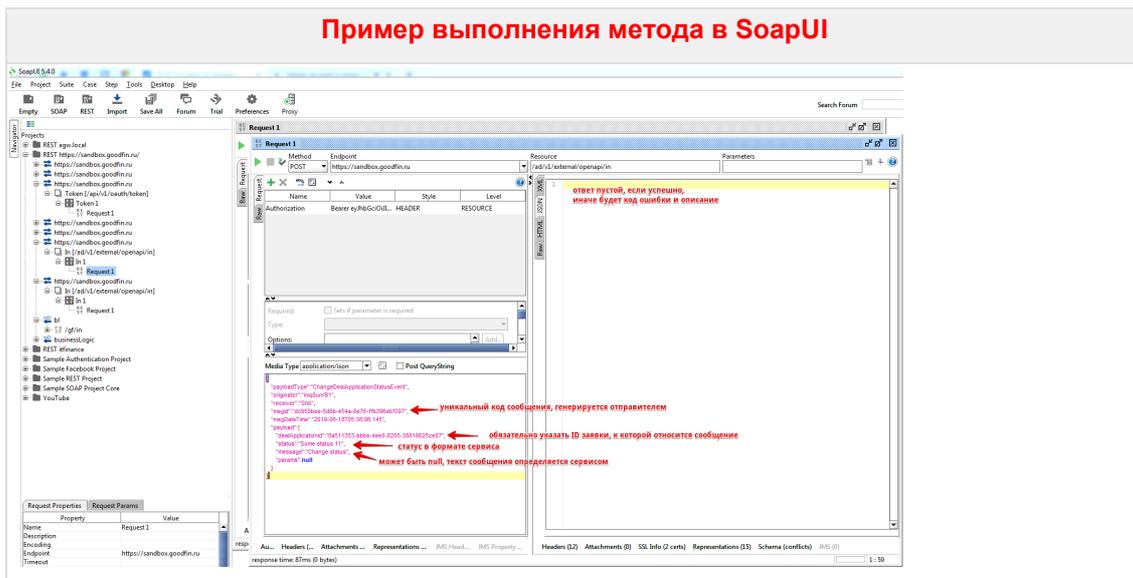
{
    "payloadType": "ChangeDealApplicationStatusEvent",
    "originator": "iniq8unrB1",
    "receiver": "Shb",
    "msgId": "4a5f52d0-26af-4fa9-a506-40aa4aba81c5",
    "msgDateTime": "2019-06-05T11:40:59.614",
    "payload": {
        "dealApplicationId": "4a5f52d0-26af-4fa9-
a506-40aa4aba81c5",
        "status": "MY_SUPER_STATUS",
        "message": "Смена статуса",
        "params": null
    }
}

```

Пример получаемого ответа:

```
<Пусто>
```

Нажмите, чтобы посмотреть/скрыть пример выполнения в SoapUI



Нажмите, чтобы посмотреть/скрыть пример 2

Например, в Goodfin пришло сообщение от сервиса "test1.srvtests.com":

```
Method: POST
URL: https://app.goodfin.ru/ad/v1/external/openapi/in
HEADERS:
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJ0b3RlciIsImpxMGp2TUplYb
noilCJzaGIiXSwic2NvcGUiOiJ0b3RlciVhZCIsIndyaXRlIj0sImV4cCI6MTU2MDE1MjA2MiwidXN
lcklIjoizjkkYmRmOTAtNzE1My00Mzk1LTg2NjEtYjZkODUyYzY4NmIwIiwianRpIjoiaWZkY
TAxNTYtODJkMy00MzZlLWJjNDMtNjFhNTkwMDFkMDY0IiwiaY2xpZW50X2lkIjoianEwanZNSlh
ueiJ9.GFN0Av8SrxfyKniHPLJvW5NUIg59f97nbgU86Gr1nbQ
BODY:
{
  "payloadType": "ChangeDealApplicationStatusEvent",
  "originator": "test1.srvtests.com",
  "receiver": "Shb",
  "msgId": "5f231fa1-fe23-40ad-a443-57e33b24504a",
  "msgDateTime": "2019-06-05T11:40:59.614",
  "payload": {
    "dealApplicationId": "bef9ebfc-6fae-4db0-9f04-bb65cbd0891c",
    "status": "16",
    "message": null,
    "params": null
  }
}
```

Этот полученный статус в составе сообщения обрабатывается согласно настроек статусов "test1.srvtests.com" на стороне Goodfin.

Пример фрагмента настройки статусной модели (вместо цифр-кодов статусов, могут быть текстовые мнемокоды):

31:

```
dealApplicationStatus: null
displayStatus: "Запрос для мониторинга"
userActions: []
```

16:

```
dealApplicationStatus: null
```

```

displayStatus: "Проект на согласовании"
userActions:
- actionType: "CLIENT_SIGN"
  label: "Принять"
  description: "Перейти по ссылке для принятия предложения"
  template: "http://test1.srvtests.com/requests/{APPLICATION_ID}/"
- actionType: "AGENT_GET_SIGN_URL"
  label: "Получить ссылку"
  description: "Получить ссылку для принятия предложения"
  template: "http://test1.srvtests.com/requests/remote-
sign/{APPLICATION_ID}/"
22:
dealApplicationStatus: "REJECTED_BY_SERVICE"
displayStatus: "Ошибка отправки"
userActions: []

```

В результате, клиент увидит у заявки статус "Проект на согласовании", а также кнопку "Принять".

А агент, если заявку готовил агент, увидит у заявки статус "Проект на согласовании", а также кнопку "Получить ссылку".

Нажмите, чтобы посмотреть/скрыть пример 3

Вы можете отправить сообщение, о смене статуса вида

```

{
  "payloadType": "ChangeDealApplicationStatusEvent",
  "originator": "test1.srvtests.com",
  "receiver": "Shb",
  "msgId": "4a5f52d0-26af-4fa9-a506-40aa4aba81c5",
  "msgDateTime": "2019-06-05T11:40:59.614",
  "payload": {
    "dealApplicationId": "4a5f52d0-26af-4fa9-a506-40aa4aba81c5",
    "status": "MY_SUPER_STATUS",
    "message": "Смена статуса",
    "params": {
      "PARAM1": "324",
      "PARAM2": "АБЫРВАЛГ"
    }
  }
}

```

В конфигурации указать, что при поступлении на Goodfin статуса MY_SUPER_STATUS - показывать пользователю кнопку "CLIENT_SIGN", с подписью "Принять", и по нажатию на которую Goodfin откроет ссылку вида http://test1.srvtests.com/path_to_action/4a5f52d0-26af-4fa9-a506-40aa4aba81c5/324?q=АБЫРВАЛГ

```

MY_SUPER_STATUS:
dealApplicationStatus: null
displayStatus: "Проект на согласовании"
userActions:
- actionType: "CLIENT_SIGN"
  label: "Принять"
  description: "Перейти по ссылке для принятия предложения"
  template:
"http://test1.srvtests.com/path\_to\_action/{APPLICATION\_ID}/{PARAM1}?q={PARAM2}"
- actionType: "AGENT_GET_SIGN_URL"
  label: "Получить
ссылку"
description: "Получить ссылку для принятия
предложения"
template:

```

```
"http://test1.srvtests.com/path to action/{APPLICATION_ID}/{PARAM1}?q={PARAM2}"
```

Мы намеренно не создавали "COMMON_LINK_ACTION" из-за сложности предсказания того, какие кейсы возможны, например, кому этот action показывать, а кому нет, какое действие должно выполняться и т.д. К тому же старались следовать логике, что системы на уровне сообщений должны минимально зависеть друг от друга. Мы исходили из того, что одна система должна только знать какой статус отправлять и какая контекстная информация может понадобиться системе-получателю о данном статусе, а другая система должна иметь возможность через конфигурацию переносить ("маппить") эти статусы на свою логику.

Вы можете прислать примеры actions, которые вы планируете показывать пользователю и в каких статусах, чтобы мы смогли сформулировать требования для расширения Goodfin.

5.3.14 После получения заявки, банковский сервис может оповестить систему Goodfin о присвоении в сервисе внутреннего номера заявке. Информация о номере заявки, присвоенном в сервисе, может помочь клиентам при обращении в службу поддержки, если возникают проблемы с заявкой у клиента.

В песочнице в точку доступа /ad/v1/external/орепари/in отправляется сообщение типа **UpdateApplicationParams**

В действующей системе в точку доступа /ad/v1/external/орепари/in отправляется сообщение типа **UpdateApplicationParams**

Нажмите, чтобы посмотреть/скрыть пример

Пример запроса для песочницы

где "receiver": "Shb" - это указание на систему Goodfin в качестве получателя сообщения;

"originator": "iniq8unrB1" - это указание на сервис - отправитель сообщения, при этом "iniq8unrB1" - логин/системное имя сервиса, которое получено на шаге 1.

```

Method: POST
URL: https://sandbox.goodfin.ru/ad/v1/external/openapi/in
HEADERS:
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOi0lsiyWRhcHRlciIsImpxMGp2TUUpYb
noilCJzaGIiXSwic2NvcGUiOlsicmVhZCIiIndyaXRlIi10sImV4cCI6MTU2MDE1MjA2MiwidXN
lcklkIjoizjkkYmRmOTAtNzE1My00Mzk1LTg2NjEtYjZkODUyYzY4NmIwIiwianRpIjoianNTZkY
TAxNTYtODJkMy00MzZlLWJjNDMtNjFhNTkwMDFkMDY0IiwiaY2xpZW50X21kIjoianEwanZNSlh
ueiJ9.GFN0Av8SrxfyKNiHPLJvW5NUIg59f97nbgU86Gr1nbQ
BODY:
{
  "payloadType": "UpdateApplicationParams",
  "originator": "iniq8unrB1",
  "receiver": "Shb",
  "msgId": "f927a962-f6c5-4682-a5ac-6c254a47b125",
  "msgDateTime": "2019-06-17T10:24:40.621",
  "payload": {
    "dealApplicationId": "0cb7e0f7-6c43-4d27-815a-111f904ea360",
    "applicationNumber": "1863"
  }
}

```

Пример получаемого ответа:

<Пусто>

5.3.15 В системе Goodfin пользователь в любой момент, пока не получен финансовый продукт по отправленной заявке, может отозвать заявку. Соответственно, в сервис в точку интеграции, объявленную на шаге 3, будет отправлено сообщение типа CancelDealApplication. В ответ сервис должен научиться переводить отозванную заявку в соответствующий статус, указывающий, что заявка больше недействительна. Далее, по результате смены статуса сервис должен отправить сообщение типа ChangeDealApplicationStatusEvent (см. шаг 4.3).

В песочнице для эмуляции отзыва заявки по инициативе системы Goodfin в точку доступа сервиса отправляется сообщение типа CancelDealApplication

В действующей системе при отзыве заявки по инициативе системы Goodfin в точку доступа сервиса отправляется сообщение типа CancelDealApplication

Нажмите, чтобы посмотреть/скрыть пример

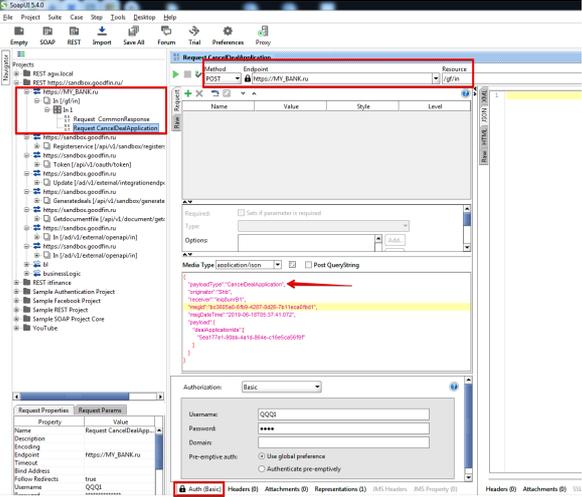
Например, в объявленную на шаге 3 точку доступа имитируем отправку сообщения от отмене заявки от Goodfin, одним сообщением может быть передан запрос на отмену сразу нескольких заявок.

```

Method: POST
URL: https://MY_BANK.ru/gf/in
HEADERS:
authentication: BASIC
userName: QQQ1
password: WWW1

BODY:
{
  "payloadType": "CancelDealApplication",
  "originator": "Shb",
  "receiver": "iniqu8unrB1",
  "msgId": "bc3605a0-6fb9-4287-8d26-7b11eca0fbd1",
  "msgDateTime": "2019-06-18T05:37:41.072",
  "payload": {
    "dealApplicationIds": [
      "5ea177e1-90bb-4a1d-864e-c16e5ca56f9f"
    ]
  }
}
    
```

Нажмите, чтобы посмотреть/скрыть пример выполнения в SoapUI



5.3.16 Goodfin шлет сообщения типа CommonResponse только с BASIC авторизацией.

Нажмите, чтобы посмотреть/скрыть пример

Например, в объявленную на шаге 3 точку доступа имитируем отправку сообщения от Goodfin.

```

Method: POST
URL: https://MY_BANK.ru/gf/in
HEADERS:
authentication: BASIC
userName: QQQ1
password: WWW1

BODY:
{
  "payloadType": "CommonResponse",
  "originator": "Shb",
  "receiver": "iniq8unrB1",
  "msgId": "77a8c8f2-5eca-4ae9-ad2d-9b5baed98593",
  "msgDateTime": "2019-06-18T05:37:52.84",
  "payload": {
    "originatorMsgType": "ChangeDealApplicationStatusEvent",
    "originatorMsgId": "dab3cdd1-3f95-41ff-8cbc-6f013630cdd0",
    "targetObjectId": "8a511353-bbbe-4ee0-8205-38816625ce07",
    "result": "SUCCESS",
    "error": null
  }
}

```

Нажмите, чтобы посмотреть/скрыть пример выполнения в SoapUI

Пример выполнения метода в SoapUI

The screenshot shows the SoapUI interface with a REST client configured for the endpoint `https://MY_BANK.ru/gf/in`. The request is a POST with Basic authentication (username: QQQ1, password: WWW1). The response is a JSON object: `{ "payloadType": "CommonResponse", "originator": "Shb", "receiver": "iniq8unrB1", "msgId": "77a8c8f2-5eca-4ae9-ad2d-9b5baed98593", "msgDateTime": "2019-06-18T05:37:52.84", "payload": { "originatorMsgType": "ChangeDealApplicationStatusEvent", "originatorMsgId": "dab3cdd1-3f95-41ff-8cbc-6f013630cdd0", "targetObjectId": "8a511353-bbbe-4ee0-8205-38816625ce07", "result": "SUCCESS", "error": null } }`. A red arrow points to the `result` field in the response body, with the text "Логин сервиса, полученный на шаге 1" (Service login, received on step 1) next to it. Another red arrow points to the `result` field in the response body, with the text "точка, объявленная на шаге 3" (point, declared on step 3) above it. The "Auth (Basic)" checkbox is checked in the Authorization section.

5.3.17 Пример реализации "заглушки" средствами SoapUI, чтобы проверить получение заявок

ПРИМЕЧАНИЕ

Данный раздел необязателен к выполнению, служит для разработчиков, желающих предварительно ознакомиться с API.
В проекте SoapUI предварительно должны быть добавлены запросы согласно шагам 1-3 из [1. Подготовительные работы по интеграции с применением API](#)

Шаг 1 - Добавляем в проект REST MockService (создаем заглушку, имитирующую точку доступа сервиса)

REST MockService

- REST <https://sandbox.goodfin.ru/>
 - https://sandbox.goodfin.ru
 - Registerservice [/api/v1/sandbox]
 - Registerservice 1
 - Request 1
 - https://sandbox.goodfin.ru
 - Token [/api/v1/oauth/token]
 - Token 1
 - Request 1
 - https://sandbox.goodfin.ru
 - Update [/ad/v1/external/integrat]
 - Update 1
 - Request 1
 - https://sandbox.goodfin.ru
 - Generatedeals [/api/v1/sandbox]
 - Generatedeals 1
 - Request 1

New MockService

Specify name of MockService

businessLogic

OK Cancel

пример наименования заглушки, которая имитирует точку доступа сервиса

Шаг 2 - Готовимся настроить заглушку

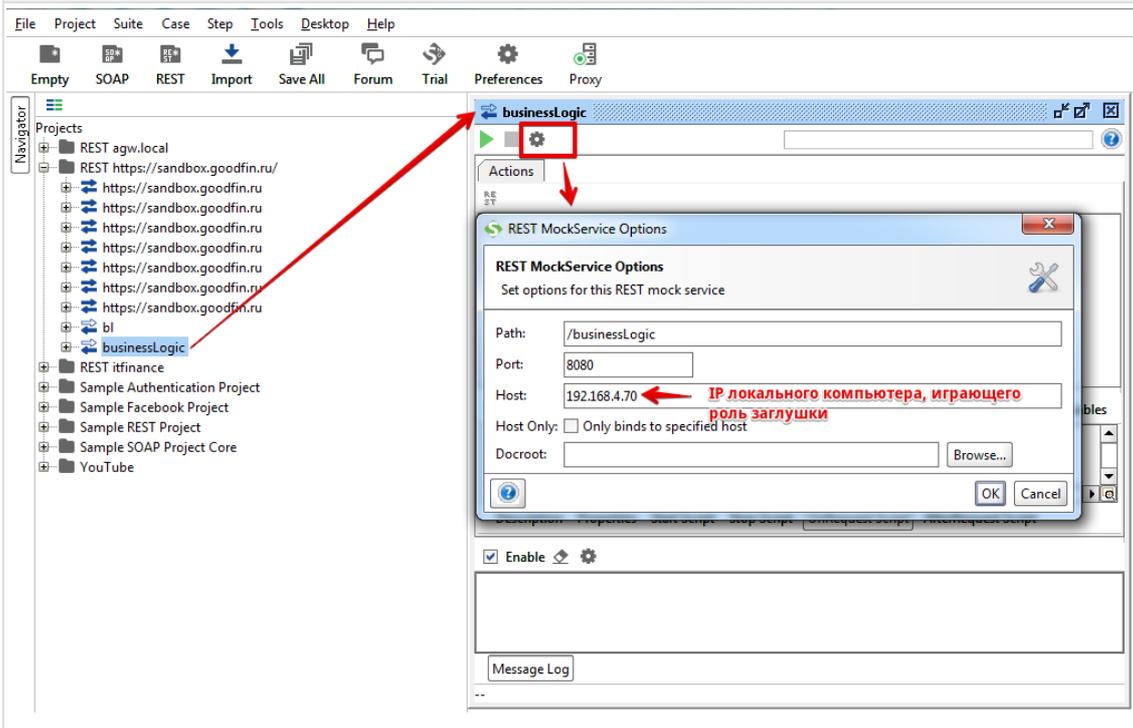
The screenshot displays a REST client interface with a project tree on the left. The tree is expanded to show a folder named 'REST https://sandbox.goodfin.ru/'. Inside this folder, several endpoints are listed, each with a corresponding mock action:

- Endpoint: `https://sandbox.goodfin.ru` (with a double-headed arrow icon)
 - Mock Action: Registerservice [/api/v1/sandbox] (with a document icon)
 - Mock Action: Registerservice 1 (with a REST icon)
 - Mock Action: Request 1 (with a REST icon)
 - Endpoint: `https://sandbox.goodfin.ru` (with a double-headed arrow icon)
 - Mock Action: Token [/api/v1/oauth/token] (with a document icon)
 - Mock Action: Token 1 (with a REST icon)
 - Mock Action: Request 1 (with a REST icon)
 - Endpoint: `https://sandbox.goodfin.ru` (with a double-headed arrow icon)
 - Mock Action: Update [/ad/v1/external/integrat] (with a document icon)
 - Mock Action: Update 1 (with a REST icon)
 - Mock Action: Request 1 (with a REST icon)
 - Endpoint: `https://sandbox.goodfin.ru` (with a double-headed arrow icon)
 - Mock Action: Generatedeals [/api/v1/sandbox] (with a document icon)
 - Mock Action: Generatedeals 1 (with a REST icon)
 - Mock Action: Request 1 (with a REST icon) - Endpoint: `bl` (with a double-headed arrow icon)
 - Mock Action: businessLogic (with a double-headed arrow icon)

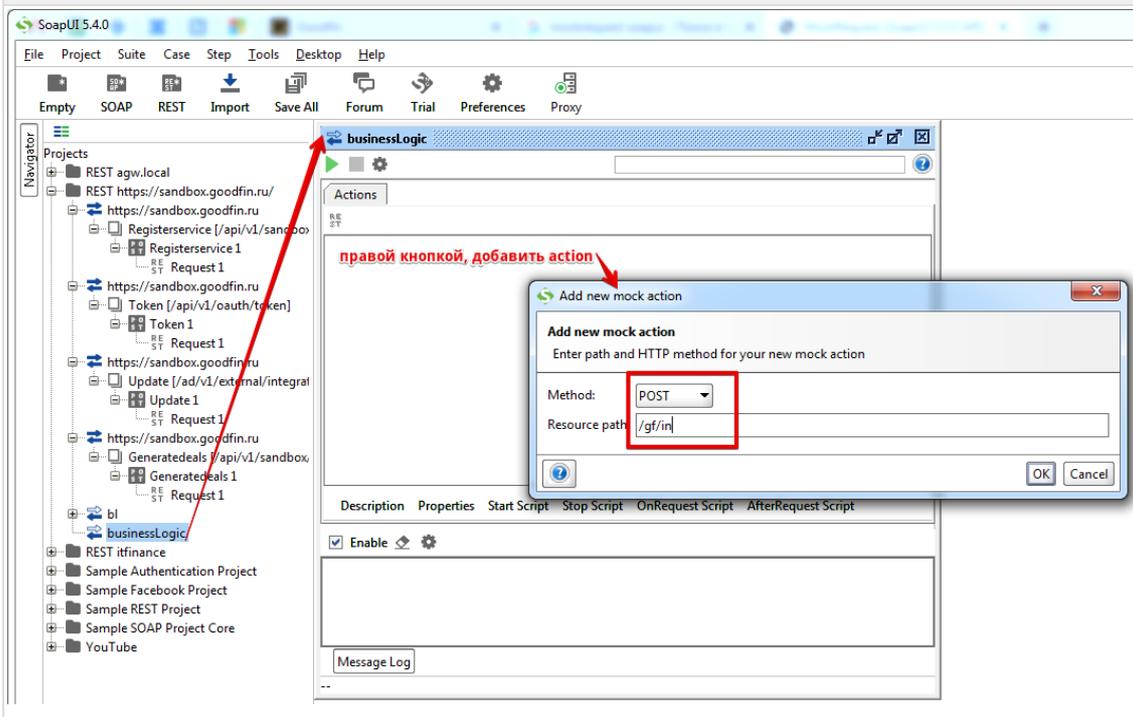
The 'businessLogic' folder is selected, and a context menu is open over it. The menu items are:

 - Show REST MockService Editor (with 'Enter' shortcut)
 - Start Minimized
 - Stop
 - Restart
 - Add new mock action
 - Rename (with 'F2' shortcut)
 - Remove (with 'Delete' shortcut)

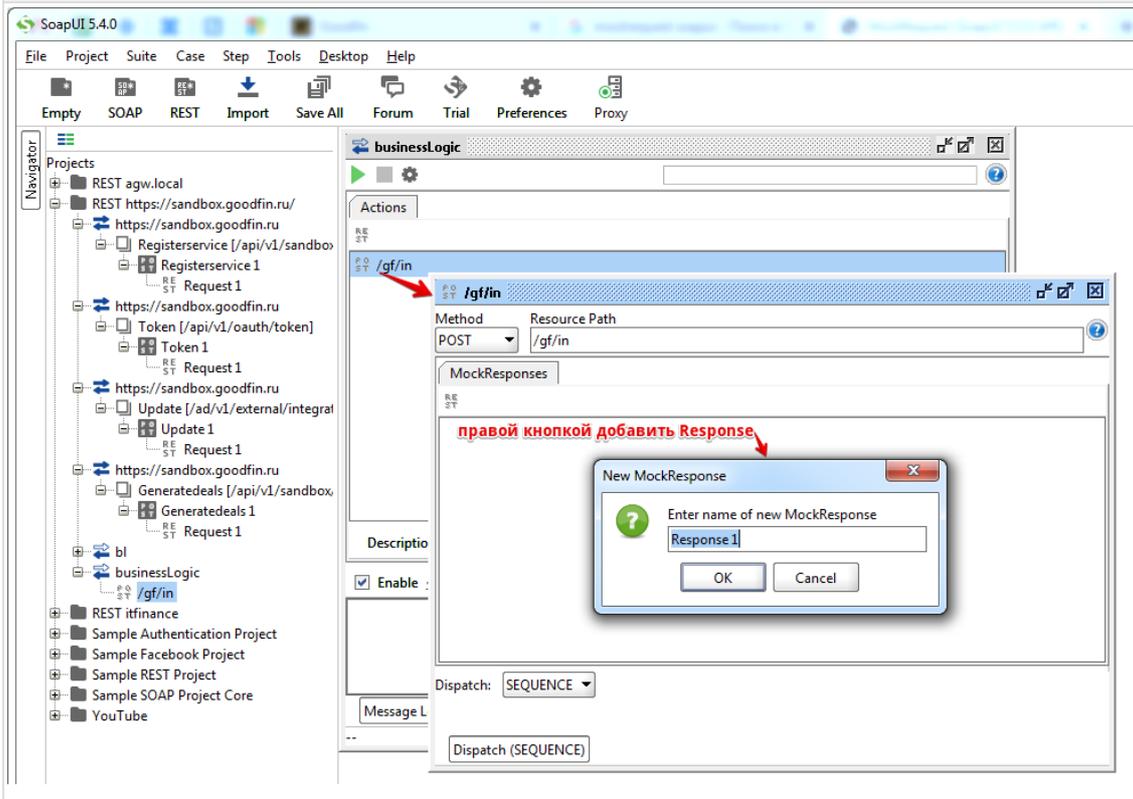
Шаг 3 - Настраиваем адрес для заглушки, используя IP локального компьютера



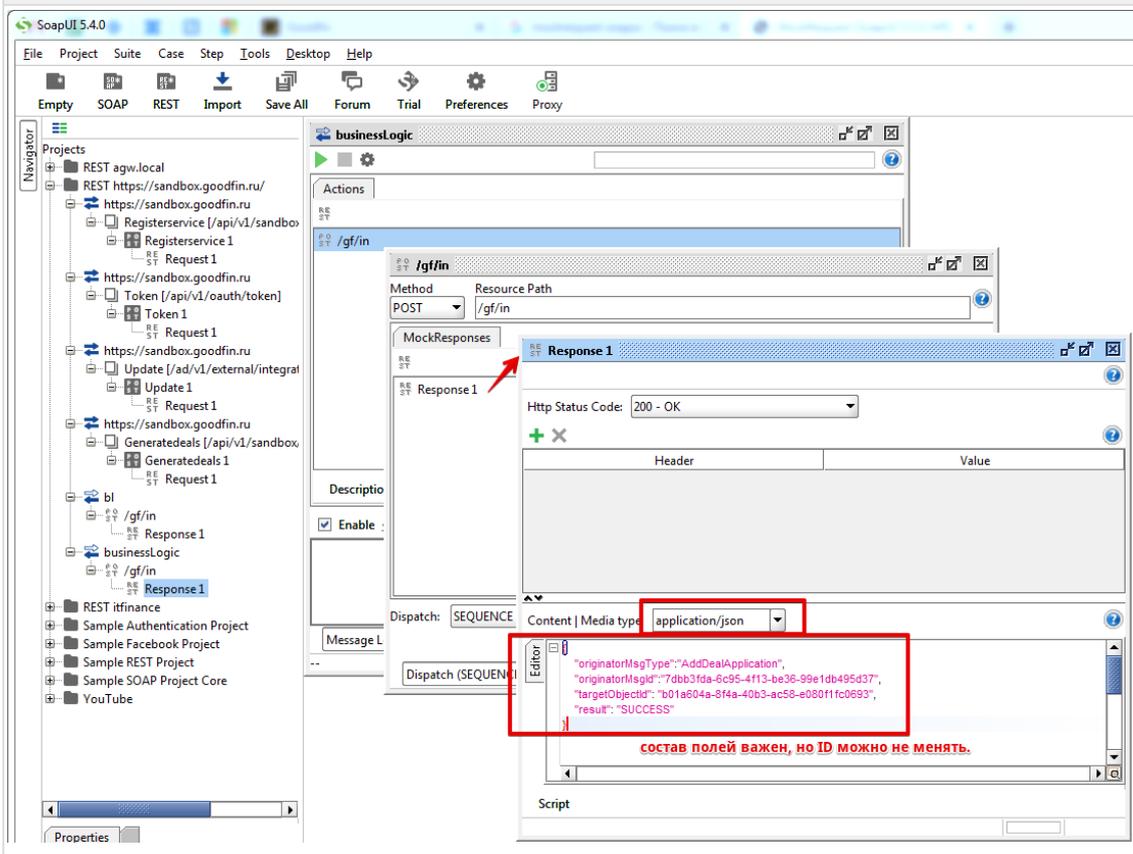
Шаг 4 - Добавляем действие-заглушку



Шаг 5 - Добавляем типовой Response для заглушки



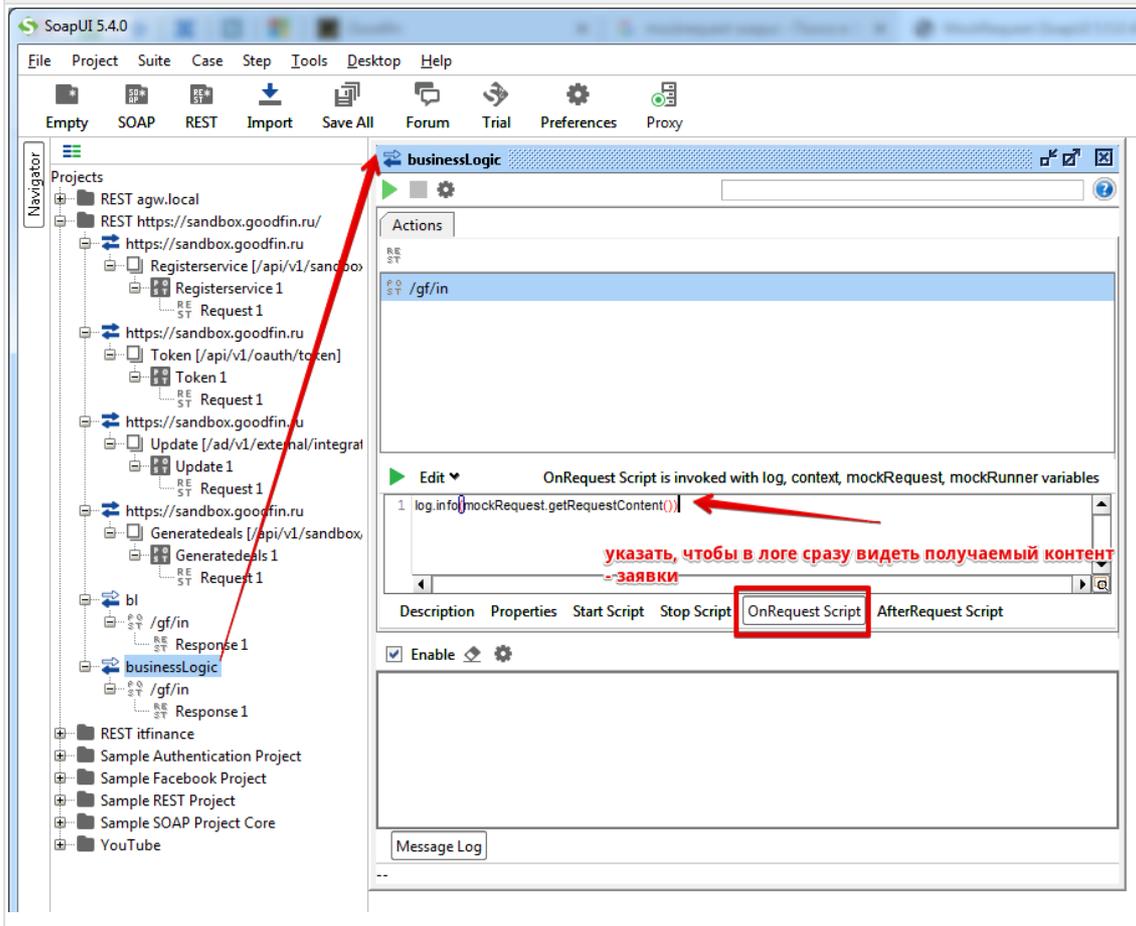
Шаг 6 - Добавляем тело Response для заглушки



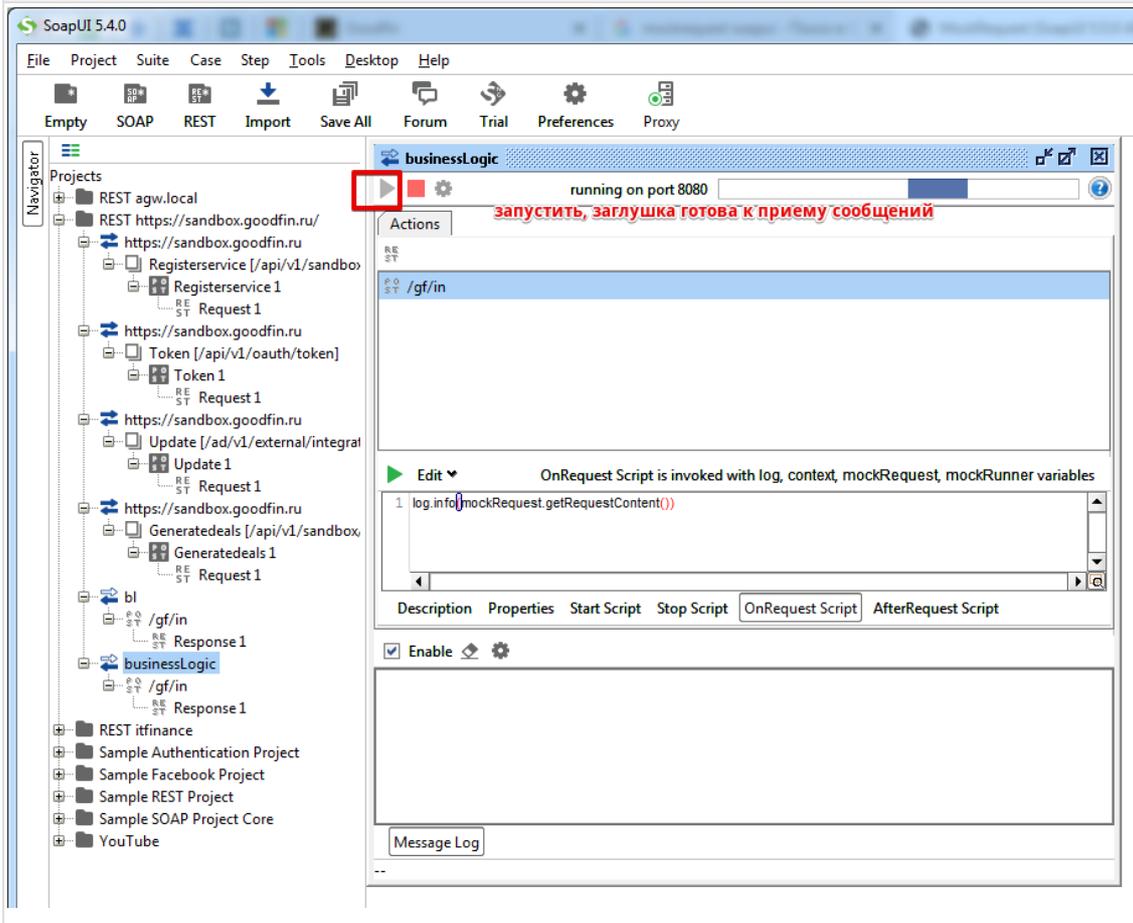
Нажмите, чтобы открыть/скрыть пример тела Response

```
{  
  "originatorMsgType": "AddDealApplication",  
  "originatorMsgId": "7dbb3fda-6c95-4f13-be36-99e1db495d37",  
  "targetObjectId": "b01a604a-8f4a-40b3-ac58-e080f1fc0693",  
  "result": "SUCCESS"  
}
```

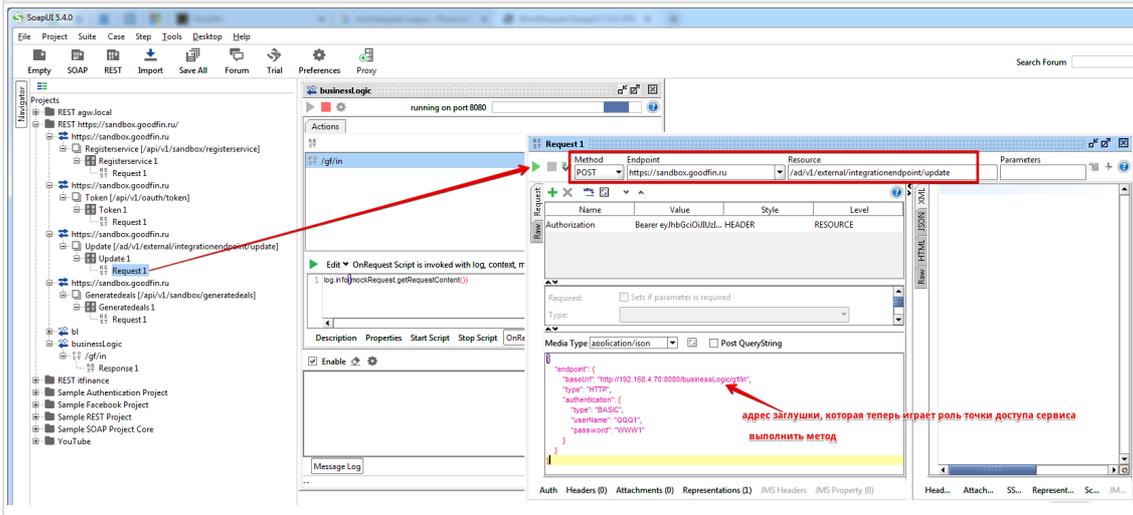
Шаг 7 - Устанавливаем режим вывода лога, чтобы видеть весь контент, приходящий на заглушку



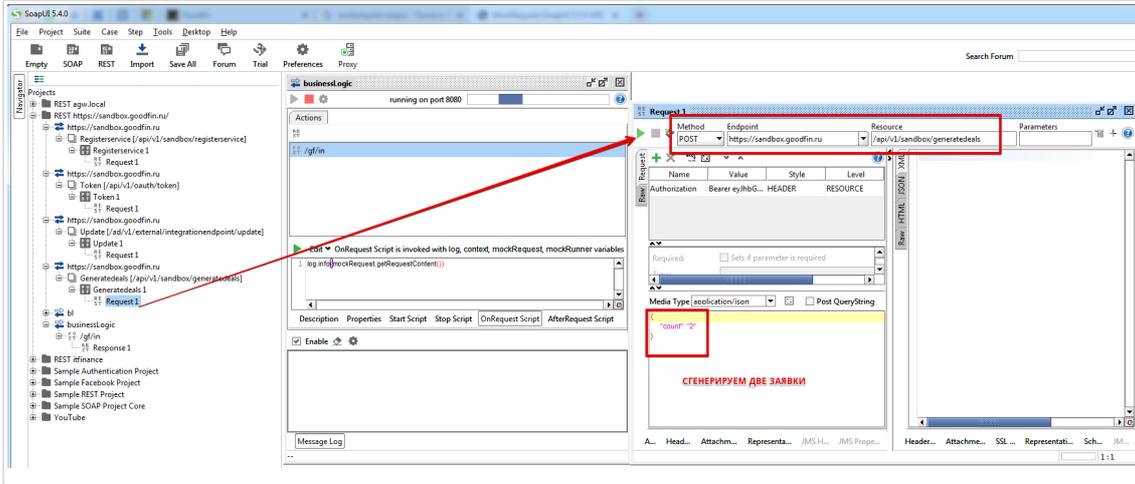
Шаг 8 - Запускаем заглушку, имитирующую точку доступа банковского сервиса



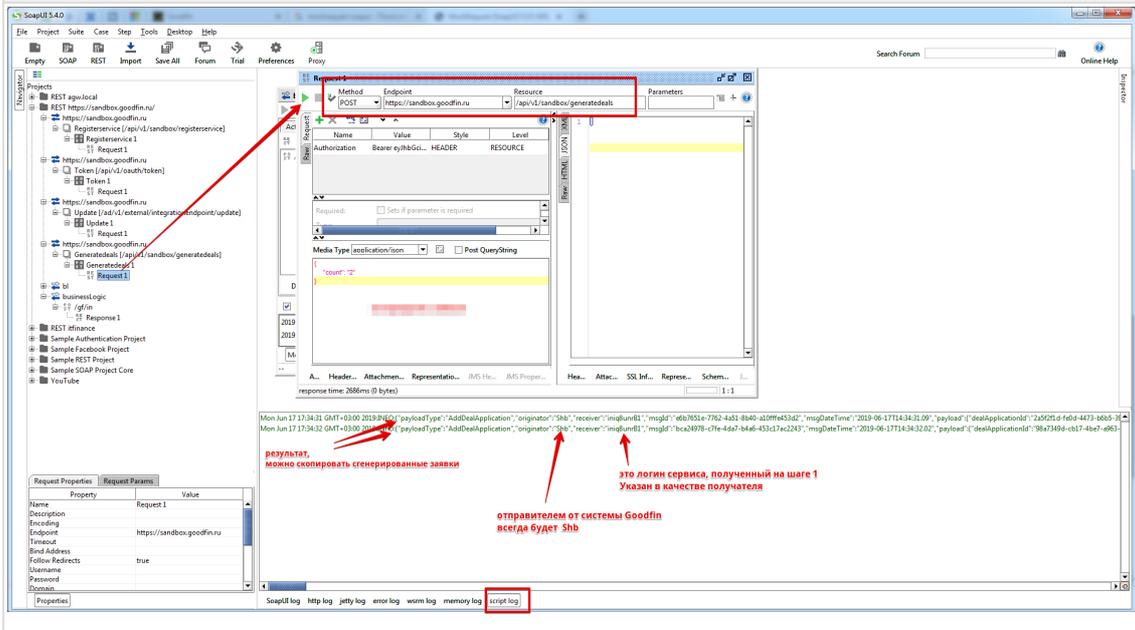
Шаг 9 - Сообщаем новые данные о местоположении точки доступа "песочнице" (тестовому окружению Goodfin)



Шаг 10 - Вызываем метод генерации заявок, имитирующий процесс отправки заявок в сервис (в данном случае, на заглушку)



Шаг 11 - Видим результат



5.4 II.1. Создание продукта в виде конфигурационного файла уaml

Требования и особенности формирования конфигурационного файла описаны в виде комментариев в примере ниже.

ВНИМАНИЕ!

Оформлять отдельными продуктами нужно отличающиеся наборы обязательных для заполнения данных. Если продукт предоставляется для ЮЛ и ИП, то требуется оформлять два продукта: "Продукт... для ЮЛ", "Продукт... для ИП", т.к. у них отличаются наборы обязательных для заполнения полей (например, у ИП нет поля КПП, нельзя это поле указать для ИП как обязательное).

ПРИМЕЧАНИЕ

Рекомендуется перед импортом проверить файл с помощью любого валидатора yaml, чтобы исключить явные ошибки форматирования.

Скачать пример можно также по ссылке [Пример для документации - Продукт БГ\(И\) для ЮЛ ред.12.04.2021.yaml](#)

```

#настройки конкретного продукта
#элемент product обязателен и один
product:
  #общие сведения о продукте
  #полное наименование продукта, необязательно должно совпадать с
наименованиями линейки продуктов банка, т.к. в системе Goodfin продукт
объединяет единые настройки обязательности по заполнению данных заявки
  #name varchar(500) NOT NULL
  name: "Банковская гарантия на исполнение для ЮЛ банка БАНК1
(ред.12.04.21)"
  #shortname varchar(50) NOT NULL
  #краткое наименование продукта, выводимое пользователям
shortName: "БГ на исполнение для ЮЛ (ред.12.04.21, БАНК1)"
  #producttype varchar(50) NOT NULL
  #мнемокод типа продукта, выбирается из справочника product_types
  #тип продукта должен совпадать с префиксам в перечне обязательных
продуктовых полей
  #например, "BG" - Банковская гарантия на исполнение контракта,
"PARTICIPATION_BG" - Банковская гарантия на участие, "TENDER_LOAN" -
Тендерный займ
  productType: "BG"
  #description varchar(1000) NULL
  #Краткое описание паспорта продукта для вывода пользователям, чтобы
пользователи могли понять ограничения по продукту
  #можно использовать при описании формат BVCODE
  description: "[b]Законы:[/b] 44, 223, 615-ПП/185.[br][b]Макс. сумма
БГ:[/b] 150 млн.р.[br][b]Макс. срок действия БГ:[/b] 1095 дн.[br][b]Срок
существования клиента:[/b] от 3 мес.-до 500 т.р.; от 6 мес.-до 1 м.р.; от
9 мес.-от 1 м.р. до 7 м.р.; от 12 мес.-от 7 м.р. до 50 м.р.; от 24 мес.-
от 50 м.р. до 150 м.р.[br][b]Макс. лимит на клиента:[/b] 150
м.р.[br][b]ОПФ клиента:[/b] АО, ООО, ИП.[br][b]Стоп регионы (по
принципалу, бенефициару и месту выполнения контракта) :[/b] Респ. Адыгея;
Карачаево-Черкесская Респ.; Кабардино-Балкарская Респ.; Респ. Северная
Осетия - Алания; Респ. Ингушетия; Чеченская Респ.; Респ. Дагестан; Респ.
Крым, г. Севастополь (допускается как место выполнения
работ).[br][b]Предметы в стопе:[/b] автодорожное строительство «с нуля»;
строительство жилых домов, высотных домов, коммерческой недвижимости,
производственных объектов; приобретение жилья в рамках 214 ФЗ от
31.12.2004; строительство объектов для нужд МВД РФ, ФСБ РФ, Министерства
обороны РФ, Администрации Президента РФ.[br]"
  # workterm integer NULL
  #Минимальное количество рабочих дней, в течение которых сервис готов
предоставить продукт или услугу в законченном виде, измеряется в рабочих
днях
  workTerm: 1
  #externalid uuid NOT NULL
  #задается согласно требованиям к guid, например, использовать
https://www.guidgenerator.com для генерации значения
  #если externalId совпадет с существующим продуктом в БД, то будет
выполнена попытка перезаписи настроек продукта, иначе создается новый
продукт
  externalId: "12f6fa12-3f16-4ab6-94b6-466c3ce3a09e"
  #правило применимости продукта, данные клиента и сделки должны
удовлетворять этому правилу, чтобы получить предложение по продукту
  #правило обязательно и одно на продукт
  productApplicabilityRule:
    #name varchar(150) NOT NULL
    #используется для понимания сути правила
    name: "БГ на исполнение ЮЛ банка БАНК1. Доступно: по ФЗ 223, 44,
185/615 ПП; для коммер. и некомм. головных компаний"
    #type varchar(50) NOT NULL, по умолчанию значение = "script"
    type: "script"
    #script character varying(4000) NOT NULL
    #Скрипт правила в виде функции
isAvailableForDealApplicationData(dealApplicationData), возвращающей TRUE,
FALSE или NULL.

```

```

#Если функция возвращает true, то считается, что продукт применим и
может быть показан клиенту в Предложениях
#примеры готовых скриптов см. в документации - статья "Правила
применимости: productApplicabilityRule, applicabilityRule"
script: "function
isAvailableForDealApplicationData(dealApplicationData) {
    var orgTypes = ['COM', 'UNCOM'];
    var companyClass =
java.lang.Class.forName('com.keyintegrity.shb.company.query.dto.CompanyDto
');
    var clientCompany =
dealApplicationData.fetchFields[companyClass][dealApplicationData.result.c
lient.id];
    if(clientCompany == null || clientCompany.orgType
== null || clientCompany.legalType == null) {
        return null;
    }
    var clientOrgTypeId = clientCompany.orgType.id;
    var clientLegalTypeId =
clientCompany.legalType.id;
    return orgTypes.indexOf(clientOrgTypeId) !== -1
&& clientLegalTypeId == 'ORGANIZATION';
}"

#-----
-----
#требования к карточке компании/ИП
#данный раздел должен существовать в настройках, если тип продукта
предоставляется компаниям/ИП, иначе будет раздел individualFields
#при анализе yaml считать, если поле в списке required не найдено, то
оно необязательно
#список доступных полей для настройки обязательности см. в справочнике
company_fields
companyFields:
  required:
  - "bankingDetails"
  - "bankingDetails.bankName"
  - "bankingDetails.bik"
  - "bankingDetails.paymentAccount"
  - "baseOkved"
  - "factAddress"
  - "factAddress.addressType"
  - "factAddress.city"
  - "factAddress.house"
  - "factAddress.isOwned"
  - "factAddress.postCode"
  - "factAddress.region"
  - "factAddress.street"
  - "fullName"
  - "inn"
  - "kpp"
  - "legalAddress"
  - "legalAddress.addressType"
  - "legalAddress.city"
  - "legalAddress.house"
  - "legalAddress.isOwned"
  - "legalAddress.postCode"
  - "legalAddress.region"
  - "legalAddress.street"
  - "legalFounders.address"
  - "legalFounders.inn"
  - "legalFounders.kpp"
  - "legalFounders.name"
  - "legalFounders.ogrn"
  - "legalFounders.percent"

```

```

- "licenses.activity"
- "licenses.date"
- "licenses.listActivities"
- "licenses.number"
- "licenses.who"
- "ogrn"
- "okopf"
- "okpo"
- "orgType"
- "persons"
- "persons.birthday"
- "persons.birthdayPlace"
- "persons.docDate"
- "persons.docNumber"
- "persons.docType"
- "persons.docWho"
- "persons.firstName"
- "persons.inn"
- "persons.lastName"
- "persons.mainOksm"
- "persons.middleName"
- "persons.regAddress"
- "regDate"
- "regOrg"
- "regPlace"
- "shortName"
- "taxSystem"
#-----
-----
#требования к продуктовым полям
#набор продуктовых полей будет отличаться в зависимости от типа продукта
#при анализе yaml считать, если поле в списке required не найдено, то
оно необязательно
#при описании продуктовых полей важно, чтобы префикс типа продукта
совпадал с типом продукта в общем описании продукта (с полем productType)
#можно использовать только те продуктовые поля, которые соответствуют
настраиваемому типу продукта
#полный перечень см. в справочнике product_types_fields
productFields:
  required:
    - "BG:productDealState.actualContractPrice"
    - "BG:productDealState.amount"
    - "BG:productDealState.concludedContracts"
    - "BG:productDealState.customer"
    - "BG:productDealState.customer.inn"
    - "BG:productDealState.customer.kpp"
    - "BG:productDealState.customer.name"
    - "BG:productDealState.customer.ogrn"
    - "BG:productDealState.customer.region"
    - "BG:productDealState.deadlineForGrant"
    - "BG:productDealState.endDate"
    - "BG:productDealState.hasLoss"
    - "BG:productDealState.isIndisputableWriteOff"
    - "BG:productDealState.isPrepayment"
    - "BG:productDealState.isQuickDecision"
    - "BG:productDealState.isWarrantyPeriod"
    - "BG:productDealState.loanCommitments"
    - "BG:productDealState.startDate"
    - "BG:productDealState.tender"
    - "BG:productDealState.tender.federalLaw"
    - "BG:productDealState.tender.initialContractPrice"
    - "BG:productDealState.tender.isMultiCustomer"
    - "BG:productDealState.tender.isMultiLot"
    - "BG:productDealState.tender.lotName"
    - "BG:productDealState.tender.lotNumber"

```

```

- "BG:productDealState.tender.name"
- "BG:productDealState.tender.placementType"
- "BG:productDealState.tender.publishDateTime"
- "BG:productDealState.tender.registryNumber"
- "BG:productDealState.tender.url"
#-----
-----
#валидаторы по табл. products_validators и validators
validators:
#обязательно в каждом блоке массива элемент type с наименованием
функции валидатора
#прочие элементы в блоке массива после type играют роль входных
параметров для функции валидатора, параметров может не быть, а может быть
несколько с разными наименованиями
#полный перечень см. в справочнике validators
- #validator
type: "COMPANY_PERSON_ROLE_REQUIRED"
role: "BOOKER"
- #validator
type: "COMPANY_PERSON_ROLE_REQUIRED"
role: "DIRECTOR"
- #validator
type: "COMPANY_FOUNDER_REQUIRED"
#-----
-----
#пакеты обязательных документов, можно настроить несколько пакетов
документов в зависимости от данных в заявке и требований продукта
#в списке столько Блоков "- docsPackage:" сколько пакетов для продукта
#если в пакете документов отсутствует блок applicabilityRule, то это
значит, что документы из пакета будут безусловно запрашиваться для
продукта
docsPackages:
- #docsPackage 1
#name varchar(500) NOT NULL
#Указать полное наименование пакета документов, включая наименование
продукта и сервиса для полноты понимания
name: "Пакет БГ(И), Банк БАНК1: ВСЕ клиенты, 1 млн < сумма БГ на
исполнение < = 50 млн"
#shortname varchar(50) NOT NULL
#Указать краткое наименование пакета документов, чтобы по нему понять
условия применимости пакета документов
shortName: "ЮЛ, 1 млн < сумма < = 50 млн"
#description varchar(1000) NULL, описание пакета документов, которое
может выводиться как подсказка в системе
description: NULL
#полный перечень типов документов см. в docs_types
#списком перечисляются мнемокоды типов обязательных документов в
пакете
#полный перечень см. в справочнике docs_types
docsTypes:
- "ANALYTICAL_BALANCE_LAST_REPORTING_YEAR"
- "LEASE_AGREEMENT_OWNERSHIP"
- "FINANCE_REPORT_LAST_QUARTER"
#правило применимости пакета документов
applicabilityRule:
#name varchar(150) NOT NULL
#наименование правила применимости
name: "сумма БГ на исполнение больше 1 млн И меньше или равна 50
млн"
#type varchar(50) NOT NULL, по умолчанию значение = "script"
type: "script"
#script character varying(4000) NOT NULL

```

```

#Скрипт правила в виде функции
isAvailableForDealApplicationData(dealApplicationData), возвращающей TRUE,
FALSE или NULL.
#Если функция возвращает true, то считается, что пакет документов
применим и при заполнении заявки у клиента будут запрошены документы в
составе пакета
#примеры готовых скриптов см. в документации - статья "Правила
применимости: productApplicabilityRule, applicabilityRule"
script: "function
isAvailableForDealApplicationData(dealApplicationData) {
    if (dealApplicationData.result.productDealState.amount
== null) {
        return null;
    }
    return
dealApplicationData.result.productDealState.amount > 1000000 &&
dealApplicationData.result.productDealState.amount <= 50000000;
}"
- #docsPackage 2
#name varchar(500) NOT NULL
#Указать полное наименование пакета документов, включая наименование
продукта и сервиса для полноты понимания
name: "Пакет БГ(И), Банк БАНК1: ВСЕ клиенты, сумма БГ на исполнение >
50 млн"
#shortname varchar(50) NOT NULL
#Указать краткое наименование пакета документов, чтобы по нему понять
условия применимости пакета документов
shortName: "ЮЛ, сумма > 50 млн"
#description varchar(1000) NULL, описание пакета документов, которое
может выводиться как подсказка в системе
description: NULL
#полный перечень типов документов см. в docs_types
#списком перечисляются мнемокоды типов обязательных документов в
пакете
#полный перечень см. в справочнике docs_types
docsTypes:
- "FINANCE_REPORT_LAST_QUARTER"
- "ANALYTICAL_BALANCE_LAST_REPORTING_YEAR"
- "ANNUAL_ACCOUNTING_FINANCIAL_STATEMENTS"
- "LEASE_AGREEMENT_OWNERSHIP"
#правило применимости пакета документов
applicabilityRule:
#name varchar(150) NOT NULL
#наименование правила применимости
name: "сумма БГ на исполнение больше 50 млн"
#type varchar(50) NOT NULL, по умолчанию значение = "script"
type: "script"
#script character varying(4000) NOT NULL
#Скрипт правила в виде функции
isAvailableForDealApplicationData(dealApplicationData), возвращающей TRUE,
FALSE или NULL.
#Если функция возвращает true, то считается, что пакет документов
применим и при заполнении заявки у клиента будут запрошены документы в
составе пакета
#примеры готовых скриптов см. в документации - статья "Правила
применимости: productApplicabilityRule, applicabilityRule"
script: "function
isAvailableForDealApplicationData(dealApplicationData) {
    if (dealApplicationData.result.productDealState.amount
== null) {
        return null;
    }
    return
dealApplicationData.result.productDealState.amount > 50000000;
}"
- #docsPackage 3

```

```

#name varchar(500) NOT NULL
#Указать полное наименование пакета документов, включая наименование
продукта и сервиса для полноты понимания
name: "Пакет БГ(И), Банк БАНК1: ЮЛ на ОСНО, сумма БГ на исполнение >
10 млн"
#shortname varchar(50) NOT NULL
#Указать краткое наименование пакета документов, чтобы по нему понять
условия применимости пакета документов
shortName: "ЮЛ на ОСНО, сумма > 10 млн"
#description varchar(1000) NULL, описание пакета документов, которое
может выводиться как подсказка в системе
description: NULL
#полный перечень типов документов см. в docs_types
#списком перечисляются мнемокоды типов обязательных документов в
пакете
#полный перечень см. в справочнике docs_types
docsTypes:
- "TAX_DECL_VALUE_ADDED_TAX_LAST_TAX_PERIOD"
- "TAX_DECL_ORGANIZATION_INCOME_TAX_LAST_TAX_PERIOD"
#правило применимости пакета документов
applicabilityRule:
#name varchar(150) NOT NULL
#наименование правила применимости
name: "ЮЛ с налогообложением = ОСНО И сумма БГ на исполнение больше
10 млн"
#type varchar(50) NOT NULL, по умолчанию значение = "script"
type: "script"
#script character varying(4000) NOT NULL
#Скрипт правила в виде функции
isAvailableForDealApplicationData(dealApplicationData), возвращающей TRUE,
FALSE или NULL.
#Если функция возвращает true, то считается, что пакет документов
применим и при заполнении заявки у клиента будут запрошены документы в
составе пакета
#примеры готовых скриптов см. в документации - статья "Правила
применимости: productApplicabilityRule, applicabilityRule"
script: "function
isAvailableForDealApplicationData(dealApplicationData) {
    var companyClass =
java.lang.Class.forName('com.keyintegrity.shb.company.query.dto.CompanyDto
');
    var clientCompany =
dealApplicationData.fetchFields[companyClass][dealApplicationData.result.c
lient.id];
    if (clientCompany == null) {
        return null;
    }
    var taxSystemClass =
java.lang.Class.forName('com.keyintegrity.shb.company.query.dto.catalog.Ta
xSystemCatalogDto');
    var taxSystemDto =
dealApplicationData.fetchFields[taxSystemClass][clientCompany.taxSystem];
    if (clientCompany.legalType == null || taxSystemDto ==
null || dealApplicationData.result.productDealState.amount == null) {
        return null;
    }
    var taxSystems = ['OSNO'];
    return taxSystems.indexOf(taxSystemDto.code) !== -1 &&
clientCompany.legalType.id == 'ORGANIZATION' &&
dealApplicationData.result.productDealState.amount > 10000000;
}"
- #docsPackage 4
#name varchar(500) NOT NULL
#Указать полное наименование пакета документов, включая наименование
продукта и сервиса для полноты понимания

```

```

    name: "Пакет БГ(И), Банк БАНК1: ЮЛ на УСН/Сельхоз, 1 млн < сумма БГ на
исполнение < = 10 млн"
    #shortname varchar(50) NOT NULL
    #Указать краткое наименование пакета документов, чтобы по нему понять
условия применимости пакета документов
    shortName: "ЮЛ на УСН/Сельхоз, 1млн < сумма <= 10млн"
    #description varchar(1000) NULL, описание пакета документов, которое
может выводиться как подсказка в системе
    description: NULL
    #полный перечень типов документов см. в docs_types
    #списком перечисляются мнемокоды типов обязательных документов в
пакете
    #полный перечень см. в справочнике docs_types
    docsTypes:
    - "TAX_DECLARATION_LAST_PERIOD"
    #правило применимости пакета документов
    applicabilityRule:
    #name varchar(150) NOT NULL
    #наименование правила применимости
    name: "ЮЛ с налогообложением = (УСНО ИЛИ Сельхоз) И сумма БГ на
исполнение больше 1 млн И меньше или равна 10 млн"
    #type varchar(50) NOT NULL, по умолчанию значение = "script"
    type: "script"
    #script character varying(4000) NOT NULL
    #Скрипт правила в виде функции
isAvailableForDealApplicationData(dealApplicationData), возвращающей TRUE,
FALSE или NULL.
    #Если функция возвращает true, то считается, что пакет документов
применим и при заполнении заявки у клиента будут запрошены документы в
составе пакета
    #примеры готовых скриптов см. в документации - статья "Правила
применимости: productApplicabilityRule, applicabilityRule"
    script: "function
isAvailableForDealApplicationData(dealApplicationData) {
    var companyClass =
java.lang.Class.forName('com.keyintegrity.shb.company.query.dto.CompanyDto
');
    var clientCompany =
dealApplicationData.fetchFields[companyClass][dealApplicationData.result.c
lient.id];
        if (clientCompany == null) {
            return null;
        }
        var taxSystemClass =
java.lang.Class.forName('com.keyintegrity.shb.company.query.dto.catalog.Ta
xSystemCatalogDto');
        var taxSystemDto =
dealApplicationData.fetchFields[taxSystemClass][clientCompany.taxSystem];
        if (clientCompany.legalType == null || taxSystemDto ==
null || dealApplicationData.result.productDealState.amount == null) {
            return null;
        }
        var taxSystems = ['USN','ESHN'];
        return taxSystems.indexOf(taxSystemDto.code) !== -1 &&
clientCompany.legalType.id == 'ORGANIZATION' &&
dealApplicationData.result.productDealState.amount > 1000000 &&
dealApplicationData.result.productDealState.amount <= 10000000;
    }"
    - #docsPackage 5
    #name varchar(500) NOT NULL
    #Указать полное наименование пакета документов, включая наименование
продукта и сервиса для полноты понимания
    name: "Пакет БГ(И), Банк БАНК1: ЮЛ на УСН/Сельхоз, сумма БГ на
исполнение > 10 млн"
    #shortname varchar(50) NOT NULL

```

```

#Указать краткое наименование пакета документов, чтобы по нему понять
условия применимости пакета документов
shortName: "ЮЛ на УСН/Сельхоз, сумма > 10 млн"
#description varchar(1000) NULL, описание пакета документов, которое
может выводиться как подсказка в системе
description: NULL
#полный перечень типов документов см. в docs_types
#списком перечисляются мнемокоды типов обязательных документов в
пакете
#полный перечень см. в справочнике docs_types
docsTypes:
- "BREAKDOWN_ACCOUNTS_RECEIVABLE_LAST_REPORTING_DATE"
- "TAX_DECLARATION_LAST_PERIOD"
#правило применимости пакета документов
applicabilityRule:
#name varchar(150) NOT NULL
#наименование правила применимости
name: "ЮЛ с налогообложением = (УСНО ИЛИ Сельхоз) И сумма БГ на
исполнение больше 10 млн"
#type varchar(50) NOT NULL, по умолчанию значение = "script"
type: "script"
#script character varying(4000) NOT NULL
#Скрипт правила в виде функции
isAvailableForDealApplicationData(dealApplicationData), возвращающей TRUE,
FALSE или NULL.
#Если функция возвращает true, то считается, что пакет документов
применим и при заполнении заявки у клиента будут запрошены документы в
составе пакета
#примеры готовых скриптов см. в документации - статья "Правила
применимости: productApplicabilityRule, applicabilityRule"
script: "function
isAvailableForDealApplicationData(dealApplicationData) {
    var companyClass =
java.lang.Class.forName('com.keyintegrity.shb.company.query.dto.CompanyDto
');
    var clientCompany =
dealApplicationData.fetchFields[companyClass][dealApplicationData.result.c
lient.id];
        if (clientCompany == null) {
            return null;
        }
        var taxSystemClass =
java.lang.Class.forName('com.keyintegrity.shb.company.query.dto.catalog.Ta
xSystemCatalogDto');
        var taxSystemDto =
dealApplicationData.fetchFields[taxSystemClass][clientCompany.taxSystem];
        if (clientCompany.legalType == null || taxSystemDto ==
null || dealApplicationData.result.productDealState.amount == null) {
            return null;
        }
        var taxSystems = ['USN','ESHN'];
        return taxSystems.indexOf(taxSystemDto.code) != -1 &&
clientCompany.legalType.id == 'ORGANIZATION' &&
dealApplicationData.result.productDealState.amount > 10000000;
    }"
- #docsPackage 6
#name varchar(500) NOT NULL
#Указать полное наименование пакета документов, включая наименование
продукта и сервиса для полноты понимания
name: "Пакет документов для АО (БГ на исполнение, БИН)"
#shortname varchar(50) NOT NULL
#Указать краткое наименование пакета документов, чтобы по нему понять
условия применимости пакета документов
shortName: "Пакет документов для АО (БГ, БИН)"
#description varchar(1000) NULL, описание пакета документов, которое
может выводиться как подсказка в системе

```

```

description: NULL
#полный перечень типов документов см. в docs_types
#списком перечисляются мнемокоды типов обязательных документов в
пакете
#полный перечень см. в справочнике docs_types
docsTypes:
- "EXTRACT_FROM_REGISTER_SHAREHOLDERS"
#правило применимости пакета документов
applicabilityRule:
#name varchar(150) NOT NULL
#наименование правила применимости
name: "Компания-клиент является акционерным обществом"
#type varchar(50) NOT NULL, по умолчанию значение = "script"
type: "script"
#script character varying(4000) NOT NULL
#Скрипт правила в виде функции
isAvailableForDealApplicationData(dealApplicationData), возвращающей TRUE,
FALSE или NULL.
#Если функция возвращает true, то считается, что пакет документов
применим и при заполнении заявки у клиента будут запрошены документы в
составе пакета
#примеры готовых скриптов см. в документации - статья "Правила
применимости: productApplicabilityRule, applicabilityRule"
script: "function
isAvailableForDealApplicationData(dealApplicationData) {
    if (dealApplicationData.result.client.clientType.id !=
'COMPANY') {
        return false;
    }
    var okopfTypes = ['12200', '12247', '12267'];
    var companyClass =
java.lang.Class.forName('com.keyintegrity.shb.company.query.dto.CompanyDto
');
    var clientCompany =
dealApplicationData.fetchFields[companyClass][dealApplicationData.result.c
lient.id];
    var okopfClass =
java.lang.Class.forName('com.keyintegrity.shb.company.query.dto.catalog.Ok
opfCatalogDto');
    var okopfDto =
dealApplicationData.fetchFields[okopfClass][clientCompany.okopf];
    return okopfDto != null &&
okopfTypes.indexOf(okopfDto.code) != -1;
}"
- #docsPackage 7
#name varchar(500) NOT NULL
#Указать полное наименование пакета документов, включая наименование
продукта и сервиса для полноты понимания
name: "Пакет документов для ЮЛ (БГ на исполнение, БИН)"
#shortname varchar(50) NOT NULL
#Указать краткое наименование пакета документов, чтобы по нему понять
условия применимости пакета документов
shortName: "Пакет документов для ЮЛ (БГ, БИН)"
#description varchar(1000) NULL, описание пакета документов, которое
может выводиться как подсказка в системе
description: NULL
#полный перечень типов документов см. в docs_types
#списком перечисляются мнемокоды типов обязательных документов в
пакете
#полный перечень см. в справочнике docs_types
docsTypes:
- "CHARTER"
- "HEAD_APPOINTMENT_PROTOCOL"
- "HEAD_PASSPORT_COPY"
#нет applicabilityRule -пакет применим безусловно для всего продукта

```

```

#-----
-----
#требования к фин. показателям
financePackages:
- #FinancePackage 1
  name: "Открытие БГ(И), до 1 млн"
  userTooltip: "Разрешается или заполнить финансовые показатели, или
прикрепить обязательные документы."
  #описываются требования к каждой фин. форме, на текущий момент
поддерживается два набора отчетности по две формы в каждом
  #набор отчетности КНД 0710098 с формами ОКУД 0710001 и ОКУД 0710002
  #набор отчетности КНД 0710099 с формами ОКУД 0710001 и ОКУД 0710002
  #полный перечень см. в справочнике finance_fields
  #в formSet подставляем значение из КНД набора форм
  #в form подставляем значение из Формы по ОКУД
  #в indicators.required подставляем значение из мнемокода
  #periodTypes определяет обязательный набор выводимых отчетных периодов
для заполнения отчетности
  #для заполнения periodTypes полный перечень см. в справочнике
period_types
forms:
- #form and indicators
  formSet: "0710098"
  form: "0710001"
  periodTypes:
  - "currentReportingPeriod"
  - "previousYear"
  - "yearPrecedingPrevious"
  indicators:
    required:
    - "noMaterial"
    - "material"
    - "1210"
    - "finAssets"
    - "1250"
    - "capital"
    - "1410"
    - "longTermDuties"
    - "1510"
    - "1520"
    - "shortTermDuties"
    - "1600"
    - "1700"
- #form and indicators
  formSet: "0710098"
  form: "0710002"
  periodTypes:
  - "currentReportingPeriod"
  - "previousYear"
  - "yearPrecedingPrevious"
  indicators:
    required:
    - "2110"
    - "expenses"
    - "otherIncomes"
    - "2330"
    - "2350"
    - "2400"
    - "incomeTaxes"
- #form and indicators
  formSet: "0710099"
  form: "0710001"
  periodTypes:
  - "currentReportingPeriod"
  - "previousYear"

```

```

- "yearPrecedingPrevious"
indicators:
  required:
    - "1100"
    - "1110"
    - "1120"
    - "1130"
    - "1140"
    - "1150"
    - "1160"
    - "1170"
    - "1180"
    - "1190"
    - "1200"
    - "1210"
    - "1220"
    - "1230"
    - "1240"
    - "1250"
    - "1260"
    - "1300"
    - "1310"
    - "1320"
    - "1340"
    - "1350"
    - "1360"
    - "1370"
    - "1400"
    - "1410"
    - "1420"
    - "1430"
    - "1450"
    - "1500"
    - "1510"
    - "1520"
    - "1530"
    - "1540"
    - "1550"
    - "1600"
    - "1700"
- #form and indicators
  formSet: "0710099"
  form: "0710002"
  periodTypes:
    - "currentReportingPeriod"
    - "previousYear"
    - "yearPrecedingPrevious"
  indicators:
    required:
      - "2100"
      - "2110"
      - "2120"
      - "2200"
      - "2210"
      - "2220"
      - "2300"
      - "2310"
      - "2320"
      - "2330"
      - "2340"
      - "2350"
      - "2400"
      - "2410"
      - "2411"
      - "2412"
      - "2460"

```

```

- "2500"
- "2510"
- "2520"
- "2530"
- "2900"
- "2910"

#правило применимости пакета фин. показателей
applicabilityRule:
  #name varchar(150) NOT NULL
  #наименование правила применимости
  name: "Сумма БГ < 1 млн И нет документов
FINANCE_REPORT_LAST_QUARTER, ANALYTICAL_BALANCE_LAST_REPORTING_YEAR"
  #type varchar(50) NOT NULL, по умолчанию значение = "script" type:
"script"
  #script character varying(4000) NOT NULL
  #Скрипт правила в виде функции
isAvailableForDealApplicationData(dealApplicationData), возвращающей TRUE
или FALSE.
  script: "function
isAvailableForDealApplicationData(dealApplicationData) {
  var clientCompany =
dealApplicationData.getFetchField('Company',
dealApplicationData.result.client.id);
  if (clientCompany == null ||
dealApplicationData.result.productDealState.amount == null) {
    return null;
  }
  var financeReportExists = false;
  var analyticalBalanceExists = false;
  var documentsMap =
dealApplicationData.getFetchFieldAsMap('ApplicationDocument');
  if (documentsMap !== null) {
    var documents = documentsMap.values();
    financeReportExists =
documents.stream().anyMatch(function(doc) {
      return doc.docType != null &&
doc.docType.id == 'FINANCE_REPORT_LAST_QUARTER' && doc.docState != null &&
doc.docState.id == 'ACTUAL';
    });
    analyticalBalanceExists =
documents.stream().anyMatch(function(doc) {
      return doc.docType != null &&
doc.docType.id == 'ANALYTICAL_BALANCE_LAST_REPORTING_YEAR' && doc.docState
!= null && doc.docState.id == 'ACTUAL';
    });
  }
  return clientCompany.legalType.id == 'ORGANIZATION' &&
!(financeReportExists && analyticalBalanceExists) &&
dealApplicationData.result.productDealState.amount < 1000000;
}"

```

Code Block 1 Пример конфигурационного файла с описанием требований в виде комментариев

5.5 II.2. Создание тарификатора в виде конфигурационного файла yaml

Порядок и особенности создания конфигурационного файла тарификатора показан на примере.

Нажмите, чтобы раскрыть/скрыть пример исходного тарификатора, по которому нужно создать конфиг. файл

Базовый размер вознаграждения по продуктам "Банковская гарантия на исполнение"

Вознаграждение за предоставление банковской гарантии, в %% от суммы гарантии (ТАРИФЫ)*

*Тариф применим к продуктам по 185 Ф3/615 ПП

Срок гарантии, дней		Сумма гарантии, руб									
		от 50 000 до 50 000	от 50 000 до 100 000	от 100 000,01 до 500 000	от 500 000,01 до 1 000 000	от 1 000 000,01 до 3 000 000	от 3 000 000,01 до 7 000 000	от 7 000 000,01 до 10 000 000	от 10 000 000,01 до 50 000 000	от 50 000 000,01 до 100 000 000	от 100 000 000,01 до 150 000 000
1-365	мин ставка	2400 руб	4200 руб	6000 руб	12000 руб	16800 руб	18000 руб	18000 руб	18000 руб	18000 руб	18000 руб
	% годовых			4,2	4,2	4,2	4,2	3,96	3,96	3,74	3,48
365-732	мин ставка	3000 руб	4800 руб	6000 руб	12000 руб	16800 руб	18000 руб	18000 руб	18000 руб	18000 руб	18000 руб
	% годовых			4,2	4,2	4,2	4,2	3,96	3,96	3,74	3,48
732-1095	мин ставка	3000 руб	6000 руб	6000 руб	12000 руб	16800 руб	18000 руб	18000 руб	18000 руб	18000 руб	18000 руб
	% годовых			4,2	4,2	4,2	4,2	3,96	3,96	3,74	3,48

Базовый размер вознаграждения по продуктам "Банковская гарантия на исполнение"

Вознаграждение за предоставление банковской гарантии, в %% от суммы гарантии (ТАРИФЫ)**

**Тариф применим к продуктам по 44,223-ФЗ, без аванса

Срок гарантии, дней		Сумма гарантии, руб									
		от 50 000 до 50 000	от 50 000 до 100 000	от 100 000,01 до 500 000	от 500 000,01 до 1 000 000	от 1 000 000,01 до 3 000 000	от 3 000 000,01 до 7 000 000	от 7 000 000,01 до 10 000 000	от 10 000 000,01 до 50 000 000	от 50 000 000,01 до 100 000 000	от 100 000 000,01 до 150 000 000
1-365	мин ставка	2000 руб	3500 руб	5000 руб	10000 руб	14000 руб	15000 руб	15000 руб	15000 руб	15000 руб	15000 руб
	% годовых			3,4	3,4	3,4	3,4	3,2	3,2	3,05	2,9
365-732	мин ставка	2500 руб	4000 руб	5000 руб	10000 руб	14000 руб	15000 руб	15000 руб	15000 руб	15000 руб	15000 руб
	% годовых			3,4	3,4	3,4	3,4	3,2	3,2	3,05	2,9
732-1095	мин ставка	2500 руб	5000 руб	5000 руб	10000 руб	14000 руб	15000 руб	15000 руб	15000 руб	15000 руб	15000 руб
	% годовых			3,4	3,4	3,4	3,4	3,2	3,2	3,05	2,9

Базовый размер вознаграждения по продуктам "Банковская гарантия на исполнение"

Вознаграждение за предоставление банковской гарантии, в %% от суммы гарантии (ТАРИФЫ)***

***Тариф применим к продуктам по 44,223-ФЗ, с авансом

Срок гарантии, дней		Сумма гарантии, руб									
		от 50 000 до 50 000	от 50 000 до 100 000	от 100 000,01 до 500 000	от 500 000,01 до 1 000 000	от 1 000 000,01 до 3 000 000	от 3 000 000,01 до 7 000 000	от 7 000 000,01 до 10 000 000	от 10 000 000,01 до 50 000 000	от 50 000 000,01 до 100 000 000	от 100 000 000,01 до 150 000 000
1-365	мин ставка	2000 руб	3500 руб	5000 руб	10000 руб	14000 руб	15000 руб	15000 руб	15000 руб	15000 руб	15000 руб
	% годовых			3,7	3,7	3,7	3,7	3,8	3,6	3,05	2,9
365-732	мин ставка	2500 руб	4000 руб	5000 руб	10000 руб	14000 руб	15000 руб	15000 руб	15000 руб	15000 руб	15000 руб
	% годовых			3,7	3,7	3,7	3,7	3,8	3,6	3,05	2,9
732-1095	мин ставка	2500 руб	5000 руб	5000 руб	10000 руб	14000 руб	15000 руб	15000 руб	15000 руб	15000 руб	15000 руб
	% годовых			3,7	3,7	3,7	3,7	3,8	3,6	3,05	2,9

Требования и особенности формирования конфигурационного файла описаны в виде комментариев в примере ниже.

ПРИМЕЧАНИЕ

Если для ИП и ЮЛ действуют одинаковые тарифы, то можно один и тот же конфигурационный файл использовать у продукта для ЮЛ и ИП.

ВНИМАНИЕ

Все значения полей, которые указываются в тарификаторе для элемента field, рекомендуется внести как обязательные в настройки продукта для гарантии того, что пользователь их заполнит и тариф будет посчитан.

Например, в примере ниже есть следующие варианты значений поля field:

"BG:productDealState.amount"
 "BG:productDealState.termDays"
 "BG:productDealState.tender.federalLaw"
 "BG:productDealState.isPrepayment"

Т.к. все эти значения типа "PRODUCT_FIELDS", то в настройках продукта перечисленные значения должны быть указаны в блоке productFields.required

ПРИМЕЧАНИЕ

Рекомендуется перед импортом проверить файл с помощью любого валидатора yamI, чтобы исключить явные ошибки форматирования.

Скачать пример можно также по ссылке [Пример для документации - Тарификатор Банка БАНК1 для БГ\(И\) \(ред. 06-06-2019\).yamI](#)

```

#-----
#-----
#тарифы, связанные с продуктом
#тарифов может быть несколько в зависимости от ограничений, для каждого
набора ограничений свой тариф, если тариф зависит от набора ограничений
tariffs:
- #tariff:
  #name varchar(250) NOT NULL
  name: "Тарифы по 185 ФЗ/615 ПП для банковской гарантии на исполнение,
БАНК1"
  #shortName varchar(50) NOT NULL
  #выводится в списке тарифов продукта в личном кабинете поставщика
сервиса
  shortName: "185 ФЗ/615 ПП, БГ(И), БАНК1"
  #description varchar(500) NULL
  #может использоваться при выводе подсказки по тарифу в списке тарифов
продукта в личном кабинете поставщика сервиса
  description: "185 ФЗ/615 ПП, БГ(И), БАНК1"
  #указание на тип продукта, для которого создается тариф, по нему
выполняется явная валидация связанных продуктов и ограничений внутри
тарифа
  #мнемокод типа продукта, выбирается из справочника product_types
  #Например, "BG" или "PARTICIPATION_BG" или "TENDER_LOAN"
  productType: "BG"
  #startDate date NOT NULL, Дата начала действия тарифа в системе в
формате "yyyy-mm-dd"
  startDate: "2017-01-01"
  #endDate date NULL, Дата окончания действия тарифа в системе, если
EndDate = Null и текущая дата > StartDate, то считаем, что тариф действует
  #endDate в формате "yyyy-mm-dd"
  endDate: NULL
  #раздел settings, состав блока зависит от типа тарификатора
  #на текущий момент поддерживается только тип тарифа = INTERNAL
  #тариф может быть встроенный = INTERNAL, при котором в goodfin
импортируются данные для расчета предварительного тарифа, который
выводится в предложениях клиенту
  #тариф может определяться по запросу к API сервиса в момент рассмотрения
предложения. Пока не реализовано
  #тариф может определяться сервисом только в момент рассмотрения заявки,
следовательно на нашей стороне мы можем только указать "Определяется в
сервисе". Пока не реализовано
  settings:
    #на текущий момент поддерживается только тип тарифа = INTERNAL
    type: "INTERNAL"
    #fieldColumn - это поле, по диапазонам значений которого будут
рисоваться колонки тарификатора
    #Нужно выбирать поле того типа продукта, для которого строится тариф
    #Поле также должно быть указано обязательным для продукта
    fieldColumn:
      #type varchar(50) NOT NULL - указывает на один из справочников -
источник поля field
      #по умолчанию, PRODUCT_FIELDS. Также может быть COMPANY_FIELDS
      type: "PRODUCT_FIELDS"
      #field varchar(100) NOT NULL указание на поле, участвующее в расчете
      #Например, "BG:productDealState.amount" или
"PARTICIPATION_BG:productDealState.amount" или
"TENDER_LOAN:productDealState.amount"
      #полный перечень возможных значений для типа COMPANY_FIELDS см. в
company_fields
      #полный перечень возможных значений для типа PRODUCT_FIELDS см. в
product_types_fields
      field: "BG:productDealState.amount"
    #fieldRow - это поле, по диапазонам значений которого будут рисоваться
строки тарификатора.

```

```

#Надо выбирать поле того типа продукта, для которого строится
тарификатор
#Поле также должно быть указано обязательным для продукта
fieldRow:
  #type varchar(50) NOT NULL - указывает на один из справочников -
источник поля field
  #по умолчанию, PRODUCT_FIELDS. Также может быть COMPANY_FIELDS
  type: "PRODUCT_FIELDS"
  #field varchar(100) NOT NULL указание на поле, участвующее в расчете
  #Например, "BG:productDealState.termDays" или
"PARTICIPATION_BG:productDealState.termDays" или
"TENDER_LOAN:productDealState.termDays"
  #полный перечень возможных значений для типа COMPANY_FIELDS см. в
company_fields
  #полный перечень возможных значений для типа PRODUCT_FIELDS см. в
product_types_fields
  field: "BG:productDealState.termDays"
  #calcField - это поле, которое является базовым для определения
тарифа.
  #От значения данного поля будут браться проценты, суммироваться
надбавки или вычитаться скидки и пр.
#Надо выбирать поле того типа продукта, для которого строится
тарификатор
#Поле также должно быть указано обязательным для продукта
calcField:
  #type varchar(50) NOT NULL - указывает на один из справочников -
источник поля field
  #по умолчанию, PRODUCT_FIELDS. Также может быть COMPANY_FIELDS
  type: "PRODUCT_FIELDS"
  #field varchar(100) NOT NULL указание на поле, участвующее в расчете
  #Например, "BG:productDealState.amount" или
"PARTICIPATION_BG:productDealState.amount" или
"TENDER_LOAN:productDealState.amount"
  #полный перечень возможных значений для типа COMPANY_FIELDS см. в
company_fields
  #полный перечень возможных значений для типа PRODUCT_FIELDS см. в
product_types_fields
  field: "BG:productDealState.amount"
#строки тарифа
tariffCalculatingRows:
#rowStartValue - указание, с какой величины начинается расчет
#rowStartValue < проверяемое значение <= rowEndValue, другими словами
диапазон указывается по правилу (rowStartValue,rowEndValue]
#rowName - имя строки, которое используется как код диапазона значений
строки тарифа при описании ячеек тарифа
-
  rowName: "row1_365"
  rowStartValue: 1
  rowEndValue: 365
-
  rowName: "row365_732"
  rowStartValue: 365
  rowEndValue: 732
-
  rowName: "row732_1095"
  rowStartValue: 732
  rowEndValue: 1095

#колонка тарифа
tariffCalculatingColumns:
#columnStartValue < период <= columnEndValue, другими словами диапазон
указывается по правилу (columnStartValue,columnEndValue]
#columnName - имя колонки, которое используется как код диапазона
значений колонки тарифа при описании ячеек тарифа
-
  columnName: "col0_50000"

```

```

columnStartValue: 0
columnEndValue: 50000
-
columnName: "col50000_100000"
columnStartValue: 50000
columnEndValue: 100000
-
columnName: "col100000_500000"
columnStartValue: 100000
columnEndValue: 500000
-
columnName: "col500000_1000000"
columnStartValue: 500000
columnEndValue: 1000000
-
columnName: "col1000000_3000000"
columnStartValue: 1000000
columnEndValue: 3000000
-
columnName: "col3000000_7000000"
columnStartValue: 3000000
columnEndValue: 7000000
-
columnName: "col7000000_10000000"
columnStartValue: 7000000
columnEndValue: 10000000
-
columnName: "col10000000_50000000"
columnStartValue: 10000000
columnEndValue: 50000000
-
columnName: "col50000000_100000000"
columnStartValue: 50000000
columnEndValue: 100000000
-
columnName: "col100000000_150000000"
columnStartValue: 100000000
columnEndValue: 150000000

tariffCells:
#rowName - ссылка на строку, NOT NULL
#columnName - ссылка на колонку, NOT NULL
#cellValue - ставка
#cellValueType - тип значения. Возможные варианты значений см. в
справочнике tariff_cell_types
#cellMinValue - минимальная ставка, может быть NULL
#cellMinValueType - тип значения, может быть NULL, если cellMinValue =
NULL. Возможные варианты значений см. в справочнике tariff_cell_types

- {rowName: "row1_365", columnName: "col0_50000", cellValue: 2400.00,
cellValueType: "RUB", cellMinValue: 2400.00, cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col50000_100000", cellValue:
4200.00, cellValueType: "RUB", cellMinValue: 4200.00, cellMinValueType:
"RUB"}
- {rowName: "row1_365", columnName: "col100000_500000", cellValue:
4.20, cellValueType: "ANNUAL_PERCENT", cellMinValue: 6000.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col500000_1000000", cellValue:
4.20, cellValueType: "ANNUAL_PERCENT", cellMinValue: 12000.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col1000000_3000000", cellValue:
4.20, cellValueType: "ANNUAL_PERCENT", cellMinValue: 16800.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col3000000_7000000", cellValue:
4.20, cellValueType: "ANNUAL_PERCENT", cellMinValue: 18000.00,
cellMinValueType: "RUB"}

```

```

- {rowName: "row1_365", columnName: "col7000000_10000000", cellValue:
3.96, cellValueType: "ANNUAL_PERCENT", cellMinValue: 18000.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col10000000_50000000", cellValue:
3.96, cellValueType: "ANNUAL_PERCENT", cellMinValue: 18000.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col50000000_100000000",
cellValue: 3.74, cellValueType: "ANNUAL_PERCENT", cellMinValue: 18000.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col100000000_150000000",
cellValue: 3.48, cellValueType: "ANNUAL_PERCENT", cellMinValue: 18000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col0_50000", cellValue:
3000.00, cellValueType: "RUB", cellMinValue: 3000.00, cellMinValueType:
"RUB"}
- {rowName: "row365_732", columnName: "col50000_100000", cellValue:
4800.00, cellValueType: "RUB", cellMinValue: 4800.00, cellMinValueType:
"RUB"}
- {rowName: "row365_732", columnName: "col100000_500000", cellValue:
4.20, cellValueType: "ANNUAL_PERCENT", cellMinValue: 6000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col500000_1000000", cellValue:
4.20, cellValueType: "ANNUAL_PERCENT", cellMinValue: 12000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col1000000_3000000", cellValue:
4.20, cellValueType: "ANNUAL_PERCENT", cellMinValue: 16800.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col3000000_7000000", cellValue:
4.20, cellValueType: "ANNUAL_PERCENT", cellMinValue: 18000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col7000000_10000000",
cellValue: 3.96, cellValueType: "ANNUAL_PERCENT", cellMinValue: 18000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col10000000_50000000",
cellValue: 3.96, cellValueType: "ANNUAL_PERCENT", cellMinValue: 18000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col50000000_100000000",
cellValue: 3.74, cellValueType: "ANNUAL_PERCENT", cellMinValue: 18000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col100000000_150000000",
cellValue: 3.48, cellValueType: "ANNUAL_PERCENT", cellMinValue: 18000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col0_50000", cellValue:
3000.00, cellValueType: "RUB", cellMinValue: 3000.00, cellMinValueType:
"RUB"}
- {rowName: "row732_1095", columnName: "col50000_100000", cellValue:
6000.00, cellValueType: "RUB", cellMinValue: 6000.00, cellMinValueType:
"RUB"}
- {rowName: "row732_1095", columnName: "col100000_500000", cellValue:
4.20, cellValueType: "ANNUAL_PERCENT", cellMinValue: 6000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col500000_1000000", cellValue:
4.20, cellValueType: "ANNUAL_PERCENT", cellMinValue: 12000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col1000000_3000000",
cellValue: 4.20, cellValueType: "ANNUAL_PERCENT", cellMinValue: 16800.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col3000000_7000000",
cellValue: 4.20, cellValueType: "ANNUAL_PERCENT", cellMinValue: 18000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col7000000_10000000",
cellValue: 3.96, cellValueType: "ANNUAL_PERCENT", cellMinValue: 18000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col10000000_50000000",
cellValue: 3.96, cellValueType: "ANNUAL_PERCENT", cellMinValue: 18000.00,
cellMinValueType: "RUB"}

```

```

- {rowName: "row732_1095", columnName: "col50000000_100000000",
cellValue: 3.74, cellValueType: "ANNUAL_PERCENT", cellMinValue: 18000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col100000000_150000000",
cellValue: 3.48, cellValueType: "ANNUAL_PERCENT", cellMinValue: 18000.00,
cellMinValueType: "RUB"}

tariffConstraints:
#блок повторяется столько раз, сколько ограничений в применимости
тарифа
#type и field - это блок для указания поля, по которому будет
определяться ограничение применимости тарифа
#Надо выбирать поле того типа продукта, для которого строится
тарификатор
#Поле также должно быть указано обязательным для продукта
#valueType varchar(50) NOT NULL. Это может быть простое перечисление
значений ENUM, может быть диапазон значений RANGE.
#для типа valueType = ENUM values оформляется списком допустимых
значений
#для типа valueType = RANGE values оформляется обязательным списком из
двух значений: начального и конечного
-
  type: "PRODUCT_FIELDS"
  field: "BG:productDealState.tender.federalLaw"
  valueType: "ENUM"
  values:
  - "185FL615PP"
- #tariff:
#name varchar(250) NOT NULL
name: "44,223-ФЗ, без аванса, БГ(И), БАНК1"
#shortName varchar(50) NOT NULL
#выводится в списке тарифов продукта в личном кабинете поставщика
сервиса
shortName: "44,223-ФЗ, без аванса, БГ(И), БАНК1"
#description varchar(500) NULL
#может использоваться при выводе подсказки по тарифу в списке тарифов
продукта в личном кабинете поставщика сервиса
description: "44,223-ФЗ, без аванса, БГ(И), БАНК1"
#указание на тип продукта, для которого создается тариф, по нему
выполняется явная валидация связанных продуктов и ограничений внутри
тарифа
#мнемокод типа продукта, выбирается из справочника product_types
#Например, "BG" или "PARTICIPATION_BG" или "TENDER_LOAN"
productType: "BG"
#startDate date NOT NULL, Дата начала действия тарифа в формате "yyyy-
mm-dd"
startDate: "2017-01-01"
#endDate date NULL, Дата окончания действия тарифа, если EndDate = Null
и текущая дата > StartDate, то считаем, что тариф действует
#endDate в формате "yyyy-mm-dd"
endDate: NULL
#раздел settings, состав блока зависит от типа тарификатора
#на текущий момент поддерживается только тип тарифа = INTERNAL
#тариф может быть встроенный = INTERNAL, при котором в goodfin
импортируются данные для расчета предварительного тарифа, который
выводится в предложениях клиенту
#тариф может определяться по запросу к API сервиса в момент рассмотрения
предложения. Пока не реализовано
#тариф может определяться сервисом только в момент рассмотрения заявки,
следовательно на нашей стороне мы можем только указать "Определяется в
сервисе". Пока не реализовано
settings:
#на текущий момент поддерживается только тип тарифа = INTERNAL
type: "INTERNAL"
#fieldColumn - это поле, по диапазонам значений которого будут
рисоваться колонки тарификатора

```

```

#Надо выбирать поле того типа продукта, для которого строится тариф
#Поле также должно быть указано обязательным для продукта
fieldColumn:
  #type varchar(50) NOT NULL - указывает на один из справочников -
источник поля field
  #по умолчанию, PRODUCT_FIELDS. Также может быть COMPANY_FIELDS
  type: "PRODUCT_FIELDS"
  #field varchar(100) NOT NULL указание на поле, участвующее в расчете
#Например, "BG:productDealState.amount" или
"PARTICIPATION_BG:productDealState.amount" или
"TENDER_LOAN:productDealState.amount"
  #полный перечень возможных значений для типа COMPANY_FIELDS см. в
company_fields
  #полный перечень возможных значений для типа PRODUCT_FIELDS см. в
product_types_fields
  field: "BG:productDealState.amount"
  #fieldRow - это поле, по диапазонам значений которого будут рисоваться
строки тарификатора.
#Надо выбирать поле того типа продукта, для которого строится
тарификатор
#Поле также должно быть указано обязательным для продукта
fieldRow:
  #type varchar(50) NOT NULL - указывает на один из справочников -
источник поля field
  #по умолчанию, PRODUCT_FIELDS. Также может быть COMPANY_FIELDS
  type: "PRODUCT_FIELDS"
  #field varchar(100) NOT NULL указание на поле, участвующее в расчете
#Например, "BG:productDealState.termDays" или
"PARTICIPATION_BG:productDealState.termDays" или
"TENDER_LOAN:productDealState.termDays"
  #полный перечень возможных значений для типа COMPANY_FIELDS см. в
company_fields
  #полный перечень возможных значений для типа PRODUCT_FIELDS см. в
product_types_fields
  field: "BG:productDealState.termDays"
  #calcField - это поле, которое является базовым для определения
тарифа.
#От значения данного поля будут браться проценты, суммироваться
надбавки или вычитаться скидки и пр.
#Надо выбирать поле того типа продукта, для которого строится
тарификатор
#Поле также должно быть указано обязательным для продукта
calcField:
  #type varchar(50) NOT NULL - указывает на один из справочников -
источник поля field
  #по умолчанию, PRODUCT_FIELDS. Также может быть COMPANY_FIELDS
  type: "PRODUCT_FIELDS"
  #field varchar(100) NOT NULL указание на поле, участвующее в расчете
#Например, "BG:productDealState.amount" или
"PARTICIPATION_BG:productDealState.amount" или
"TENDER_LOAN:productDealState.amount"
  #полный перечень возможных значений для типа COMPANY_FIELDS см. в
company_fields
  #полный перечень возможных значений для типа PRODUCT_FIELDS см. в
product_types_fields
  field: "BG:productDealState.amount"
  #строки тарифа
tariffCalculatingRows:
  #rowStartValue - указание, с какой величины начинается расчет
  #rowStartValue < проверяемое значение <= rowEndValue, другими словами
диапазон указывается по правилу (rowStartValue,rowEndValue]
  #rowName - имя строки, которое используется как код диапазона значений
строки тарифа при описании ячеек тарифа
  -
  rowName: "row1_365"
  rowStartValue: 1

```

```

    rowEndValue: 365
  -
    rowName: "row365_732"
    rowStartValue: 365
    rowEndValue: 732
  -
    rowName: "row732_1095"
    rowStartValue: 732
    rowEndValue: 1095

#колонка тарифа
tariffCalculatingColumns:
#columnStartValue < период <= columnEndValue, другими словами диапазон
указывается по правилу (columnStartValue,columnEndValue]
#columnName - имя колонки, которое используется как код диапазона
значений колонки тарифа при описании ячеек тарифа
  -
    columnName: "col0_50000"
    columnStartValue: 0
    columnEndValue: 50000
  -
    columnName: "col50000_100000"
    columnStartValue: 50000
    columnEndValue: 100000
  -
    columnName: "col100000_500000"
    columnStartValue: 100000
    columnEndValue: 500000
  -
    columnName: "col500000_1000000"
    columnStartValue: 500000
    columnEndValue: 1000000
  -
    columnName: "col1000000_3000000"
    columnStartValue: 1000000
    columnEndValue: 3000000
  -
    columnName: "col3000000_7000000"
    columnStartValue: 3000000
    columnEndValue: 7000000
  -
    columnName: "col7000000_10000000"
    columnStartValue: 7000000
    columnEndValue: 10000000
  -
    columnName: "col10000000_50000000"
    columnStartValue: 10000000
    columnEndValue: 50000000
  -
    columnName: "col50000000_100000000"
    columnStartValue: 50000000
    columnEndValue: 100000000
  -
    columnName: "col100000000_150000000"
    columnStartValue: 100000000
    columnEndValue: 150000000

tariffCells:
#rowName - ссылка на строку, NOT NULL
#columnName - ссылка на колонку, NOT NULL
#cellValue - ставка
#cellValueType - тип значения. Возможные варианты значений см. в
справочнике tariff_cell_types
#cellMinValue - минимальная ставка, может быть NULL
#cellMinValueType - тип значения, может быть NULL, если cellMinValue =
NULL. Возможные варианты значений см. в справочнике tariff_cell_types

```

```

- {rowName: "row1_365", columnName: "col0_50000", cellValue: 2000.00,
cellValueType: "RUB", cellMinValue: 2000.00, cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col50000_100000", cellValue:
3500.00, cellValueType: "RUB", cellMinValue: 3500.00, cellMinValueType:
"RUB"}
- {rowName: "row1_365", columnName: "col100000_500000", cellValue:
3.40, cellValueType: "ANNUAL_PERCENT", cellMinValue: 5000.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col500000_1000000", cellValue:
3.40, cellValueType: "ANNUAL_PERCENT", cellMinValue: 10000.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col1000000_3000000", cellValue:
3.40, cellValueType: "ANNUAL_PERCENT", cellMinValue: 14000.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col3000000_7000000", cellValue:
3.40, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col7000000_10000000", cellValue:
3.20, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col10000000_50000000", cellValue:
3.20, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col50000000_100000000",
cellValue: 3.05, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col100000000_150000000",
cellValue: 2.90, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col0_50000", cellValue:
2500.00, cellValueType: "RUB", cellMinValue: 2500.00, cellMinValueType:
"RUB"}
- {rowName: "row365_732", columnName: "col50000_100000", cellValue:
4000.00, cellValueType: "RUB", cellMinValue: 4500.00, cellMinValueType:
"RUB"}
- {rowName: "row365_732", columnName: "col100000_500000", cellValue:
3.40, cellValueType: "ANNUAL_PERCENT", cellMinValue: 5000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col500000_1000000", cellValue:
3.40, cellValueType: "ANNUAL_PERCENT", cellMinValue: 10000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col1000000_3000000", cellValue:
3.40, cellValueType: "ANNUAL_PERCENT", cellMinValue: 14000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col3000000_7000000", cellValue:
3.40, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col7000000_10000000",
cellValue: 3.20, cellValueType: "ANNUAL_PERCENT", cellMinValue: 16000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col10000000_50000000",
cellValue: 3.20, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col50000000_100000000",
cellValue: 3.05, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col100000000_150000000",
cellValue: 2.90, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col0_50000", cellValue:
2500.00, cellValueType: "RUB", cellMinValue: 2500.00, cellMinValueType:
"RUB"}
- {rowName: "row732_1095", columnName: "col50000_100000", cellValue:
5000.00, cellValueType: "RUB", cellMinValue: 5000.00, cellMinValueType:
"RUB"}

```

```

- {rowName: "row732_1095", columnName: "col100000_500000", cellValue:
3.40, cellValueType: "ANNUAL_PERCENT", cellMinValue: 5000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col500000_1000000", cellValue:
3.40, cellValueType: "ANNUAL_PERCENT", cellMinValue: 10000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col1000000_3000000",
cellValue: 3.40, cellValueType: "ANNUAL_PERCENT", cellMinValue: 14000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col3000000_7000000",
cellValue: 3.40, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col7000000_10000000",
cellValue: 3.20, cellValueType: "ANNUAL_PERCENT", cellMinValue: 16000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col10000000_50000000",
cellValue: 3.20, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col50000000_100000000",
cellValue: 3.05, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col100000000_150000000",
cellValue: 2.90, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
tariffConstraints:
#блок повторяется столько раз, сколько ограничений в применимости
тарифа
#type и field - это блок для указания поля, по которому будет
определяться ограничение применимости тарифа
#Надо выбирать поле того типа продукта, для которого строится
тарификатор
#Поле также должно быть указано обязательным для продукта
#valueType varchar(50) NOT NULL. Это может быть простое перечисление
значений ENUM, может быть диапазон значений RANGE.
#для типа valueType = ENUM values оформляется списком допустимых
значений
#для типа valueType = RANGE values оформляется обязательным списком из
двух значений: начального и конечного
-
  type: "PRODUCT_FIELDS"
  field: "BG:productDealState.isPrepayment"
  valueType: "ENUM"
  values:
  - "false"
-
  type: "PRODUCT_FIELDS"
  field: "BG:productDealState.tender.federalLaw"
  valueType: "ENUM"
  values:
  - "223FL"
  - "44FL"
- #tariff:
#name varchar(250) NOT NULL
name: "44,223-ФЗ, с авансом, БГ(И), БАНК1"
#shortName varchar(50) NOT NULL
#выводится в списке тарифов продукта в личном кабинете поставщика
сервиса
shortName: "44,223-ФЗ, с авансом, БГ(И), БАНК1"
#description varchar(500) NULL
#может использоваться при выводе подсказки по тарифу в списке тарифов
продукта в личном кабинете поставщика сервиса
description: "44,223-ФЗ, с авансом, БГ(И), БАНК1"
#указание на тип продукта, для которого создается тариф, по нему
выполняется явная валидация связанных продуктов и ограничений внутри
тарифа
#мнемокод типа продукта, выбирается из справочника product_types

```

```

#Например, "BG" или "PARTICIPATION_BG" или "TENDER_LOAN"
productType: "BG"
#startDate date NOT NULL, Дата начала действия тарифа в формате "yyyy-
mm-dd"
startDate: "2017-01-01"
#endDate date NULL, Дата окончания действия тарифа, если EndDate = Null
и текущая дата > StartDate, то считаем, что тариф действует
#endDate в формате "yyyy-mm-dd"
endDate: NULL
#раздел settings, состав блока зависит от типа тарификатора
#на текущий момент поддерживается только тип тарифа = INTERNAL
#тариф может быть встроенный = INTERNAL, при котором в goodfin
импортируются данные для расчета предварительного тарифа, который
выводится в предложениях клиенту
#тариф может определяться по запросу к API сервиса в момент рассмотрения
предложения. Пока не реализовано
#тариф может определяться сервисом только в момент рассмотрения заявки,
следовательно на нашей стороне мы можем только указать "Определяется в
сервисе". Пока не реализовано
settings:
#на текущий момент поддерживается только тип тарифа = INTERNAL
type: "INTERNAL"
#fieldColumn - это поле, по диапазонам значений которого будут
рисоваться колонки тарификатора
#Нужно выбирать поле того типа продукта, для которого строится тариф
#Поле также должно быть указано обязательным для продукта
fieldColumn:
#type varchar(50) NOT NULL - указывает на один из справочников -
источник поля field
#по умолчанию, PRODUCT_FIELDS. Также может быть COMPANY_FIELDS
type: "PRODUCT_FIELDS"
#field varchar(100) NOT NULL указание на поле, участвующее в расчете
#Например, "BG:productDealState.amount" или
"PARTICIPATION_BG:productDealState.amount" или
"TENDER_LOAN:productDealState.amount"
#полный перечень возможных значений для типа COMPANY_FIELDS см. в
company_fields
#полный перечень возможных значений для типа PRODUCT_FIELDS см. в
product_types_fields
field: "BG:productDealState.amount"
#fieldRow - это поле, по диапазонам значений которого будут рисоваться
строки тарификатора.
#Нужно выбирать поле того типа продукта, для которого строится
тарификатор
#Поле также должно быть указано обязательным для продукта
fieldRow:
#type varchar(50) NOT NULL - указывает на один из справочников -
источник поля field
#по умолчанию, PRODUCT_FIELDS. Также может быть COMPANY_FIELDS
type: "PRODUCT_FIELDS"
#field varchar(100) NOT NULL указание на поле, участвующее в расчете
#Например, "BG:productDealState.termDays" или
"PARTICIPATION_BG:productDealState.termDays" или
"TENDER_LOAN:productDealState.termDays"
#полный перечень возможных значений для типа COMPANY_FIELDS см. в
company_fields
#полный перечень возможных значений для типа PRODUCT_FIELDS см. в
product_types_fields
field: "BG:productDealState.termDays"
#calcField - это поле, которое является базовым для определения
тарифа.
#От значения данного поля будут браться проценты, суммироваться
надбавки или вычитаться скидки и пр.
#Нужно выбирать поле того типа продукта, для которого строится
тарификатор
#Поле также должно быть указано обязательным для продукта

```

```

calcField:
  #type varchar(50) NOT NULL - указывает на один из справочников -
  источник поля field
  #по умолчанию, PRODUCT_FIELDS. Также может быть COMPANY_FIELDS
  type: "PRODUCT_FIELDS"
  #field varchar(100) NOT NULL указание на поле, участвующее в расчете
  #Например, "BG:productDealState.amount" или
  "PARTICIPATION_BG:productDealState.amount" или
  "TENDER_LOAN:productDealState.amount"
  #полный перечень возможных значений для типа COMPANY_FIELDS см. в
  company_fields
  #полный перечень возможных значений для типа PRODUCT_FIELDS см. в
  product_types_fields
  field: "BG:productDealState.amount"
  #строки тарифа
  tariffCalculatingRows:
    #rowStartValue - указание, с какой величины начинается расчет
    #rowStartValue < проверяемое значение <= rowEndValue, другими словами
    диапазон указывается по правилу (rowStartValue,rowEndValue]
    #rowName - имя строки, которое используется как код диапазона значений
    строки тарифа при описании ячеек тарифа
    -
      rowName: "row1_365"
      rowStartValue: 1
      rowEndValue: 365
    -
      rowName: "row365_732"
      rowStartValue: 365
      rowEndValue: 732
    -
      rowName: "row732_1095"
      rowStartValue: 732
      rowEndValue: 1095

  #колонка тарифа
  tariffCalculatingColumns:
    #columnStartValue < период <= columnEndValue, другими словами диапазон
    указывается по правилу (columnStartValue,columnEndValue]
    #columnName - имя колонки, которое используется как код диапазона
    значений колонки тарифа при описании ячеек тарифа
    -
      columnName: "col0_50000"
      columnStartValue: 0
      columnEndValue: 50000
    -
      columnName: "col50000_100000"
      columnStartValue: 50000
      columnEndValue: 100000
    -
      columnName: "col100000_500000"
      columnStartValue: 100000
      columnEndValue: 500000
    -
      columnName: "col500000_1000000"
      columnStartValue: 500000
      columnEndValue: 1000000
    -
      columnName: "col1000000_3000000"
      columnStartValue: 1000000
      columnEndValue: 3000000
    -
      columnName: "col3000000_7000000"
      columnStartValue: 3000000
      columnEndValue: 7000000
    -
      columnName: "col7000000_10000000"

```

```

columnStartValue: 7000000
columnEndValue: 10000000
-
columnName: "col10000000_50000000"
columnStartValue: 10000000
columnEndValue: 50000000
-
columnName: "col50000000_100000000"
columnStartValue: 50000000
columnEndValue: 100000000
-
columnName: "col100000000_150000000"
columnStartValue: 100000000
columnEndValue: 150000000

tariffCells:
#rowName - ссылка на строку, NOT NULL
#columnName - ссылка на колонку, NOT NULL
#cellValue - ставка
#cellValueType - тип значения. Возможные варианты значений см. в
справочнике tariff_cell_types
#cellMinValue - минимальная ставка, может быть NULL
#cellMinValueType - тип значения, может быть NULL, если cellMinValue =
NULL. Возможные варианты значений см. в справочнике tariff_cell_types
- {rowName: "row1_365", columnName: "col0_50000", cellValue: 2000.00,
cellValueType: "RUB", cellMinValue: 2000.00, cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col50000_100000", cellValue:
3500.00, cellValueType: "RUB", cellMinValue: 3500.00, cellMinValueType:
"RUB"}
- {rowName: "row1_365", columnName: "col100000_500000", cellValue:
3.70, cellValueType: "ANNUAL_PERCENT", cellMinValue: 5000.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col500000_1000000", cellValue:
3.70, cellValueType: "ANNUAL_PERCENT", cellMinValue: 10000.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col1000000_3000000", cellValue:
3.70, cellValueType: "ANNUAL_PERCENT", cellMinValue: 14000.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col3000000_7000000", cellValue:
3.70, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col7000000_10000000", cellValue:
3.80, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col10000000_50000000", cellValue:
3.60, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col50000000_100000000",
cellValue: 3.05, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row1_365", columnName: "col100000000_150000000",
cellValue: 2.90, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col0_50000", cellValue:
2500.00, cellValueType: "RUB", cellMinValue: 2500.00, cellMinValueType:
"RUB"}
- {rowName: "row365_732", columnName: "col50000_100000", cellValue:
4000.00, cellValueType: "RUB", cellMinValue: 4500.00, cellMinValueType:
"RUB"}
- {rowName: "row365_732", columnName: "col100000_500000", cellValue:
3.70, cellValueType: "ANNUAL_PERCENT", cellMinValue: 5000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col500000_1000000", cellValue:
3.70, cellValueType: "ANNUAL_PERCENT", cellMinValue: 10000.00,
cellMinValueType: "RUB"}

```

```

- {rowName: "row365_732", columnName: "col1000000_3000000", cellValue:
3.70, cellValueType: "ANNUAL_PERCENT", cellMinValue: 14000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col3000000_7000000", cellValue:
3.70, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col7000000_10000000",
cellValue: 3.80, cellValueType: "ANNUAL_PERCENT", cellMinValue: 16000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col10000000_50000000",
cellValue: 3.60, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col50000000_100000000",
cellValue: 3.05, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row365_732", columnName: "col100000000_150000000",
cellValue: 2.90, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col0_50000", cellValue:
2500.00, cellValueType: "RUB", cellMinValue: 2500.00, cellMinValueType:
"RUB"}
- {rowName: "row732_1095", columnName: "col50000_100000", cellValue:
5000.00, cellValueType: "RUB", cellMinValue: 5000.00, cellMinValueType:
"RUB"}
- {rowName: "row732_1095", columnName: "col100000_500000", cellValue:
3.70, cellValueType: "ANNUAL_PERCENT", cellMinValue: 5000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col500000_1000000", cellValue:
3.70, cellValueType: "ANNUAL_PERCENT", cellMinValue: 10000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col1000000_3000000",
cellValue: 3.70, cellValueType: "ANNUAL_PERCENT", cellMinValue: 14000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col3000000_7000000",
cellValue: 3.70, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col7000000_10000000",
cellValue: 3.80, cellValueType: "ANNUAL_PERCENT", cellMinValue: 16000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col10000000_50000000",
cellValue: 3.60, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col50000000_100000000",
cellValue: 3.05, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
- {rowName: "row732_1095", columnName: "col100000000_150000000",
cellValue: 2.90, cellValueType: "ANNUAL_PERCENT", cellMinValue: 15000.00,
cellMinValueType: "RUB"}
tariffConstraints:
#блок повторяется столько раз, сколько ограничений в применимости
тарифа
#type и field - это блок для указания поля, по которому будет
определяться ограничение применимости тарифа
#Нужно выбирать поле того типа продукта, для которого строится
тарификатор
#Поле также должно быть указано обязательным для продукта
#valueType varchar(50) NOT NULL. Это может быть простое перечисление
значений ENUM, может быть диапазон значений RANGE.
#для типа valueType = ENUM values оформляется списком допустимых
значений
#для типа valueType = RANGE values оформляется обязательным списком из
двух значений: начального и конечного
-
type: "PRODUCT_FIELDS"
field: "BG:productDealState.isPrepayment"
valueType: "ENUM"

```

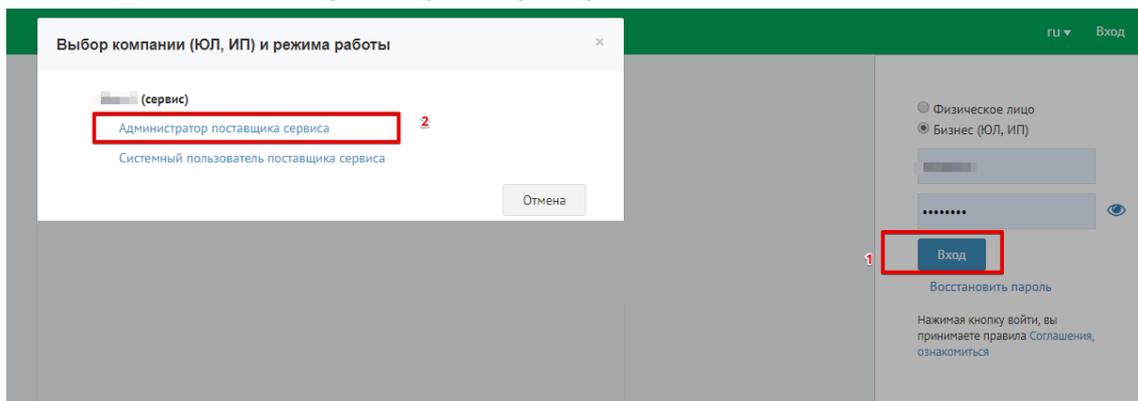
```

values:
- "true"
-
  type: "PRODUCT_FIELDS"
  field: "BG:productDealState.tender.federalLaw"
  valueType: "ENUM"
  values:
  - "223FL"
  - "44FL"
    
```

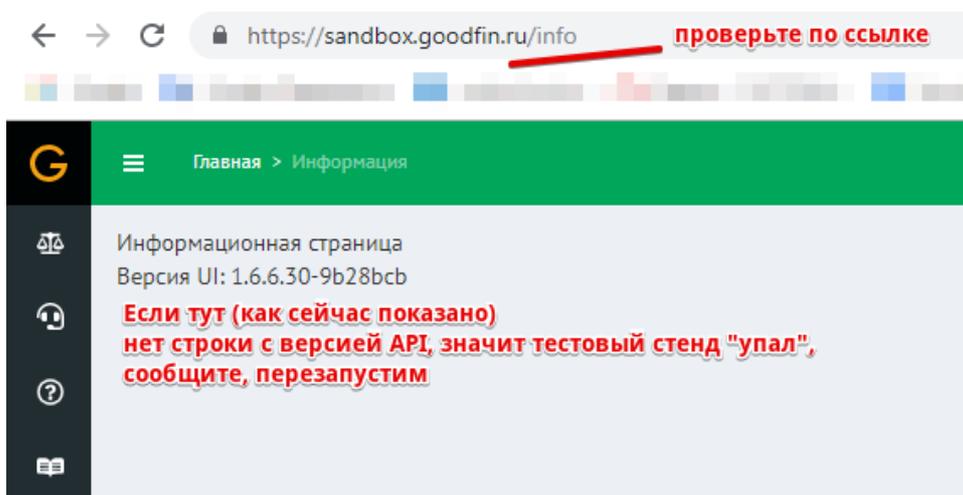
5.6 II.3. Тестирование и доработка банковского продукта

Шаг 1. В тестовом окружении по адресу <https://sandbox.goodfin.ru> войти под учетной записью user_<serviceSystemName> (см. правила получения логина на шаге 1 в [Подготовительные работы по интеграции с применением API](#)) в личный кабинет поставщика сервиса с ролью "Администратор сервиса".

Нажмите, чтобы посмотреть/скрыть пример

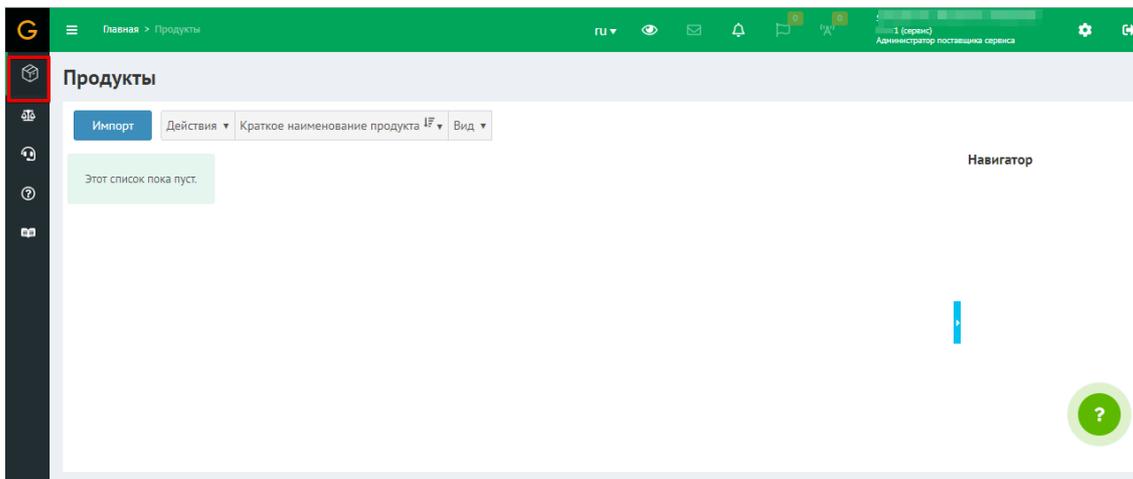


Если возникает ошибка входа, то возможно что-то с сервером пошло не так. Выполните проверку, как показано на рисунке ниже:



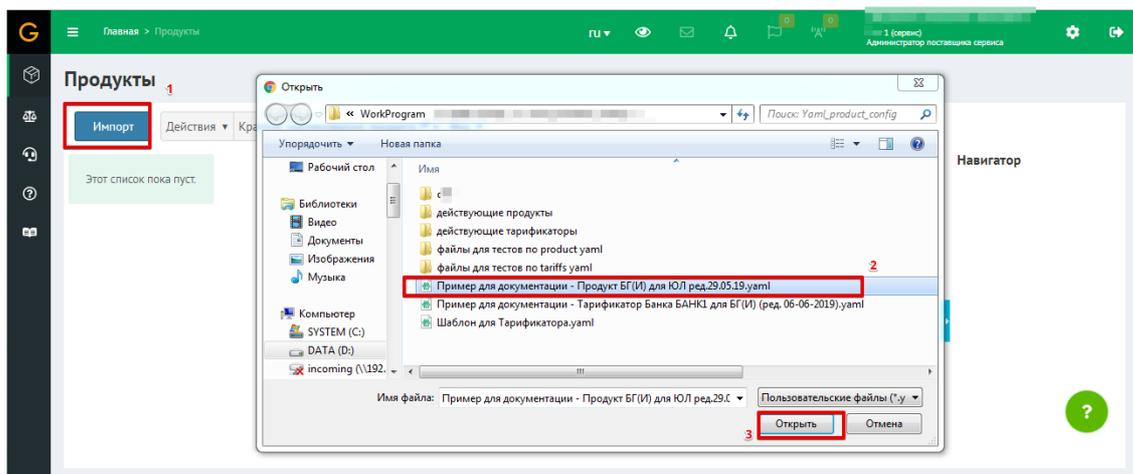
Шаг 2. После входа в личный кабинет поставщика сервиса перейдите к разделу "Продукты". Если Вы входите впервые и не импортировали еще ни одного продукта, то раздел будет пустой.

Нажмите, чтобы посмотреть/скрыть пример



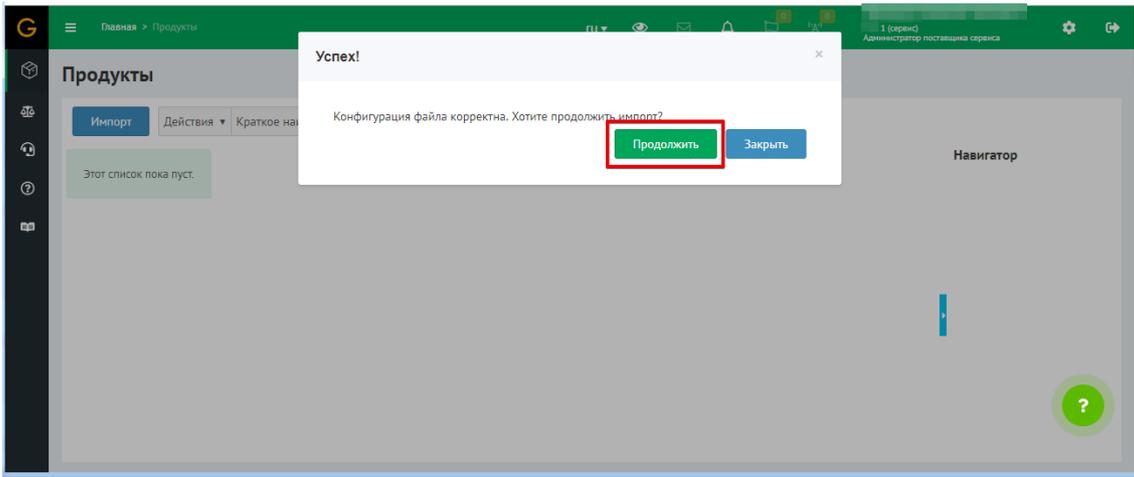
Шаг 3. Импортируйте созданный вами (согласно инструкции [II.1. Создание продукта в виде конфигурационного файла yam!](#)) конфигурационный файл продукта. Будет выполнена проверка синтаксической корректности файла.

Нажмите, чтобы посмотреть/скрыть пример



Шаг 4. Если проверка прошла успешно, то система предложит закончить импорт.

Нажмите, чтобы посмотреть/скрыть пример

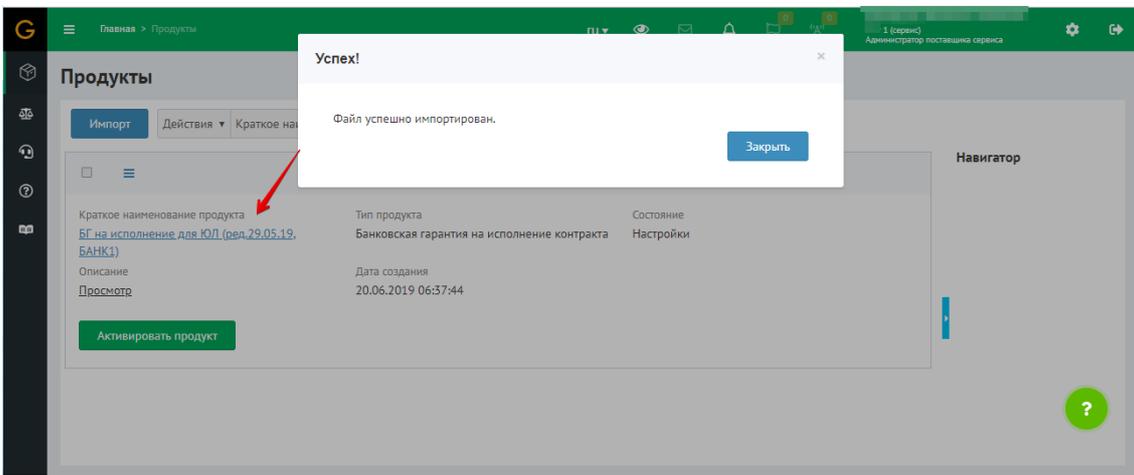


Шаг 5. Результат импорта вы увидите в списке продуктов.

ВНИМАНИЕ!

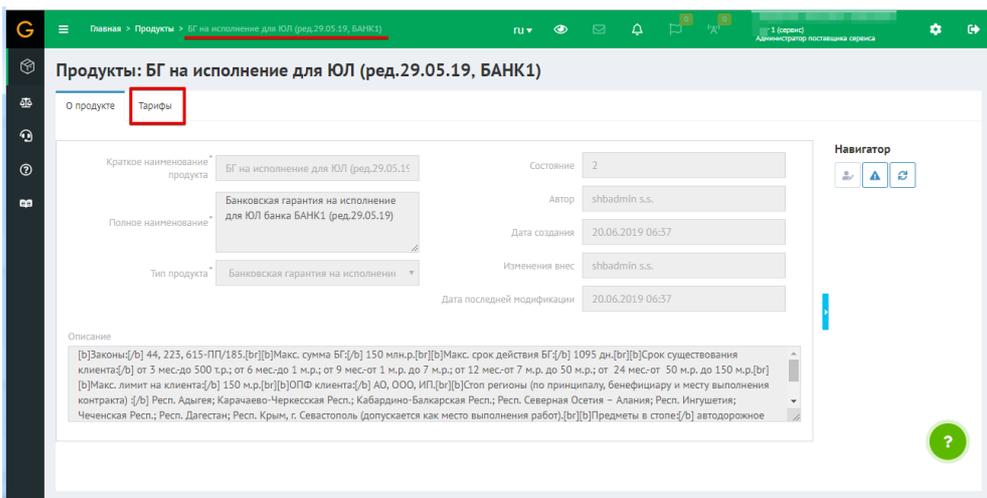
Импортированный продукт, по умолчанию, отключен и не доступен для выбора клиентом в предложениях. Продукт станет доступен только после нажатия кнопки "Активировать продукт". Но пока к продукту не добавлено ни одного тарифа, пользователь не сможет по данному продукту отправить заявку в сервис.

Нажмите, чтобы посмотреть/скрыть пример



Шаг 6. Для добавления/обновления тарифов по продукту перейдите в продукт.

Нажмите, чтобы посмотреть/скрыть пример

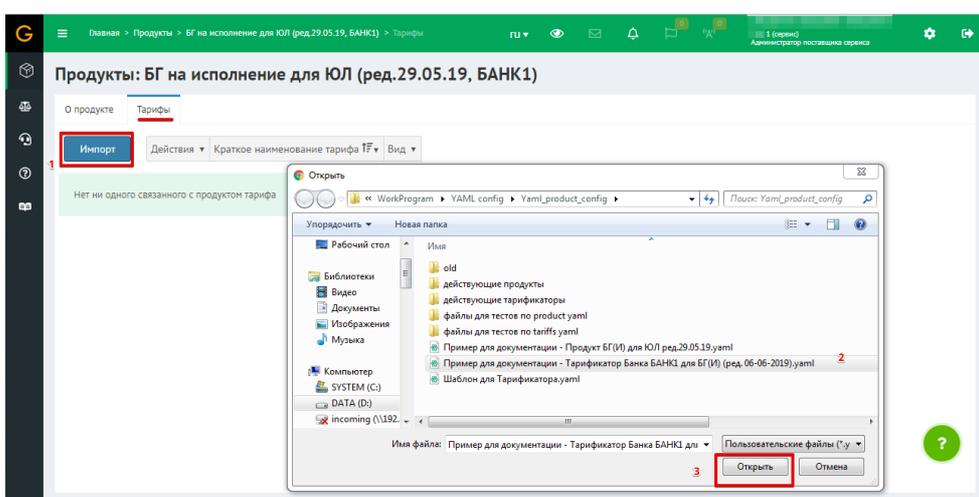


Шаг 7. Импортируйте созданный вами (согласно инструкции [II.2. Создание тарификатора в виде конфигурационного файла yam1](#)) конфигурационный файл тарификатора. Будет выполнена проверка синтаксической корректности файла.

ВНИМАНИЕ!

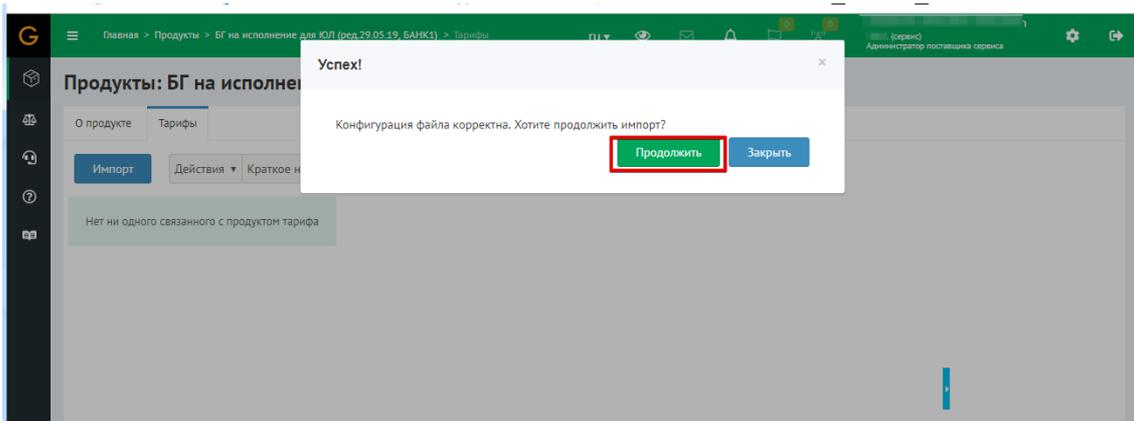
При повторном импорте тарификатора будут удалены все ранее имеющиеся тарифы у продукта и загружены новые из файла.

Нажмите, чтобы посмотреть/скрыть пример



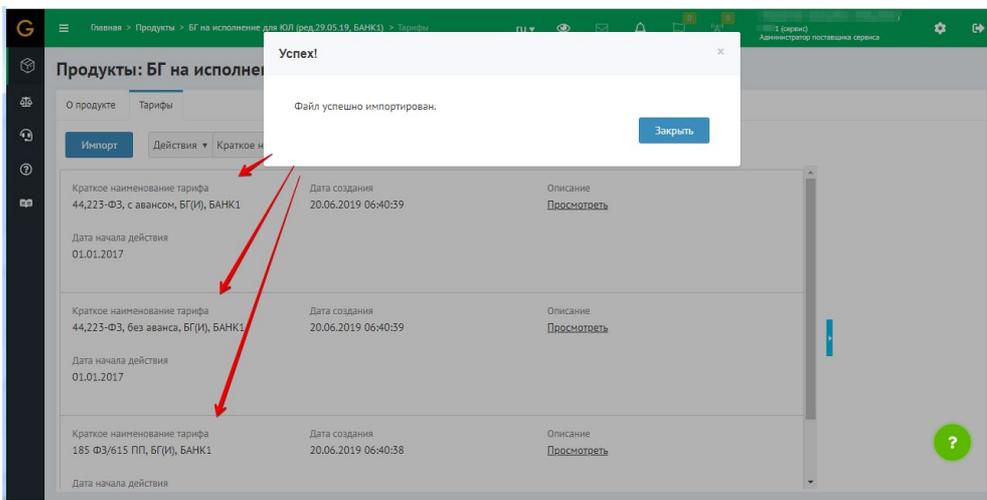
Шаг 8. Если проверка прошла успешно, то система предложит закончить импорт.

Нажмите, чтобы посмотреть/скрыть пример



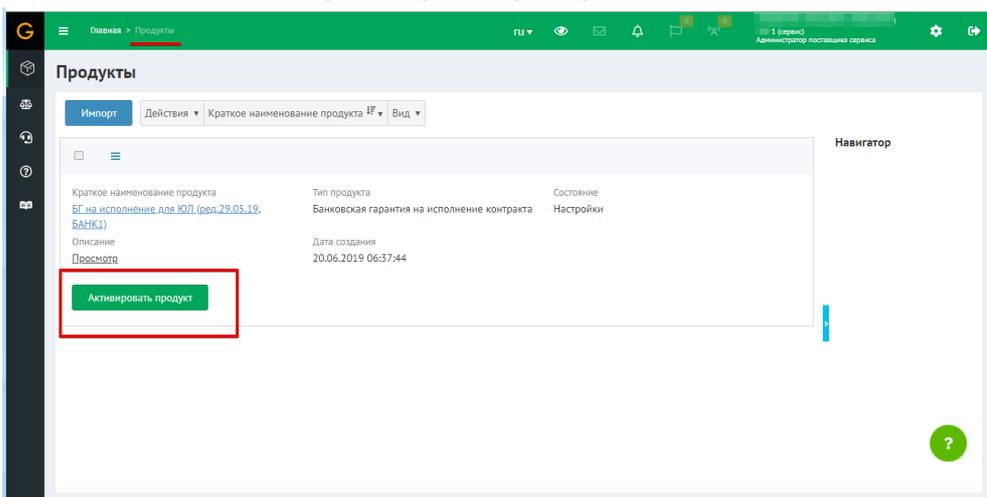
Шаг 9. Результат импорта вы увидите в списке тарифов продукта.

Нажмите, чтобы посмотреть/скрыть пример



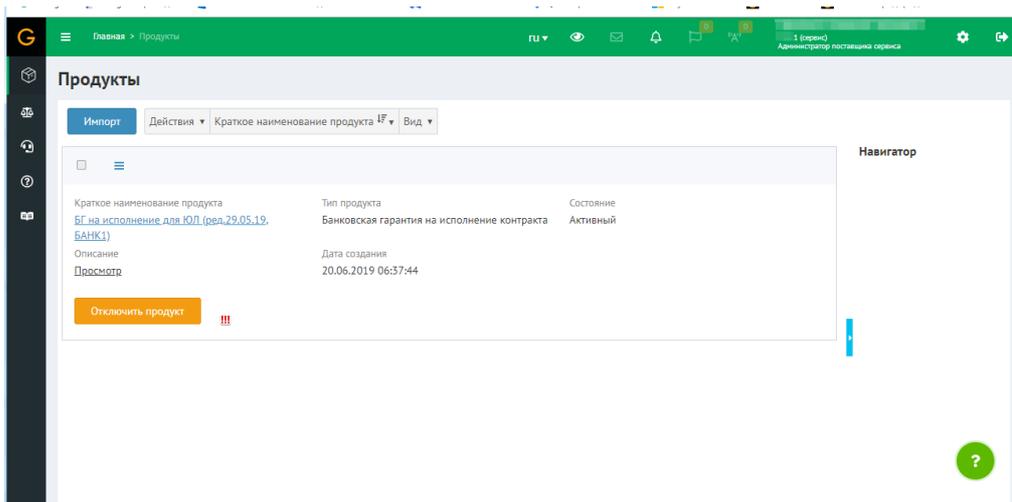
Шаг 10. Для тестирования продукта активируйте его.

Нажмите, чтобы посмотреть/скрыть пример



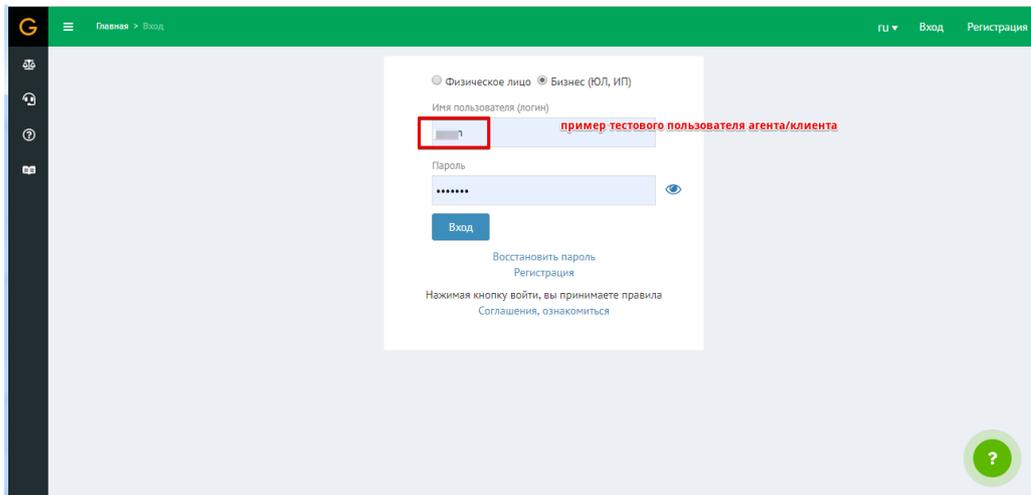
Шаг 11. Теперь продукт активен.

Нажмите, чтобы посмотреть/скрыть пример



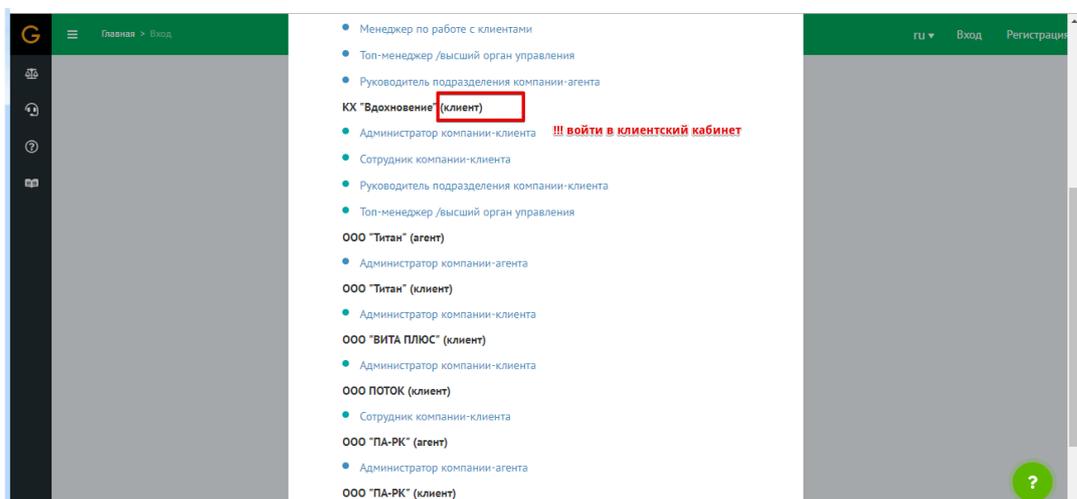
Шаг 12. Теперь требуется войти в систему как **клиент с помощью тестовой учетной записи клиента** `ss_<serviceSystemName>` (см. правила получения логина на шаге 1 в [Подготовительные работы по интеграции с применением API](#))

Нажмите, чтобы посмотреть/скрыть пример



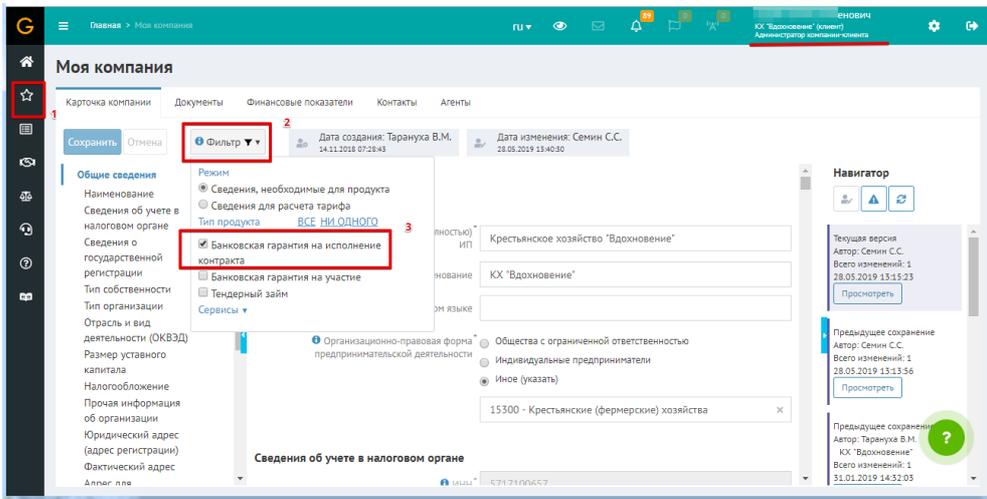
Шаг 13. Выбрать любую клиентскую роль.

Нажмите, чтобы посмотреть/скрыть пример



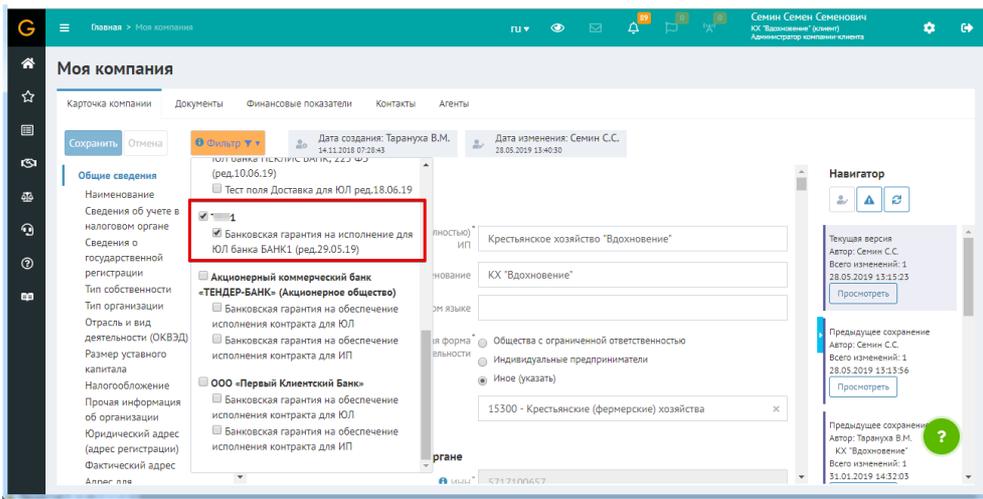
Шаг 14. Перейти в раздел "Моя компания" и в фильтре выбрать тот тип продукта, который тестируете.

Нажмите, чтобы посмотреть/скрыть пример

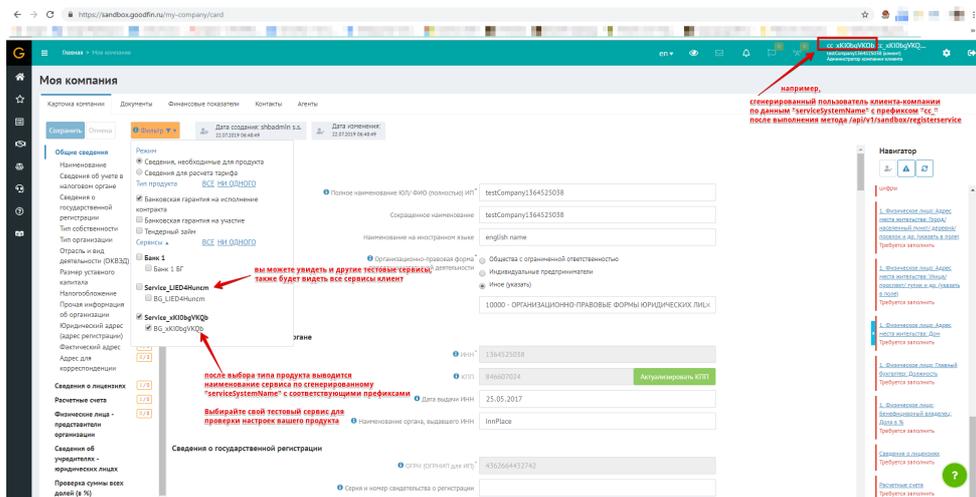


Шаг 15. Далее в фильтре выберите свой тестируемый продукт.

Нажмите, чтобы посмотреть/скрыть пример



На тестовом стенде наименования продуктов и сервисов будут выглядеть как в примере ниже:



Шаг 16. Для проверки обязательности заполнения полей карточки компании (в конфигурационном файле продукта это блок **companyFields.required**) вы должны очистить эти поля. Система в ответ должна подсветить их, показать в панели Категории счетчик ошибок, а в панели Навигатор ошибки типа "Требуется заполнить".

ВНИМАНИЕ!

Если вы обнаружили ошибки в настройке обязательности, то требуется исправить конфигурационный файл продукта, перезагрузить его (см. шаги 29-30). Затем снова вернуться в клиентский кабинет для тестирования исправленных настроек.

Нажмите, чтобы посмотреть/скрыть пример

The screenshot displays the 'Моя компания' (My Company) configuration interface. The main area contains a form with several fields, many of which are highlighted in orange and labeled 'Требуется заполнить' (Must be filled). The fields include: 'Дата государственной регистрации' (01.01.2000), 'Дата выдачи свидетельства о регистрации' (dd.mm.yyyy), 'Наименование регистрирующего органа' (Межрайонная Инспекция Министерства Российской Федерации), 'Место государственной регистрации' (Требуется заполнить), 'ОКХОФ (код и наименование)' (15300 - Крестьянские (фермерские) хозяйства), 'ОКПО (код)' (Требуется заполнить), 'ОКАТО (код и наименование)' (54236802001 - д Верхнее Алябеево), 'ОКТМО (код и наименование)' (54636402101 - д Верхнее Алябеево), and 'ОКОГУ (код и наименование)' (4210005 - Крестьянские (фермерские) хозяйства). The 'Навигатор' (Navigator) panel on the right shows a list of errors, including 'ОКПО (код) Требуется заполнить' and 'Место государственной регистрации Требуется заполнить'. Red arrows point from the Navigator to the corresponding fields in the main form.

Шаг 17. Проверка настроек пакетов документов. Если документы уже прикреплены, то вы можете перевести их в статус "Архивный" (для этого у документа нажмите на иконку "Редактировать", далее нажмите "Перевести в статус Архивный" и вернитесь к списку документов). В системе документы в статусе "Архивный" считаются утратившими актуальность. Тогда можно увидеть и проверить работу настроенного пакета документа без правила применимости (у такого пакета отсутствует блок applicabilityRule).

ПРИМЕЧАНИЕ

При настройке пакетов документов возможно два варианта реализации:

1. Без правила применимости (без блока applicabilityRule), все документы в составе такого пакета считаются безусловно обязательными. И именно такой пакет документов срабатывает в момент включения фильтра на вкладке Документы в карточке компании.
2. с правилом применимости (оно срабатывает в клиентском кабинете, когда клиент просматривает возможные предложения и уже заполнил карточку своей компании и данные сделки), см. пример в шаге 26.

Нажмите, чтобы посмотреть/скрыть пример

В примере настроек продукта (см. [II.1. Создание продукта в виде конфигурационного файла yam!](#)) есть один пакет документов без правила применимости:

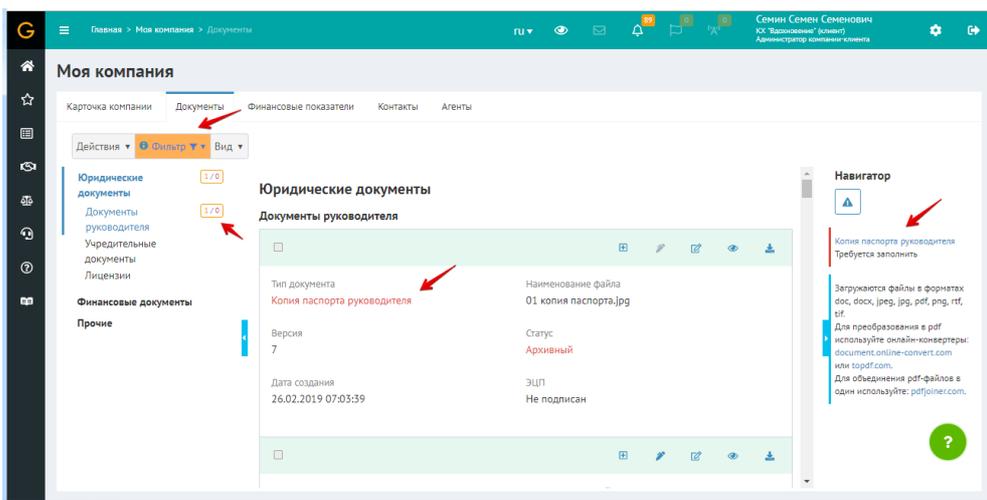
```

- #docsPackage 7
  #name varchar(500) NOT NULL
  #Указать полное наименование пакета документов, включая наименование
  продукта и сервиса для полноты понимания
  name: "Пакет документов для ЮЛ (БГ на исполнение, БИН)"
  #shortname varchar(50) NOT NULL
  #Указать краткое наименование пакета документов, чтобы по нему понять
  условия применимости пакета документов
  shortName: "Пакет документов для ЮЛ (БГ, БИН)"
  #description varchar(1000) NULL, описание пакета документов, которое
  может выводиться как подсказка в системе
  description: NULL
  #полный перечень типов документов см. в docs_types
  #списком перечисляются мнемокоды типов обязательных документов в
  пакете
  #полный перечень см. в справочнике docs_types
  docsTypes:
  - "CHARTER"
  - "HEAD_APPOINTMENT_PROTOCOL"
  - "HEAD_PASSPORT_COPY"
  #нет applicabilityRule -пакет применим безусловно для всего продукта

```

Code Block 2 Фрагмент конфигурационного файла

Так при включенном фильтре показано срабатывание пакета документов, система требует прикрепить копию паспорта (HEAD_PASSPORT_COPY), т.к. он в статусе "Архивный":



Шаг 18. Проверка настроек фин. показателей. Чтобы проверить настройки обязательности отчетных периодов и показателей, вы можете очистить ячейки с данными.

При включенном фильтре система должна подсветить отчетные периоды (за это отвечает блок настроек продукта periodTypes), а также конкретные показатели (за это отвечает блок indicators.required).

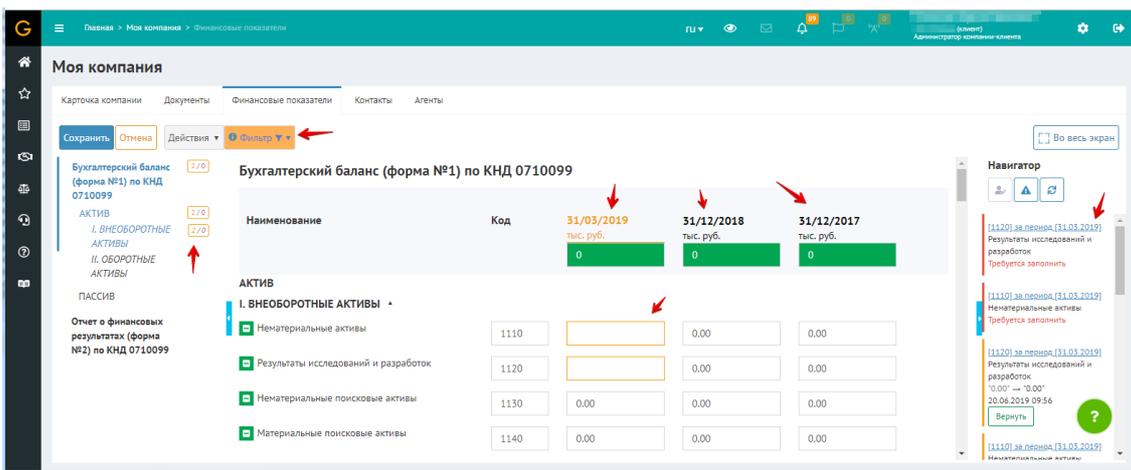
Нажмите, чтобы посмотреть/скрыть пример

В примере настроек продукта (см. [II.1. Создание продукта в виде конфигурационного файла yam!](#)) для Формы 1 по общей системе налогообложения настроены три периода.

```

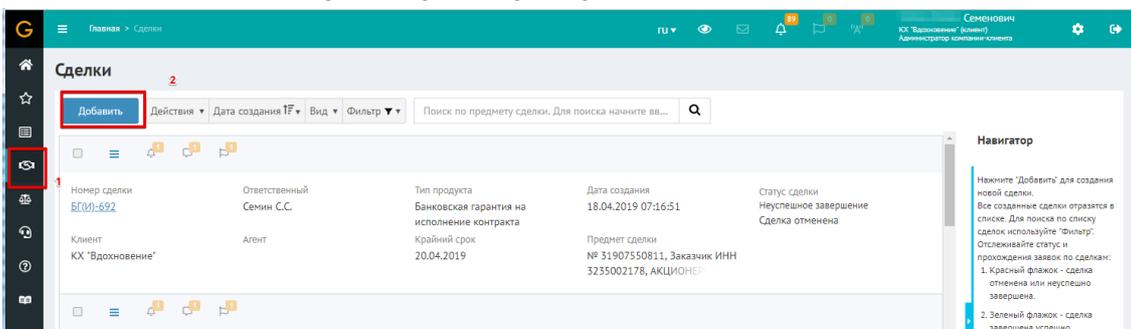
- #form and indicators
  formSet: "0710099"
  form: "0710001"
  periodTypes:
  - "currentReportingPeriod"
  - "previousYear"
  - "yearPrecedingPrevious"
    
```

В примере ниже показано, что при очищении нескольких показателей автоматически включается подсветка, указывающая на обязательность заполнения.



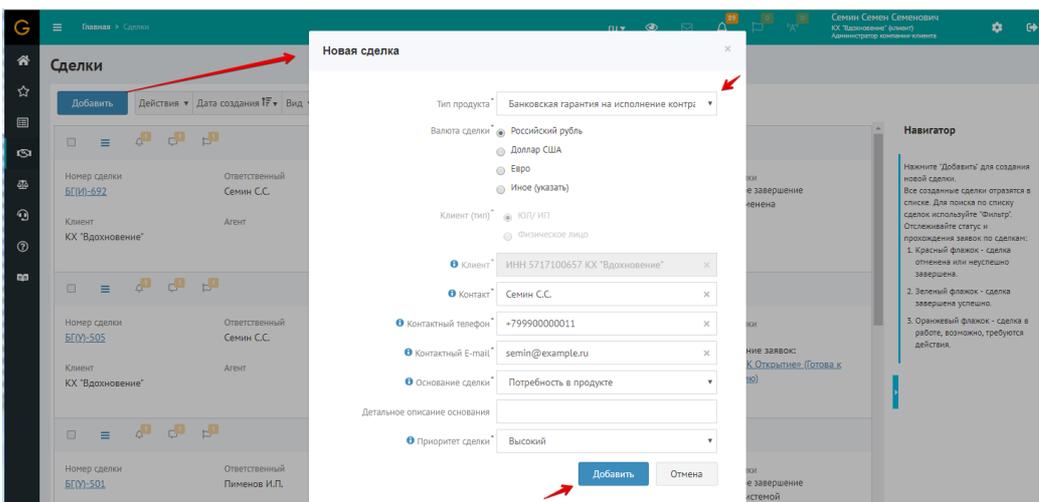
Шаг 19. Создание сделки для проверки настроек продуктовых полей и просмотра предложений.

Нажмите, чтобы посмотреть/скрыть пример



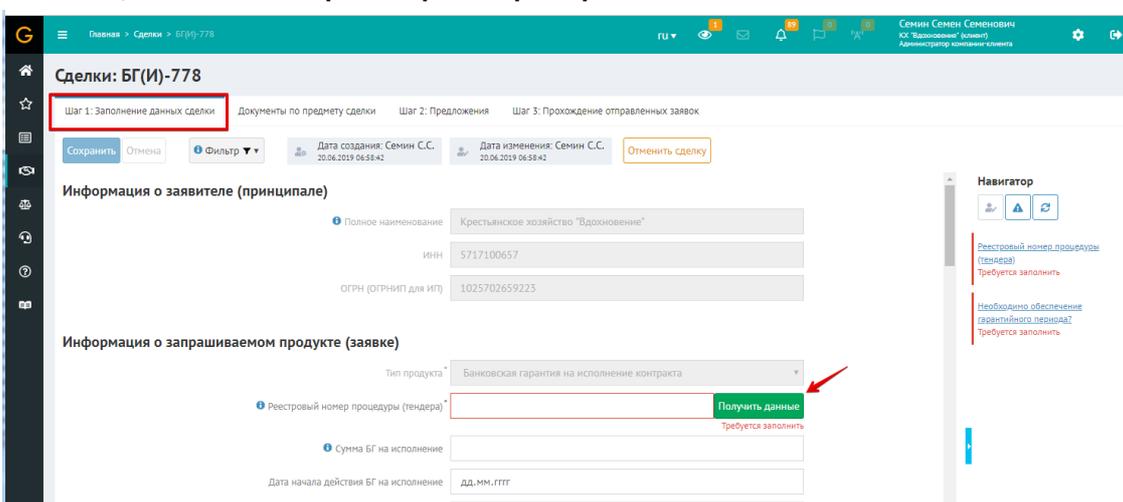
Шаг 20. Ввод минимально необходимых данных для создания сделки. Должен быть выбран тип продукта, соответствующий тестируемому продукту.

Нажмите, чтобы посмотреть/скрыть пример



Шаг 21. Заполнение данных о сделке

Нажмите, чтобы посмотреть/скрыть пример

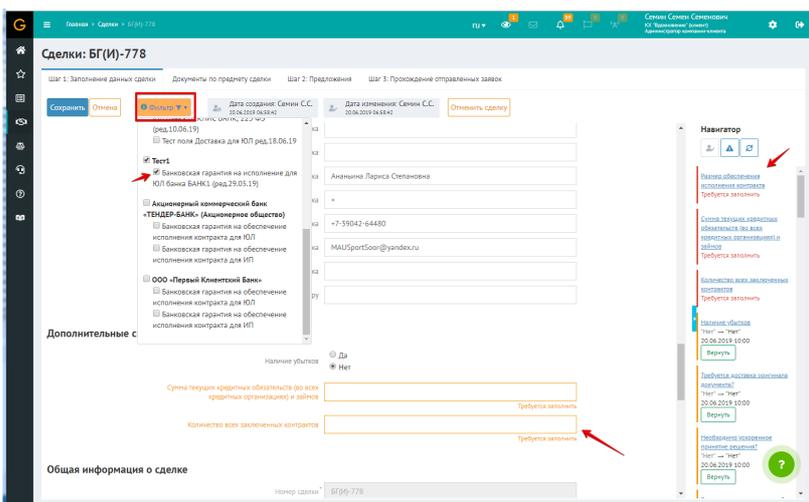


Шаг 22. Проверка настроек обязательности продуктовых полей с помощью фильтра. Для проверки обязательности заполнения продуктовых полей в сделке (в конфигурационном файле продукта это блок **productFields.required**) вы должны очистить эти поля. Система в ответ должна подсветить их, показать в панели Категории счетчик ошибок, а в панели Навигатор ошибки типа "Требуется заполнить".

ВНИМАНИЕ!

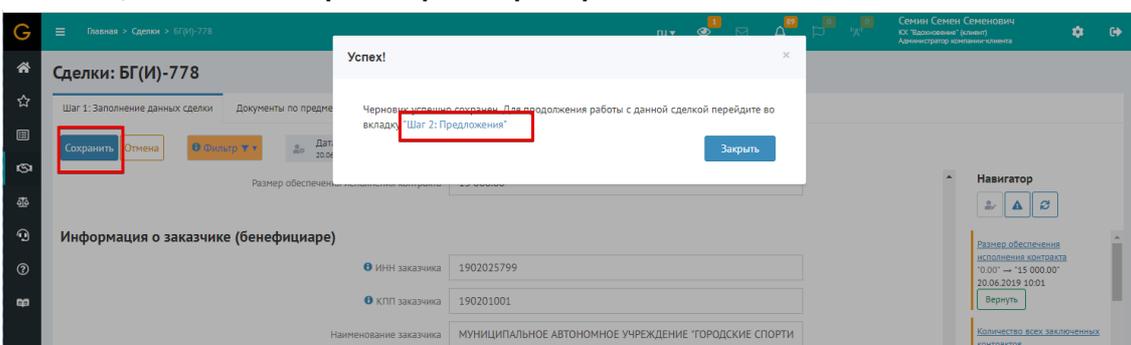
Если вы обнаружили ошибки в настройке обязательности, то требуется исправить конфигурационный файл продукта, перезагрузить его (см. шаги 29-30). Затем снова вернуться в клиентский кабинет для тестирования исправленных настроек.

Нажмите, чтобы посмотреть/скрыть пример



Шаг 23. Переход к просмотру предложений

Нажмите, чтобы посмотреть/скрыть пример



Шаг 24. Просмотр предложения, проверка счетчиков. Счетчики предназначены для информирования клиентом о том, сколько еще обязательных элементов нужно заполнить, чтобы получить возможность отправить заявку в сервис.

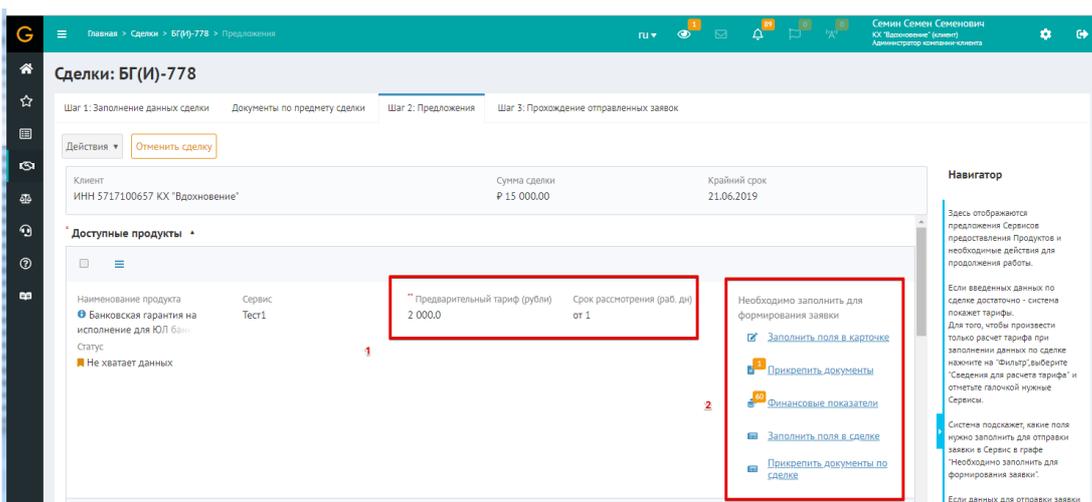
Подробнее о работе со сделками см. [Проведение сделок](#)

ВНИМАНИЕ!

Для проверки правильности расчета предварительного тарифа вы можете оперативно менять значения полей сделки на вкладке "Шаг 1: Заполнение данных сделки", которые влияют на выбор тарифа. Сохранять изменения и проверять на вкладке "Шаг 2: Предложения" значение "Предварительный тариф". Поле "Срок рассмотрения (раб. дней)" равно значению поля **workTerm** из конфигурационного файла продукта.

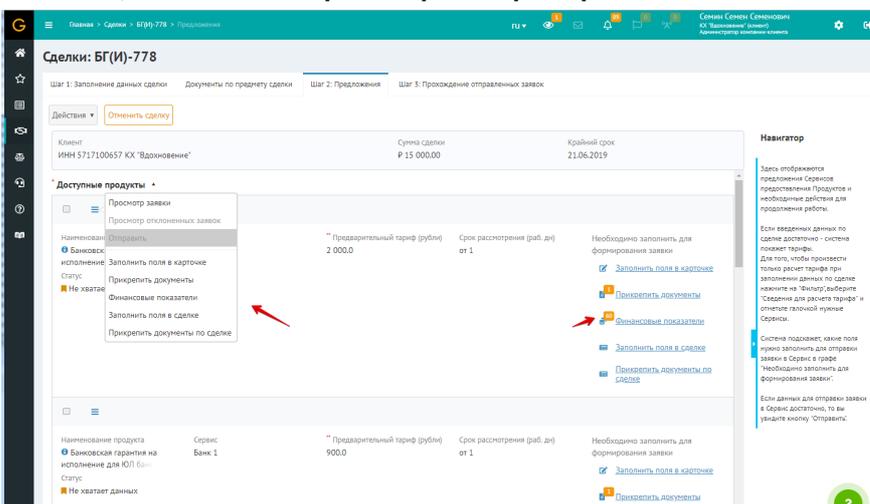
Если вы обнаружили ошибки в настройке тарифа, то требуется исправить конфигурационный файл тарифа, перезагрузить его (см. шаги 6-9). Затем снова вернуться в клиентский кабинет для тестирования исправленных настроек.

Нажмите, чтобы посмотреть/скрыть пример



Шаг 25. Для дозаполнения обязательных параметров можно нажимать или на счетчики, или использовать контекстное меню.

Нажмите, чтобы посмотреть/скрыть пример



Шаг 26. Пример срабатывания пакета документов, имеющего правило применимости.

Нажмите, чтобы посмотреть/скрыть пример

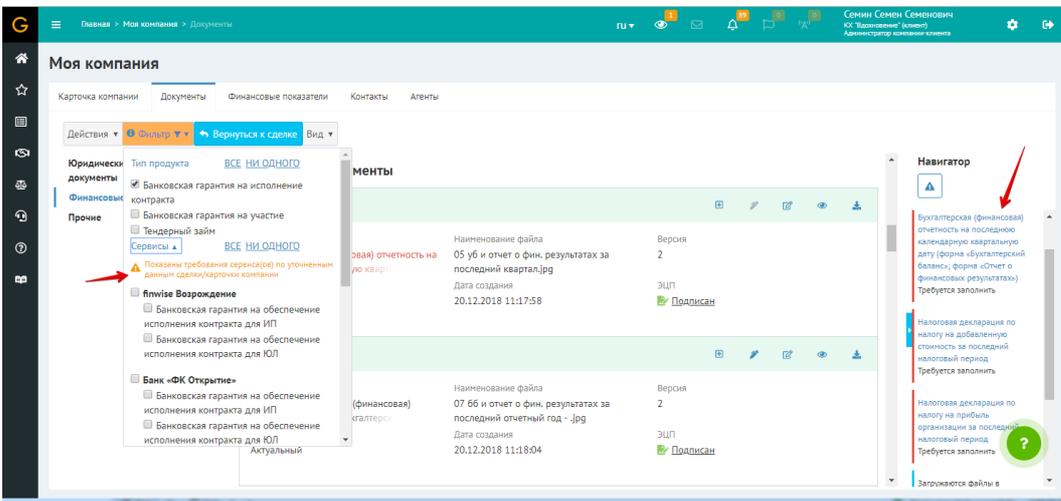
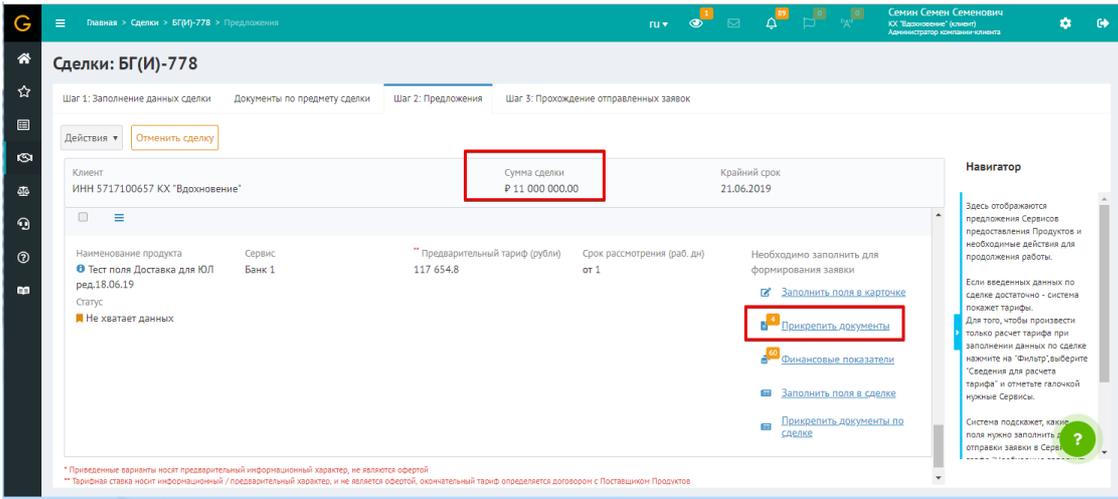
В примере настроек продукта (см. [II.1. Создание продукта в виде конфигурационного файла yam1](#)) есть пакет документов с правилом применимости, который показан ниже.

```

- #docsPackage 3
  #name varchar(500) NOT NULL
  #Указать полное наименование пакета документов, включая наименование
  продукта и сервиса для полноты понимания
  name: "Пакет БГ(И), Банк БАНК1: ЮЛ на ОСНО, сумма БГ на исполнение >
  10 млн"
  #shortname varchar(50) NOT NULL
  #Указать краткое наименование пакета документов, чтобы по нему понять
  условия применимости пакета документов
  shortName: "ЮЛ на ОСНО, сумма > 10 млн"
  #description varchar(1000) NULL, описание пакета документов, которое
  может выводиться как подсказка в системе
  description: NULL
  #полный перечень типов документов см. в docs_types
  #списком перечисляются мнемокоды типов обязательных документов в
  пакете
  #полный перечень см. в справочнике docs_types
  docsTypes:
  - "TAX_DECL_VALUE_ADDED_TAX_LAST_TAX_PERIOD"
  - "TAX_DECL_ORGANIZATION_INCOME_TAX_LAST_TAX_PERIOD"
  #правило применимости пакета документов
  applicabilityRule:
  #name varchar(150) NOT NULL
  #наименование правила применимости
  name: "ЮЛ с налогообложением = ОСНО И сумма БГ на исполнение больше
  10 млн"
  #type varchar(50) NOT NULL, по умолчанию значение = "script"
  type: "script"
  #script character varying(4000) NOT NULL
  #Скрипт правила в виде функции
  isAvailableForDealApplicationData(dealApplicationData), возвращающей TRUE,
  FALSE или NULL.
  #Если функция возвращает true, то считается, что пакет документов
  применим и при заполнении заявки у клиента будут запрошены документы в
  составе пакета
  #примеры готовых скриптов см. в документации - статья "Правила
  применимости: productApplicabilityRule, applicabilityRule"
  script: "function
  isAvailableForDealApplicationData(dealApplicationData) {
      var companyClass =
  java.lang.Class.forName('com.keyintegrity.shb.company.query.dto.CompanyDto
  ');
      var clientCompany =
  dealApplicationData.fetchFields[companyClass][dealApplicationData.result.c
  lient.id];
      if (clientCompany == null) {
          return null;
      }
      var taxSystemClass =
  java.lang.Class.forName('com.keyintegrity.shb.company.query.dto.catalog.Ta
  xSystemCatalogDto');
      var taxSystemDto =
  dealApplicationData.fetchFields[taxSystemClass][clientCompany.taxSystem];
      if (clientCompany.legalType == null || taxSystemDto ==
  null || dealApplicationData.result.productDealState.amount == null) {
          return null;
      }
      var taxSystems = ['OSNO'];
      return taxSystems.indexOf(taxSystemDto.code) != -1 &&
  clientCompany.legalType.id == 'ORGANIZATION' &&
  dealApplicationData.result.productDealState.amount > 10000000;
  }"

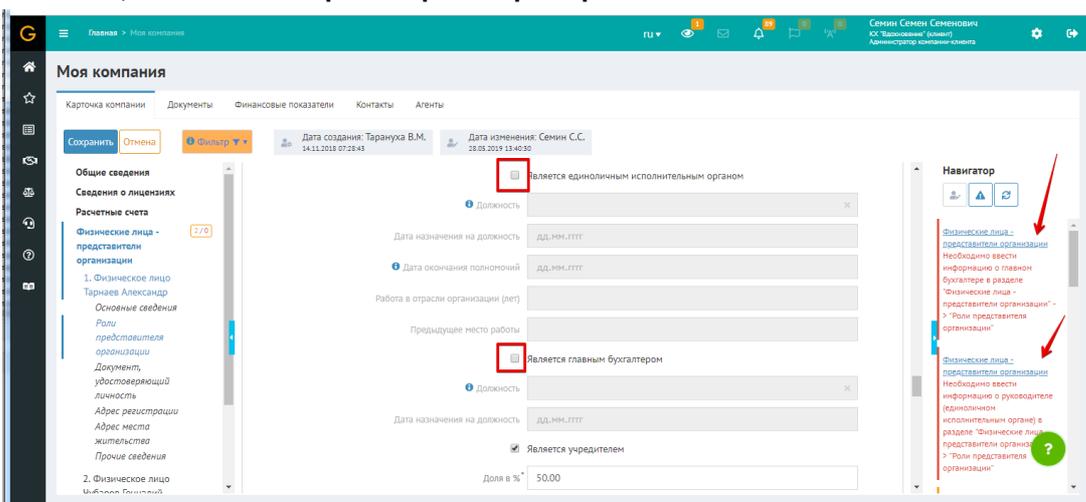
```

В примере следки исправлена сумма БГ на 11 млн. и сохранены изменения. В результате, сработало правило применимости показанного выше пакета документов.



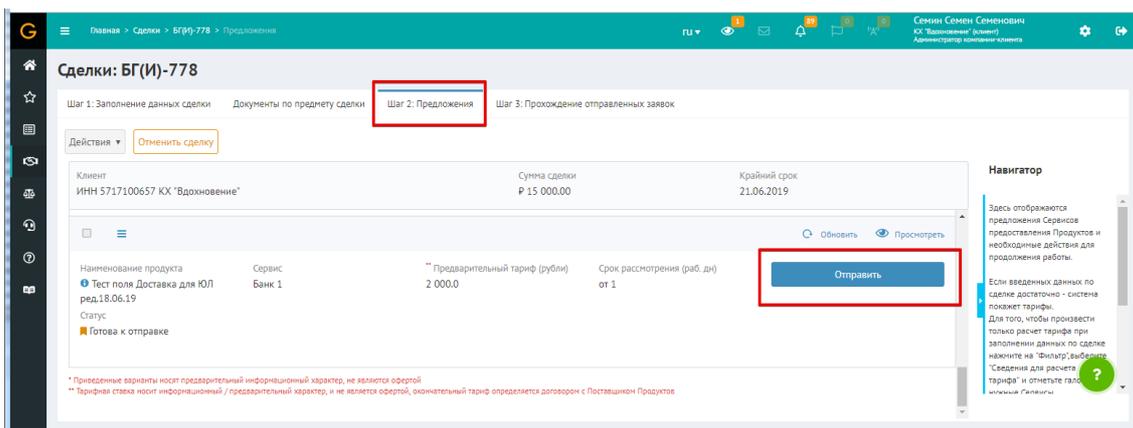
Шаг 27. Пример срабатывания валидаторов.

Нажмите, чтобы посмотреть/скрыть пример



Шаг 28. Пример успешного заполнения обязательных данных, как это видит Клиент.

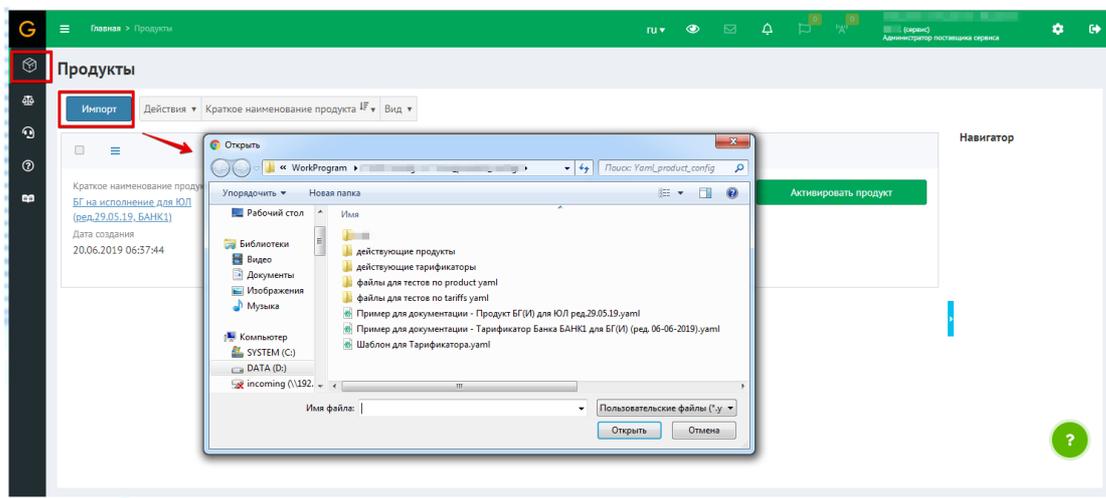
Нажмите, чтобы посмотреть/скрыть пример



Шаг 29. Загрузка исправленного конфигурационного файла продукта и/или тарификатора.

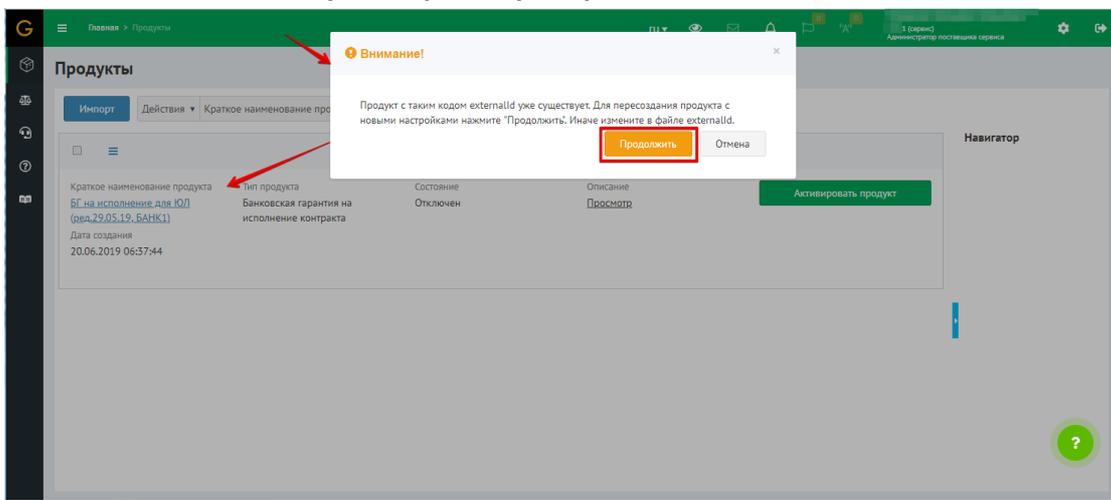
Если были обнаружены ошибки в настройке продукта или тарифа, то требуется исправить соответствующий конфигурационный файл, перезагрузить его. Затем снова вернуться в клиентский кабинет для тестирования исправленных настроек.

Нажмите, чтобы посмотреть/скрыть пример



Шаг 30. Предупреждение о перезаписи продукта, если он перезаписывается после исправления.

Нажмите, чтобы посмотреть/скрыть пример



5.7 III. Получение доступа в личный кабинет поставщика сервиса действующей системы Goodfin

Регистрация и первичная настройка выполняется сотрудниками владельца системы, данные для входа в кабинет передается представителю поставщика.

5.8 IV. Импорт подготовленных продуктов в действующую систему Goodfin

ВНИМАНИЕ!

Действующая система Goodfin не предназначена для отладки продуктов. Конфигурационные файлы должны быть подготовлены и проверены на базе тестового окружения (см. [II.3. Тестирование и доработка банковского продукта](#)).

Для импорта готовых, протестированных продуктов выполните в действующей Системе шаги 2-11 по аналогии с описанием [II.3. Тестирование и доработка банковского продукта](#).

ПРИМЕЧАНИЕ!

Если позднее у вас изменятся тарифы, вы можете актуализировать конфигурационный файл тарификатора, протестировать тарификатор согласно инструкции [II.3. Тестирование и доработка банковского продукта](#). Затем обновить тарификатор действующего продукта без перезаписи самого продукта.

5.9 Справочники, используемые при настройке уaml продукта и тарификатора

Документы, удостоверяющие личность: identity doc types	Отчётные периоды: period types	Поля в заявках по продуктам: product_types_fields	Поля карточки компании: company_fields
Поля финансовых показателей: finance fields	Правила применимости: productApplicabilityRule, applicabilityRule	Прикрепляемые документы: docs types	Типы валидаторов: validators
Типы продуктов: product types	Типы тарифов: tariff cell types	Федеральные законы: federallaw types	Формы бухгалтерской и финансовой отчетности: finance_forms
Формы предпринимательской деятельности: legal_types			

5.10 Справочники, используемые в отправляемых заявках

Документы, удостоверяющие личность: identity_doc_types	Поля финансовых показателей: finance_fields	Поставщики сервиса: services	Прикрепляемые документы: docs_types
Способы размещения заказа: placingway_types	Типы прикрепляемых файлов: file_types	Типы продуктов: product_types	Типы расчётных счетов: types_banking_details
Федеральные законы: federallaw_types	Формы бухгалтерской и финансовой отчетности: finance_forms	Формы предпринимательской деятельности: legal_types	Электронные торговые площадки: etp_types

6 Для поставщиков лидов в систему Goodfin

6.1 Оглавление

6.2 Введение и краткое описание стадий работ

Настоящий документ представляет собой руководство для разработчиков со стороны банковского сервиса/CRM - **поставщика лидов**, ответственных за подготовку банковской системы/CRM к интеграции с системой Goodfin.

6.2.1 Доступ к документации с описанием типов и методов API

Базовое описание типов AddLead, CommonResponse и эндпоинта /ad/v1/external см. в <https://sandbox.goodfin.ru/docs/shb-open-api/v1/index.html>.

ПРИМЕЧАНИЕ

В песочнице (<https://sandbox.goodfin.ru>) доступны методы только для обучения работе со сделками. По лидам методы в песочнице не реализованы, имеется только описание типов в документации.

6.2.2 Краткое описание возможностей

1. Система Goodfin предоставляет возможность банковским системам и CRM передавать лиды для их последующей обработки и отправки заявок в банки-партнеры на получение банковских продуктов. Для этого требуется согласно предлагаемой API спецификации выполнить доработку системы.
2. Требования к набору обязательных полей в составе лида минимален, поэтому поставщик лидов может выбрать для себя желаемый объем передаваемой информации в Goodfin.
3. Поставщик лидов может определить целевую аудиторию - получателей лидов в системе Goodfin:
 - a. Если в системе поставщика лидов добавить возможность клиенту отправлять свою заявку в Goodfin как лид, то лид будет отображен в личной кабинете клиента. Если клиент не зарегистрирован в системе, то ему будет отправлено приглашение на регистрацию в Goodfin, после выполнения которой клиент сможет обработать свой лид.
 - b. Если в системе поставщика лидов добавить возможность агенту отправлять карточку клиента/клиентов в Goodfin как лиды, то они будут доступны в личном кабинете агента на Goodfin.
 - c. Если инициатором отправки лидов будет сама система - поставщик лидов, то есть возможность указать правило обработки лида в Goodfin, а именно

перечислить те банки, в которые могут быть отправлены заявки в ходе обработки лидов в системе Goodfin.

4. Поставщики лидов смогут получать аналитику по результатам обработки лидов в Goodfin (пока не реализовано, идет сбор требований).

6.2.3 Краткое описание стадий работ для интеграции с действующей системой Goodfin

I. Для начала интеграционного взаимодействия поставщика лидов с системой Goodfin проводятся работы силами разработчиков поставщика лидов по поддержке API, предоставленного системой Goodfin. См. в API документации описание эндпоинта `/ad/v1/external/openapi/in`, а именно тип `AddLead` для подготовки и отправки лидов, тип `CommonResponse` для получения асинхронного ответа от Goodfin по результату получения лида с "SUCCESS" или "ERROR"). Также примеры сформированных лидов для передачи в Goodfin показаны в [Примеры структуры передаваемого лида](#).

Если требуется обрабатывать изменение статусов заявок, созданных по лиду, то см. тип `LeadDealApplicationStatusChanged` в <https://sandbox.goodfin.ru/docs/bl-open-api/v1/index.html> и пример сообщения ниже:

Нажмите, чтобы посмотреть пример сообщения

```
{
  "payloadType": "LeadDealApplicationStatusChanged",
  "originator": "Shb",
  "receiver": "test-lp1.io",
  "msgId": "8d778182-5067-4419-8207-a955dd1e111a",
  "msgDateTime": "2020-04-20T08:20:32.749514",
  "payload": {
    "lead": {
      "id": "cd08bc21-9b0c-4f84-b9d5-c9068ea989b3",
      "number": "10106",
      "deal": {
        "id": "46f3084a-cc99-4b1f-898b-4b31356d63ff",
        "createDateTime": "2020-04-17T11:53:28.031809",
        "dealApplication": {
          "id": "fc13db5f-8606-454c-8eb2-9fa84cc9cf03",
          "number": null,
          "createDateTime": "2020-04-17T11:53:28.031809",
          "serviceSysVal": "client2-shb.srvtests.com",
          "status": "REJECTED_BY_SERVICE",
          "serviceSysStatus": "22",
          "statusDateTime": "2020-04-20T08:20:32.664338"
        }
      }
    },
    "responsibles": [{"email": "agent@agent.ru"}]
  }
}
```

II. Далее для начала взаимодействия поставщику лидов передается логин и секрет. Эти данные должны использоваться поставщиком лидов для получения токена, чтобы поставщик лидов как система мог авторизоваться перед Goodfin как системой.

В ари документации можно посмотреть метод [/api/v1/oauth/token](#). Подробнее см. стандарт [jwt.io](#) по правилам получения и "времени жизни" токена.

III. Далее, если поставщик лидов будет передавать в Goodfin документы клиента в составе лида, то требуется предоставить системе Goodfin настройки, с помощью которых система Goodfin будет авторизовываться перед хранилищем документов поставщика лидов, чтобы иметь возможность скачать документы (см. в API описание эндпоинта [/leadprovider/documentdownloadsettings/update](#)). Ссылка для скачивания документа(ов) или ссылка для запроса конечного URL для скачивания передается в составе лида (см. в API документации типы Lead и LeadDocument).

IV. Отчет по результатам обработки лидов (частота, формат, объемы данных, метод передачи) согласуется с поставщиком лидов дополнительно.

6.2.4 Краткое описание стадий работ для предварительного тестирования интеграции в песочнице

Вы можете самостоятельно проверить, как будет выглядеть ваши лиды в системе Goodfin. Для этого вы можете воспользоваться тестовым окружением песочницы <https://sandbox.goodfin.ru/>.

Вы можете импортировать проект в Postman для тестирования:
окружение: [sandbox-leads.postman_environment.json](#)
коллекция: [SANDBOX-LEADS.postman_collection.json](#)

Порядок действий

1. Зарегистрируйте тестового поставщика лидов в песочнице с помощью метода [/api/v1/sandbox/registerleadprovider](#). **Обратите внимание:**
 - a. В теле передается пара ИНН/ОГРН (для генерации можно использовать ресурс <http://mellarius.ru/random-inn>), на основании которой в песочнице создается тестовый агент.
 - b. При отправке лидов типа agentLead вы должны использовать эту пару ИНН/ОГРН в теле передаваемого лида, а именно в объекте "agent". Так система узнает, какому агенту адресован лид.
 - c. **Метод выполнить один раз** и запомнить значения полученной пары ключей "serviceSystemName" и "password".
 - d. Значение ключа "serviceSystemName" с добавленным префиксом **сс_** (англ.) используйте как логин для входа в песочницу и проверки отправленных лидов. Например, в ответе вы получили {"serviceSystemName": "J91nDdRWAZ", "password": "RPTxt3jwsK"}, тогда для входа в песочницу и просмотра со стороны тестового агента в качестве логина использовать **сс_**J91nDdRWAZ, в качестве пароля RPTxt3jwsK.
 - e. Также значения полученной пары ключей "serviceSystemName" и "password" используются при генерации токена в методе [/api/v1/oauth/token](#).
2. Для получения/обновление токена используйте метод [/api/v1/oauth/token](#).
3. Чтобы получать от Goodfin сообщения об успешном/неуспешном приеме лида, смену статусов по заявке, созданной на основании лида, используйте

метод `/ad/external/integrationendpoint/update`, в котором сообщите ваш эндпоинт, куда Goodfin будет слать ответные сообщения. **Обратите внимание:**

- a. Эндпоинт должен быть виден "снаружи", чтобы при тестировании можно было получать сообщения от песочницы Goodfin по результату приема лидов.
4. Если в составе лидов отправляется информация о прикрепленных документах, то используйте метод `/api/v1/leadprovider/documentdownloadsettings/update` для передачи информации по правилам аутентификации для скачивания документов.
5. Для отправки лидов используйте метод `/ad/v1/external/openapi/in` с `"payloadType": "AddLead"`. **Обратите внимание:**
 - a. В объекте "lead" ключ "id": UUID должен быть уникальным.
 - b. При попытке отправить два лида с одним и тем же id все, кроме первого полученного, будут игнорироваться.
 - c. В качестве значения ключа "originator" установить значение ключа "serviceSystemName", полученное при выполнении метода `/api/v1/sandbox/registerleadprovider`.
 - d. В качестве значения ключа "receiver" всегда устанавливайте "Shb".
 - e. Если отправляете лид типа `agentLead`, то в объекте "agent" укажите пару ИНН/ОГРН, которую указывали при вызове метода `/api/v1/sandbox/registerleadprovider`. **Примечание:** при интеграции с действующей системой Goodfin вы должны будете передавать реальные ИНН/ОГРН того агента, которому адресован агентский лид.
6. Проверить получение лида в песочнице <https://sandbox.goodfin.ru/>. Для этого войти с помощью логина/пароля (см. пункт 1.d). О работе с лидами можно почитать в статье [Как обработать лид](#).

6.3 Примеры структуры передаваемого лида

Подробное описание всех полей в составе типа Lead см. в <https://sandbox.goodfin.ru/docs/shb-open-api/v1/index.html>.

6.3.1 Пример структуры передаваемого лида по типу продукта "Банковская гарантия на исполнение", если инициатором отправки был агент.

Скачать: [пример AddLead агентский с почти полным наполнением данными.json](#)

Нажмите, чтобы скрыть/раскрыть пример

```

{
  "payloadType": "AddLead",
  "originator": "test-lp1.io",
  "receiver": "Shb",
  "msgId": "f4144c65-0809-4f2d-9ce1-c44634c49915",
  "msgDateTime": "2019-06-05T11:40:59.614",
  "payload": {
    "lead": {
      "id": "c3eac434-2143-4fe5-b329-a6e14c4145d5",
      "number": "10",
      "createDateTime": "2019-07-25T08:03:37.076",
      "currencyType": "RUB",
      "resolutionDate": "2019-07-26",
      "pipeline": "SERVICELEAD",
      "targetServices": [
        "bin.itfinance.io"
      ],
      "client": {
        "clientType": "COMPANY",
        "id": "8798b3f4-b4cc-4ed1-a7c2-23b8c8403ce1",
        "fullName": "Общество с ограниченной ответственностью
\"Сверчок\"",
        "shortName": "ООО \"Сверчок\"",
        "englishName": "ООО Sverchok",
        "okopf": "12300",
        "inn": "2348033111",
        "kpp": "234801001",
        "innDate": "2012-03-22",
        "innPlace": "Владимир УВД",
        "ogrn": "1122348000103",
        "regNumber": "15 45871",
        "regDate": "2012-01-24",
        "regDocDate": "2012-01-24",
        "regOrg": "Инспекция Федеральной Налоговой Службы по Северскому
Району Краснодарского Края",
        "regPlace": "КРАСНОДАРСКИЙ КРАЙ",
        "okpo": "37111027",
        "okato": "03243806001",
        "oktmo": "03643406101",
        "okoqu": "4210014",
        "okfs": "16",
        "baseOkved": "011",
        "okveds": [
          "105",
          "494"
        ],
        "regCapital": 10000.00,
        "paidCapital": 9000.00,
        "taxSystem": "OSNO",
        "pfr": "033051018439",
        "fss": "231748217023171",
        "staff": 55,
        "orgType": "COM",
        "topType": "100_PERCENT",
        "addresses": [
          {
            "addressType": "LEGAL_ADDRESS",
            "kladr": "7700000000070310012",
            "fias": "19b37ff4-44eb-438f-8668-1008866cf653",
            "addressString": "115035, Россия, город Москва,
набережная Космодамианская, дом 26/55, строение 6, квартира 3",
            "postCode": "115035",
            "country": "Россия",
            "region": "Москва город",
            "district": null,
            "city": "город Москва",

```

```

    "street": "набережная Космодамианская",
    "house": "дом 26/55",
    "building": "строение 6",
    "flat": "квартира 3",
    "isOwned": false,
    "rentOrOwnerContract": {
      "date": null,
      "serial": null,
      "num": null,
      "owner": null,
      "endDate": null
    }
  },
  {
    "addressType": "FACT_ADDRESS",
    "kladr": "7700000000070310012",
    "fias": "19b37ff4-44eb-438f-8668-1008866cf653",
    "addressString": "115035, Россия, город Москва,
набережная Космодамианская, дом 26/55, строение 6, квартира 3",
    "postCode": "115035",
    "country": "Россия",
    "region": "Москва город",
    "district": null,
    "city": "город Москва",
    "street": "набережная Космодамианская",
    "house": "дом 26/55",
    "building": "строение 6",
    "flat": "квартира 3",
    "isOwned": false,
    "rentOrOwnerContract": {
      "date": null,
      "serial": null,
      "num": null,
      "owner": null,
      "endDate": null
    }
  },
  {
    "addressType": "CORR_ADDRESS",
    "kladr": "7700000000070310012",
    "fias": "19b37ff4-44eb-438f-8668-1008866cf653",
    "addressString": "115035, Россия, город Москва,
набережная Космодамианская, дом 26/55, строение 6, квартира 3",
    "postCode": "115035",
    "country": "Россия",
    "region": "Москва город",
    "district": null,
    "city": "город Москва",
    "street": "набережная Космодамианская",
    "house": "дом 26/55",
    "building": "строение 6",
    "flat": "квартира 3",
    "isOwned": false,
    "rentOrOwnerContract": {
      "date": null,
      "serial": null,
      "num": null,
      "owner": null,
      "endDate": null
    }
  }
],
"legalFounders": [
  {
    "name": "ООО Восход",
    "percent": 5.35,

```

```

        "inn": "3968760398",
        "kpp": "054943877",
        "ogrn": "1036579367538",
        "address": "г. Москва, ул. Московская 1, офис 35"
    }
],
"persons": [
    {
        "lastName": "Василькова",
        "firstName": "Людмила",
        "middleName": "Яковлевна",
        "docType": "PASSPORT",
        "docCodePod": "111-111",
        "docDate": "2015-01-01",
        "docNumber": "111111",
        "docSeries": "1111",
        "docWho": "КРАСНОДАРСКИЙ КРАЙ",
        "docOther": null,
        "docEndDate": null,
        "inn": "323901319790",
        "snils": "00284990751",
        "birthdayPlace": "КРАСНОДАРСКИЙ КРАЙ",
        "birthday": "1970-03-01",
        "sex": "F",
        "mainOksm": "RUS",
        "otherOksm": "USA",
        "addresses": [
            {
                "addressType": "FACT_ADDRESS",
                "kladr": null,
                "fias": null,
                "postCode": "353251",
                "country": null,
                "region": "КРАСНОДАРСКИЙ КРАЙ",
                "district": "РАЙОН СЕВЕРСКИЙ",
                "city": "СТАНИЦА КАЛУЖСКАЯ",
                "street": "УЛИЦА СОВЕТСКАЯ",
                "house": "дом 3",
                "building": "корпус 5",
                "flat": "кв 456",
                "addressString": "353251, КРАСНОДАРСКИЙ КРАЙ, РАЙОН
СЕВЕРСКИЙ, СТАНИЦА КАЛУЖСКАЯ, УЛИЦА СОВЕТСКАЯ, дом 3, корпус 5, кв 456"
            },
            {
                "addressType": "REG_ADDRESS",
                "kladr": null,
                "fias": null,
                "postCode": "353251",
                "country": null,
                "region": "КРАСНОДАРСКИЙ КРАЙ",
                "district": "РАЙОН СЕВЕРСКИЙ",
                "city": "СТАНИЦА КАЛУЖСКАЯ",
                "street": "УЛИЦА СОВЕТСКАЯ",
                "house": "дом 3",
                "building": "корпус 5",
                "flat": "кв 456",
                "addressString": "353251, КРАСНОДАРСКИЙ КРАЙ, РАЙОН
СЕВЕРСКИЙ, СТАНИЦА КАЛУЖСКАЯ, УЛИЦА СОВЕТСКАЯ, дом 3, корпус 5, кв 456"
            }
        ]
    },
    {
        "isPdl": true,
        "pdlType": "RF_RELATIVE",
        "moneyPdl": [
            "WAGE",
            "INSURANCE",
            "BUSINESS"
        ]
    }
]

```

```

],
"otherMoneyPdl":null,
"personRoles":[
  {
    "roleType":"BENEFICIAR",
    "percent":94.65,
    "date":"2012-01-24"
  },
  {
    "roleType":"FOUNDER",
    "percent":94.65,
    "date":"2012-01-24"
  },
  {
    "roleType":"DIRECTOR",
    "fullPostName":"Генеральный директор",
    "postDate":"2017-01-01",
    "endDate":"2023-01-01",
    "workYears":15,
    "workPrev":"ООО Каштан"
  },
  {
    "roleType":"BOOKER",
    "fullPostName":"Главный бухгалтер",
    "postDate":"2016-01-01",
    "endDate":null,
    "workYears":null,
    "workPrev":null
  }
]
},
{
  "lastName":"Кузьмина",
  "firstName":"Ирина",
  "middleName":"Матвеевна",
  "docType":"PASSPORT",
  "docCodePod":"666-666",
  "docDate":"2015-01-01",
  "docNumber":"222222",
  "docSeries":"2222",
  "docWho":"Псковское УВД",
  "docOther":null,
  "docEndDate":null,
  "inn":"913105511344",
  "snils":"79517289059",
  "birthdayPlace":"Псков",
  "birthday":"2000-12-31",
  "sex":"F",
  "mainOksm":"RUS",
  "otherOksm":"USA",
  "addresses":[
    {
      "addressType":"REG_ADDRESS",
      "kladr":null,
      "fias":null,
      "postCode":"127253",
      "country":"Россия",
      "region":"Москва город",
      "district":null,
      "city":"город Москва",
      "street":"улица Псковская",
      "house":"дом 12",
      "building":null,
      "flat":null,
      "addressString":"127253, Россия, город Москва,
улица Псковская, дом 12"
    }
  ]
}

```

```

    },
    {
      "addressType": "FACT_ADDRESS",
      "kladr": null,
      "fias": null,
      "postCode": "127253",
      "country": "Россия",
      "region": "Москва город",
      "district": null,
      "city": "город Москва",
      "street": "улица Псковская",
      "house": "дом 12",
      "building": null,
      "flat": null,
      "addressString": "127253, Россия, город Москва,
улица Псковская, дом 12"
    }
  ],
  "personRoles": [
    {
      "roleType": "TRUSTEE",
      "startDate": "2018-12-12",
      "endDate": "2019-12-12",
      "isEdsSigner": true,
      "docNumber": "1522222",
      "controlFirstName": "Людмила",
      "controlLastName": "Василькова",
      "controlMiddleName": "Яковлевна",
      "controlPersonRoleType": "DIRECTOR"
    }
  ]
},
"bankingDetails": [
  {
    "bik": "044525225",
    "paymentAccount": "40702810500020107079",
    "isMainAccount": true,
    "bankName": "ПАО СБЕРБАНК",
    "shortBankName": "ПАО СБЕРБАНК",
    "legalAddress": null,
    "actualAddress": null,
    "corrAccount": "30101810400000000225",
    "kpp": null,
    "inn": null,
    "phones": null,
    "faxes": null,
    "website": null,
    "emails": null,
    "typeBankingDetails": "ACCOUNT_RUB",
    "ufk": null,
    "kbc": null
  }
],
"companyLicenses": [
  {
    "number": "№ ЛО-86-01-003413",
    "date": "2019-09-10",
    "activity": "Медицинская деятельность",
    "who": "Служба по контролю и надзору в сфере
здравоохранения Ханты-Мансийского автономного округа-Югры",
    "listActivities": "100. При оказании первичной, в том
числе доврачебной, врачебной и специализированной, медико-санитарной
помощи организуются и выполняются следующие работы (услуги):, 100.1. при
оказании первичной доврачебной медико-санитарной помощи в амбулаторных
условиях по:, 100.1.12. лечебному делу, 100.1.24. сестринскому делу",
  }
]

```

```

        "startDate": "2019-09-10",
        "endDate": "2037-06-27",
        "place": "628606, Ханты-Мансийский автономный округ -
Югра, г. Нижневартовск, ул. Индустриальная, д. 51. Районно-диспетчерский
пункт"
    }
  ],
  },
  "agent": {
    "inn": "5717100657",
    "ogrn": "1025702659223"
  },
  "clientContactFirstName": "Людмила",
  "clientContactLastName": "Василькова",
  "clientContactMiddleName": "Яковлевна",
  "clientContactPhone": "+79008887799",
  "clientContactEmail": "sverchok@example.com",
  "product": {
    "productType": "BG",
    "amount": 9092169,
    "actualContractPrice": 60614460.00,
    "startDate": "2019-07-26",
    "endDate": "2020-07-26",
    "isWarrantyPeriod": false,
    "isIndisputableWriteOff": false,
    "isPrepayment": true,
    "prepaymentAmount": 350000.00,
    "tender": {
      "registryNumber": "0134200000119002028",
      "name": "Строительство спортивного оздоровительного
комплекса в селе Казачинское, Казачинско-Ленского района, Иркутской
области",
      "federalLaw": "44FL",

      "url": "http://zakupki.gov.ru/epz/order/notice/ok44/view/common-
info.html?regNumber=0134200000119002028",
      "placementType": "E_AUCTION",
      "publishDateTime": "2019-06-20T16:13:20.339",
      "isMultiCustomer": false,
      "isMultiLot": false,
      "lotNumber": "1",
      "lotName": "Строительство спортивного оздоровительного
комплекса в селе Казачинское, Казачинско-Ленского района, Иркутской
области",
      "initialContractPrice": 60614460,
      "currencyType": "RUB",
      "enforceAmount": 9092169
    },
  },
  "customer": {
    "inn": "3828000051",
    "kpp": "382801001",
    "name": "АДМИНИСТРАЦИЯ КАЗАЧИНСКО-ЛЕНСКОГО МУНИЦИПАЛЬНОГО
РАЙОНА",
    "ogrn": "1023802527583",
    "region": "Иркутская обл",
    "oktmo": "44979",
    "postAddress": "Российская Федерация, 666511, Иркутская обл,
Казачинско-Ленский р-н, Казачинское с, УЛ ЛЕНИНА, ДОМ 10",
    "factAddress": "Российская Федерация, 666511, Иркутская обл,
УЛ ЛЕНИНА, ДОМ 10",
    "bankName": "ОТДЕЛЕНИЕ ИРКУТСК (УФК по Иркутской области)",
    "bankBik": "042520001",
    "bankAccountMain": "40302810600003000007",
    "bankAccountCorr": null,
    "contactFio": "Орлова Яна Артемовна",
    "contactPhone": null,
  }
}

```

```

        "contactEmail": "y.orlova@govirk.ru",
        "regNum": "01343000273",
        "consolidatedRegNum": "25302900"
    }
},
"documents": [
    {
        "docName": "passport",
        "files": [
            {
                "id": "221",
                "fileName": "Памятка по используемым типам связей
между требованиями и пр.docx",
                "fileType": "docx",
                "url": "https://drive.google.com/open?id=0ByV41_gcqNpIWng0WHItaVE2TzQ",
                "fileSize": 53248,
                "fileHashes": [
                    {
                        "value": "abc",
                        "algorithm": "MD5"
                    }
                ],
                "fileSignatures": [
                    {
                        "createDateTime": "2018-08-16T16:57:06.41",
                        "signature": "c2lnbmF0dXJlIGNvb3RlbnQgdGhhdCBmb3IgLnNpZyBmaWxl"
                    }
                ]
            },
            {
                "id": "222",
                "fileName": "picture1.jpg",
                "fileType": "jpg",
                "url": "https://bipbap.ru/wp-content/uploads/2017/10/0_8eb56_842bba74_XL-640x400.jpg",
                "fileSize": 53248,
                "fileHashes": [
                    {
                        "value": "abc",
                        "algorithm": "MD5"
                    }
                ],
                "fileSignatures": [
                    {
                        "createDateTime": "2018-08-16T16:57:06.41",
                        "signature": "c2lnbmF0dXJlIGNvb3RlbnQgdGhhdCBmb3IgLnNpZyBmaWxl"
                    }
                ]
            }
        ]
    },
    {
        "docName": "License",
        "files": [
            {
                "id": "223",
                "fileName": "picture1.jpg",
                "fileType": "jpg",
                "url": "https://bipbap.ru/wp-content/uploads/2017/10/0_8eb56_842bba74_XL-640x400.jpg",
                "fileSize": 53248,
                "fileHashes": [
                    {

```

```

        "value": "abc",
        "algorithm": "MD5"
    }
],
"fileSignatures": [
    {
        "createDateTime": "2018-08-16T16:57:06.41",
"signature": "c2lnbmF0dXJlIGVbnRlbnQgdGhhdCBmb3IgLnNpZyBmaWxl"
    }
]
}
]
},
"financeFormValues": [
    {
        "formSetKndCode": "0710099",
        "formOKUD": "0710001",
        "monetaryUnit": "THOUSAND_RUB",
        "periods": [
            "2019-03-31",
            "2018-12-31",
            "2017-12-31"
        ],
        "formValues": [
            {
                "code": "1100",
                "values": [
                    -1.000,
                    0.000,
                    0.000
                ]
            },
            {
                "code": "1150",
                "values": [
                    0.000,
                    0.000,
                    0.000
                ]
            },
            {
                "code": "1430",
                "values": [
                    0.000,
                    -10.000,
                    0.000
                ]
            },
            {
                "code": "1320",
                "values": [
                    0.000,
                    0.000,
                    0.000
                ]
            },
            {
                "code": "1420",
                "values": [
                    0.000,
                    0.000,
                    0.000
                ]
            }
        ]
    },

```

```

    {
      "code": "1350",
      "values": [
        0.000,
        0.000,
        0.000
      ]
    },
    {
      "code": "1550",
      "values": [
        0.000,
        0.000,
        0.000
      ]
    },
    {
      "code": "1190",
      "values": [
        0.000,
        0.000,
        0.000
      ]
    },
    {
      "code": "1110",
      "values": [
        0.000,
        0.000,
        0.000
      ]
    },
    {
      "code": "1510",
      "values": [
        0.000,
        0.000,
        0.000
      ]
    },
    {
      "code": "1300",
      "values": [
        0.000,
        0.000,
        0.000
      ]
    },
    {
      "code": "1230",
      "values": [
        0.000,
        null,
        0.000
      ]
    }
  ],
  {
    "formSetKndCode": "0710099",
    "formOKUD": "0710002",
    "monetaryUnit": "THOUSAND_RUB",
    "periods": [
      "2019-03-31",
      "2018-12-31"
    ]
  },
  ],
  {

```



```

{
  "lead": {
    "id": "23c67b55-832b-41e9-86f3-8540a8f4069f",
    "createDateTime": "2019-07-25T08:03:37.076",
    "currencyType": "RUB",
    "pipeline": "SERVICELEAD",
    "targetServices": [
      "bin.itfinance.io"
    ],
    "client": {
      "clientType": "COMPANY",
      "id": "8798b3f4-b4cc-4ed1-a7c2-23b8c8403ce1",
      "fullName": "Общество с ограниченной ответственностью \"Сверчок\"",
      "inn": "2348033111",
      "ogrn": "1122348000103",
      "orgType": "COM",
      "addresses": [{
        "addressType": "LEGAL_ADDRESS",
        "addressString": "115035, Россия, город Москва, набережная
Космодамианская, дом 26/55, строение 6, квартира 3"
      }],
      "persons": [
        {
          "lastName": "Василькова",
          "firstName": "Людмила",
          "personRoles": [
            {
              "roleType": "BENEFICIAR",
              "percent": 94.65
            },
            {
              "roleType": "FOUNDER",
              "percent": 94.65
            },
            {
              "roleType": "DIRECTOR",
              "fullPostName": "Генеральный директор"
            },
            {
              "roleType": "BOOKER",
              "fullPostName": "Главный бухгалтер"
            }
          ]
        }
      ]
    },
    "bankingDetails": [{
      "bik": "044525225",
      "paymentAccount": "40702810500020107079",
      "isMainAccount": true
    }],
    "clientContactFirstName": "Людмила",
    "clientContactLastName": "Василькова",
    "clientContactMiddleName": "Яковлевна",
    "clientContactPhone": "+79008887799",
    "clientContactEmail": "sverchok@example.com",
    "product": {
      "productType": "BG",
      "tender": {
        "registryNumber": "0134200000119002028",
        "federalLaw": "44FL"
      }
    }
  }
}

```

6.4 Варианты требований к наполненности лидов в зависимости от целей

ОГЛАВЛЕНИЕ

- [Вариант I: Хотим передавать минимум информации, чтобы клиент максимально быстро передавался в работу менеджеру Goodfin](#)
 - [Перечень данных в составе лида](#)
- [Вариант II: Хотим передавать максимум информации, чтобы по переданному лиду без задержек формировалась заявка для отправки в банк](#)
 - [Пример требований к обязательности полей и документов на примере продукта "Банковская гарантия на исполнение" для ЮЛ банка Открытие](#)
 - [Перечень полей по продукту](#)
 - [Перечень полей с данными о клиенте](#)
 - [Перечень бухг. и финансовых показателей](#)
 - [Перечень пакетов обязательных документов](#)
 - [Перечень данных контактного лица клиента](#)
 - [Перечень данных об агенте, которому передается лид](#)

6.4.1 Вариант I: Хотим передавать минимум информации, чтобы клиент максимально быстро передавался в работу менеджеру Goodfin

Плюсы

1. Затраты ресурсов разработчиков со стороны поставщика лидов минимальны.
2. Время тестирования и вероятных ошибок минимально.
3. Вероятность доработки CRM или др. системы, которая будет передавать лиды в Goodfin минимальна.
4. Если клиент новый, то в системе Goodfin автоматически будут импортированы сведения о клиенте по ИНН и сведения о продукте по реестровому номеру процедуры (для банковских гарантий).

Минусы

1. Менеджеры агента будут вынуждены дозаполнять карточку клиента и финансовые показатели на стороне Goodfin, взаимодействовать с клиентом для получения требуемых документов в электронном виде. В открытых источниках нет информации о паспортных данных ответственных лиц и учредителей, их нужно получить и ввести для новых клиентов.
2. Время подготовки данных по клиенту для отправки первой заявки больше по сравнению с отправкой заявок по клиенту с собранными данными.

6.4.1.1 Перечень данных в составе лида

Блок данных	Наименование поля	Комментарий
Контактное лицо клиента	Фамилия	
	Имя	
	Отчество	если есть
	Телефон	если есть
	E-mail	если есть
Клиент	ИНН	
	Полное наименование клиента	
Агент		если лид передается конкретному агенту в Goodfin
	ИНН агента	
	ОГРН/ОГРНИП агента	
Продукт	Тип продукта	
	Реестровый номер процедуры	Если тип продукта связан с банковскими гарантиями

6.4.2 Вариант II: Хотим передавать максимум информации, чтобы по переданному лиду без задержек формировалась заявка для отправки в банк

Плюсы

1. Менеджеры агента быстро формируют заявку для отправки в банк.
2. Вероятность получить отказ от клиента при подписании заявки в банке минимальна, т.к. менеджер фактически не беспокоит клиента.
3. Решается проблема сбора и прикрепления документов клиента (как самая затратная по времени операция, если клиент новый).

Минусы

1. Затраты ресурсов разработчиков со стороны поставщика лидов.
2. Время тестирования увеличивается.
3. Высокая вероятность доработки CRM или др. системы с точки зрения добавления полей по клиенту.
4. Нет возможности гарантировать, что набор обязательных полей и документов окончательный, т.к. требования банков постоянно меняются. Следовательно, сохраняется вероятность дозаполнения недостающих данных на стороне

Goodfin. А также сохраняется вероятность дополнительных запросов документов во время обработки заявки банком.

6.4.2.1 Пример требований к обязательности полей и документов на примере продукта "Банковская гарантия на исполнение" для ЮЛ банка Открытие

6.4.2.1.1 Перечень полей по продукту

Наименование поля	Комментарий
Фактическая цена контракта (в валюте контракта)	
Сумма БГ (в валюте контракта)	
(Информация о гос. заказчике) ИНН	
(Информация о гос. заказчике) КПП	
(Информация о гос. заказчике) Наименование заказчика	
(Информация о гос. заказчике) ОГРН	
(Информация о гос. заказчике) Регион (копируется из тендера)	
Требуется доставка оригиналов документов?	
Дата окончания действия БГ (дд.мм.гггг)	
Наличие убытков у заявителя	Да/Нет
Контрактом предусмотрено беспорное списание?	Да/Нет
В контракте оговорен аванс?	Да/Нет
Необходимо обеспечение гарантийного периода?	Да/Нет
Сумма аванса (в валюте контракта)	Если указан аванс
Дата начала действия БГ (дд.мм.гггг)	
(Информация о лоте) Валюта контракта	По умолчанию, RUB
(Информация о лоте) Размер обеспечения исполнения контракта (в валюте контракта)	
Федеральный закон, регулирующий торги	
(Информация о лоте) Начальная (максимальная) цена контракта (в валюте контракта)	
Процедура (тендер) с несколькими заказчиками	Да/Нет
Процедура (тендер) с несколькими лотами	Да/Нет
(Информация о лоте) Предмет лота	
(Информация о лоте) Номер лота	
Предмет процедуры (тендера)	
Способ размещения заказа	

Наименование поля	Комментарий
Дата и время публикации процедуры (тендера) (дд.мм.гггг чч:мм:сс)	
Реестровый номер процедуры (тендера)	
Ссылка на процедуру (тендер)	

6.4.2.1.2 Перечень полей с данными о клиенте

Блок данных	Наименование поля	Комментарий
Общие сведения	Полное наименование организации	
	Сокращенное наименование организации	
	ИНН	
	КПП	только для ЮЛ
	ОГРН (ОГРНИП для ИП)	
	ОКОПФ (код)	
	ОКПО (код)	
	Тип организации	Коммерческая/Некоммерческая
	Дата государственной регистрации	
	Наименование регистрирующего органа	
	Место государственной регистрации	
	Система налогообложения	
	Базовый ОКВЭД (код)	
Реквизиты расч. счета	(Реквизиты расч. счета) Наименование банка	
	(Реквизиты расч. счета) БИК	
	(Реквизиты расч. счета) Расчетный счет	
Факт. адрес	(Факт. адрес) Адрес одной строкой	
	(Факт. адрес) Населенный пункт (город, село и т.п.)	
	(Факт. адрес) Дом	

Блок данных	Наименование поля	Комментарий
	(Факт. адрес) Помещение находится в собственности	
	(Факт. адрес) Индекс	
	(Факт. адрес) Область (республика, край)	
Юр. Адрес	(Юр. Адрес) Адрес одной строкой	
	(Юр. Адрес) Населенный пункт (город, село и т.п.)	
	(Юр. Адрес) Дом	
	(Юр. Адрес) Помещение находится в собственности	
	(Юр. Адрес) Индекс	
	(Юр. Адрес) Область (республика, край)	
Учредители-ЮЛ	(Учредители-ЮЛ) Адрес организации одной строкой	
	(Учредители-ЮЛ) ИНН организации	
	(Учредители-ЮЛ) КПП	
	(Учредители-ЮЛ) Полное наименование организации	
	(Учредители-ЮЛ) ОГРН	
	(Учредители-ЮЛ) Доля в уставном капитале, %	
Лицензии	(Лицензии) Вид деятельности	Лицензии указываются, если есть
	(Лицензии) Дата выдачи лицензии	
	(Лицензии) Перечень видов лицензируемой деятельности	
	(Лицензии) Номер лицензии	
	(Лицензии) Кем выдана лицензия	

Блок данных	Наименование поля	Комментарий
Физ. лица - представители ЮЛ/ИП		Это информация обо всех физ. лицах - представителях ЮЛ/ИП, которые играют роль учредителей, бенефициарных владельцев, исполнительного органа, главного бухгалтера.
	(Физ. лица - представители) Дата рождения	
	(Физ. лица - представители) Место рождения	
	(Физ. лица - представители, паспорт) Код подразделения	
	(Физ. лица - представители, паспорт) Дата выдачи	
	(Физ. лица - представители, паспорт) Номер документа	
	(Физ. лица - представители, паспорт) Серия документа	
	(Физ. лица - представители, паспорт) Документ, удостоверяющий личность	
	(Физ. лица - представители, паспорт) Кем выдан	
	(Физ. лица - представители) Имя	
	(Физ. лица - представители) ИНН	
	(Физ. лица - представители) Фамилия	
	(Физ. лица - представители) Гражданство (ОКСМ)	
	(Физ. лица - представители) Отчество	если есть

Блок данных	Наименование поля	Комментарий
	(Физ. лица - представители) адрес регистрации	
	(Физ. лица, роль - руководитель) Является единоличным исполнительным органом	
	(Физ. лица, роль - руководитель) Должность	
	(Физ. лица, роль - руководитель) Дата назначения на должность	
	(Физ. лица, роль - гл. бухг.) Является главным бухгалтером	
	(Физ. лица, роль - гл. бухг.) Должность	
	(Физ. лица, роль - учредитель) Является учредителем	
	(Физ. лица, роль - учредитель) Доля в %	
	(Физ. лица, роль - учредитель) Дата начала владения	

6.4.2.1.3 Перечень бухг. и финансовых показателей

Форма	Показатель	Комментарий
Форма 1		Предоставляется за последний отчетный период, за последний заверченный год, за год, предшествующий последнему заверченному (если есть)
	1100	
	1110	
	1120	
	1130	
	1140	
	1150	
	1160	
	1170	
	1180	

Форма	Показатель	Комментарий
	1190	
	1200	
	1210	
	1220	
	1230	
	1240	
	1250	
	1260	
	1300	
	1310	
	1320	
	1340	
	1350	
	1360	
	1370	
	1400	
	1410	
	1420	
	1430	
	1450	
	1500	
	1510	
	1520	
	1530	
	1540	
	1550	
	1600	
	1700	
Форма 2		Предоставляется за последний отчетный период, за последний завершённый год, за год, предшествующий последнему завершённому (если есть)
	2100	
	2110	

Форма	Показатель	Комментарий
	2120	
	2200	
	2210	
	2220	
	2300	
	2310	
	2320	
	2330	
	2340	
	2350	
	2400	
	2410	
	2421	
	2430	
	2450	
	2460	
	2500	
	2510	
	2520	
	2900	
	2910	

6.4.2.1.4 Перечень пакетов обязательных документов

Пакет документов (настройка в системе Goodfin)	Наименование документа
Пакет БГ(И), Банк Открытие: для всех ЮЛ	
	Устав в действующей редакции, изменения и/или дополнения в Устав (при наличии)
	Решение/протокол о назначении руководителя
	Копия паспорта руководителя
Пакет БГ(И), Банк Открытие: ВСЕ клиенты, 1 млн < сумма БГ < = 50 млн	

Пакет документов (настройка в системе Goodfin)	Наименование документа
	Годовая бухгалтерская (финансовая) отчетность (форма «Бухгалтерский баланс»; форма «Отчет о финансовых результатах»)
	Бухгалтерская (финансовая) отчетность на последнюю календарную квартальную дату (форма «Бухгалтерский баланс»; форма «Отчет о финансовых результатах»)
Пакет БГ(И), Банк Открытие: ВСЕ клиенты, сумма БГ > 50 млн	
	Бухгалтерская (финансовая) отчетность на последнюю календарную квартальную дату (форма «Бухгалтерский баланс»; форма «Отчет о финансовых результатах»)
	Годовая бухгалтерская (финансовая) отчетность (форма «Бухгалтерский баланс»; форма «Отчет о финансовых результатах»)
	Годовая бухгалтерская (финансовая) отчетность (форма «Бухгалтерский баланс»; форма «Отчет о финансовых результатах») за отчетный год, предшествующий последнему завершенному
Пакет БГ(И), Банк Открытие: ЮЛ на ОСНО, сумма БГ > 10 млн	
	Налоговая декларация по налогу на прибыль организации за последний налоговый период (год)
Пакет БГ(И), Банк Открытие: ЮЛ на УСН/ЕНВД/Сельхоз, сумма БГ > 10 млн	
	Налоговая декларация за последний налоговый период
Пакет БГ(И), Банк Открытие: для АО	
	Выписка из реестра акционеров
Пакет БГ(И), Банк Открытие: ЮЛ на УСН/ЕНВД/Сельхоз, сумма БГ>10 млн, 223 Ф3,615 Ф3	
	Расшифровка дебиторской задолженности на последнюю отчетную дату

6.4.2.1.5 Перечень данных контактного лица клиента

Наименование поля	Комментарий
Фамилия	
Имя	
Отчество	если есть

Наименование поля	Комментарий
Телефон	если есть
E-mail	если есть

6.4.2.1.6 Перечень данных об агенте, которому передается лид

Блок данных	Наименование поля	
Сведение об агенте - ЮЛ/ИП	ИНН агента	
	ОГРН/ОГРНИП агента	
Сведения о менеджере агента		Если указаны, то лид попадет сразу к конкретному менеджеру агента в Goodfin (менеджер должен существовать в системе Goodfin)
	Фамилия	
	Имя	
	E-mail	

7 Единый указатель по справочникам

7.1 Поля карточки компании: company_fields

Справочник полей, доступных при настройке обязательности данных в заявке по клиенту (компании/ИП).

id/мнемокод поля	Описание поля
legalType	Категория клиента (организация, ИП, физ. лицо)
mainContact	Основной контакт
fullName	Полное наименование организации
shortName	Сокращенное наименование организации
englishName	Наименование на иностранном языке
okopf	ОКОПФ, Организационно-правовая форма (полностью)
inn	ИНН
kpp	КПП
innDate	Дата выдачи ИНН
innPlace	Наименование органа, выдавшего ИНН
ogrn	ОГРН (ОГРНИП для ИП)
regNumber	Серия и номер свидетельства о регистрации
regDate	Дата государственной регистрации
regDocDate	Дата выдачи свидетельства о регистрации
regOrg	Наименование регистрирующего органа
regPlace	Место государственной регистрации
okpo	ОКПО
okato	ОКАТО
oktmo	ОКТМО
okogu	ОКОГУ
okfs	ОКФС
regCapital	Величина зарегистрированного уставного (складочного) капитала или уставного фонда, имущества (руб.)
paidCapital	Величина оплаченного уставного (складочного) капитала или уставного фонда, имущества (руб.)

id/мнемокод поля	Описание поля
taxSystem	Система налогообложения
pfr	Номер в пенсионном фонде РФ
fss	Номер в фонде социального страхования РФ
staff	Штатная численность сотрудников
topType	Тип высшего органа управления
legalAddress	Юридический адрес (адрес регистрации)
factAddress	Фактический адрес
corrAddress	Адрес для корреспонденции
licenses	Лицензии
legalAddress.kladr	КЛАДР
legalAddress.fias	ФИАС
legalAddress.postCode	Индекс
legalAddress.country	Страна
legalAddress.region	Область (республика, край)
legalAddress.district	Район
legalAddress.city	Населенный пункт (город, село и т.п.)
legalAddress.street	Улица, проспект, переулок и пр.
legalAddress.house	Дом
legalAddress.addressType	Тип адреса
legalAddress.isOwned	Помещение находится в собственности
legalAddress.rentOrOwnerContract	Документ собственности/аренды
legalAddress.rentOrOwnerContract.date	Дата документа
legalAddress.rentOrOwnerContract.serial	Серия документа
legalAddress.rentOrOwnerContract.num	Номер документа
legalAddress.rentOrOwnerContract.owner	Наименование собственника/арендодателя
legalAddress.rentOrOwnerContract.endDate	Срок действия
factAddress.kladr	КЛАДР
factAddress.fias	ФИАС
factAddress.postCode	Индекс
factAddress.country	Страна
factAddress.region	Область (республика, край)
factAddress.district	Район
factAddress.city	Населенный пункт (город, село и т.п.)

id/мнемокод поля	Описание поля
factAddress.street	Улица, проспект, переулок и пр.
factAddress.house	Дом
factAddress.addressType	Тип адреса
factAddress.isOwned	Помещение находится в собственности
factAddress.rentOrOwnerContract	Документ собственности/аренды
factAddress.rentOrOwnerContract.date	Дата документа
factAddress.rentOrOwnerContract.serial	Серия документа
factAddress.rentOrOwnerContract.num	Номер документа
factAddress.rentOrOwnerContract.owner	Наименование собственника/арендодателя
factAddress.rentOrOwnerContract.endDate	Срок действия
corrAddress.kladr	КЛАДР
corrAddress.fias	ФИАС
corrAddress.postCode	Индекс
corrAddress.country	Страна
corrAddress.region	Область (республика, край)
corrAddress.district	Район
corrAddress.city	Населенный пункт (город, село и т.п.)
corrAddress.street	Улица, проспект, переулок и пр.
corrAddress.house	Дом
corrAddress.addressType	Тип адреса
corrAddress.isPresent	Присутствует по юридическому адресу (по месту регистрации)
corrAddress.isPost	Используется как почтовый адрес
corrAddress.isOwned	Помещение находится в собственности
corrAddress.rentOrOwnerContract	Документ собственности/аренды
corrAddress.rentOrOwnerContract.date	Дата документа
corrAddress.rentOrOwnerContract.serial	Серия документа
corrAddress.rentOrOwnerContract.num	Номер документа
corrAddress.rentOrOwnerContract.owner	Наименование собственника/арендодателя
corrAddress.rentOrOwnerContract.endDate	Срок действия
licenses.activity	Вид деятельности
licenses.number	Номер лицензии
licenses.date	Дата выдачи лицензии

id/мнемокод поля	Описание поля
licenses.who	Кем выдана лицензия
licenses.listActivities	Перечень видов лицензируемой деятельности
licenses.dateend	Дата окончания действия лицензии
licenses.place	Место действия лицензии
baseOkved	Базовый ОКВЭД
okveds	ОКВЭД
orgType	Тип организации
bankingDetails	Реквизиты расчетного счета
bankingDetails.actualAddress	(Реквизиты расч. счета) Фактический адрес банка
bankingDetails.bankName	(Реквизиты расч. счета) Наименование банка
bankingDetails.bik	(Реквизиты расч. счета) БИК
bankingDetails.corrAccount	(Реквизиты расч. счета) Корреспондентский счет
bankingDetails.emails	(Реквизиты расч. счета) E-mail банка
bankingDetails.faxes	(Реквизиты расч. счета) Факс банка
bankingDetails.inn	(Реквизиты расч. счета) ИНН банка
bankingDetails.kbk	(Реквизиты расч. счета) КБК
bankingDetails.kpp	(Реквизиты расч. счета) КПП банка
bankingDetails.legalAddress	(Реквизиты расч. счета) Юридический адрес банка
bankingDetails.paymentAccount	(Реквизиты расч. счета) Расчетный счет
bankingDetails.phones	(Реквизиты расч. счета) Телефоны банка
bankingDetails.shortBankName	(Реквизиты расч. счета) Сокращенное наименование банка
bankingDetails.ufk	(Реквизиты расч. счета) УФК
bankingDetails.website	(Реквизиты расч. счета) Web-страница банка
corrAddress.building	(Почт. адрес) Строение/корпус
corrAddress.flat	(Почт. адрес) Офис/квартира
factAddress.building	(Факт. адрес) Строение/корпус
factAddress.flat	(Факт. адрес) Офис/квартира
legalAddress.building	(Юр. Адрес) Строение/корпус
legalAddress.flat	(Юр. Адрес) Офис/квартира
legalFounders	Учредители-ЮЛ

id/мнемокод поля	Описание поля
legalFounders.address	(Учредители-ЮЛ) Адрес организации одной строкой
legalFounders.inn	(Учредители-ЮЛ) ИНН организации
legalFounders.kpp	(Учредители-ЮЛ) КПП
legalFounders.name	(Учредители-ЮЛ) Полное наименование организации
legalFounders.ogrn	(Учредители-ЮЛ) ОГРН
legalFounders.percent	(Учредители-ЮЛ) Доля в уставном капитале, %
persons	Физ. лица - представители
persons.birthday	(Физ. лица - представители) Дата рождения
persons.birthdayPlace	(Физ. лица - представители) Место рождения
persons.firstName	(Физ. лица - представители) Имя
persons.inn	(Физ. лица - представители) ИНН
persons.lastName	(Физ. лица - представители) Фамилия
persons.middleName	(Физ. лица - представители) Отчество
persons.sex	(Физ. лица - представители) Пол
persons.snils	(Физ. лица - представители) СНИЛС
persons.mainOksm	(Физ. лица - представители) Гражданство (ОКСМ)
persons.otherOksm	(Физ. лица - представители) Двойное гражданство (ОКСМ)
persons.docType	(Физ. лица - представители, паспорт) Документ, удостоверяющий личность
persons.docSeries	(Физ. лица - представители, паспорт) Документ, удостоверяющий личность
persons.docNumber	(Физ. лица - представители, паспорт) Номер документа
persons.docWho	(Физ. лица - представители, паспорт) Кем выдан
persons.docDate	(Физ. лица - представители, паспорт) Дата выдачи
persons.docCodePod	(Физ. лица - представители, паспорт) Код подразделения
persons.docOther	(Физ. лица - представители, паспорт) Прочие сведения о документе
persons.docEndDate	(Физ. лица - представители, паспорт) Документ действителен до

id/мнемокод поля	Описание поля
persons.regAddress	Физ. лица - представители, адрес регистр.
persons.regAddress.addressString	(Физ. лица - представители, адрес регистр.) Адрес одной строкой
persons.regAddress.postCode	(Физ. лица - представители, адрес регистр.) Индекс
persons.regAddress.country	(Физ. лица - представители, адрес регистр.) Страна
persons.regAddress.region	(Физ. лица - представители, адрес регистр.) Регион/область
persons.regAddress.district	(Физ. лица - представители, адрес регистр.) Район
persons.regAddress.city	(Физ. лица - представители, адрес регистр.) Город/ населенный пункт
persons.regAddress.street	(Физ. лица - представители, адрес регистр.) Улица
persons.regAddress.house	(Физ. лица - представители, адрес регистр.) Дом
persons.regAddress.building	(Физ. лица - представители, адрес регистр.) Корпус
persons.regAddress.flat	(Физ. лица - представители, адрес регистр.) Офис/ квартира
persons.regAddress.fias	(Физ. лица - представители, адрес регистр.) ФИАС
persons.regAddress.kladr	(Физ. лица - представители, адрес регистр.) КЛАДР
persons.regAddress.isPresent	(Физ. лица - представители, адрес регистр.) Адрес регистрации совпадает с адресом фактического места жительства
persons.factAddress	(Физ. лица - представители, адрес места жит.)
persons.factAddress.addressString	(Физ. лица - представители, адрес места жит.) Адрес одной строкой
persons.factAddress.postCode	(Физ. лица - представители, адрес места жит.) Индекс
persons.factAddress.country	(Физ. лица - представители, адрес места жит.) Страна
persons.factAddress.region	(Физ. лица - представители, адрес места жит.) Регион/область
persons.factAddress.district	(Физ. лица - представители, адрес места жит.) Район
persons.factAddress.city	(Физ. лица - представители, адрес места жит.) Город/ населенный пункт

id/мнемокод поля	Описание поля
persons.factAddress.street	(Физ. лица - представители, адрес места жит.) Улица
persons.factAddress.house	(Физ. лица - представители, адрес места жит.) Дом
persons.factAddress.building	(Физ. лица - представители, адрес места жит.) Корпус
persons.factAddress.flat	(Физ. лица - представители, адрес места жит.) Офис/ квартира
persons.factAddress.fias	(Физ. лица - представители, адрес места жит.) ФИАС
persons.factAddress.kladr	(Физ. лица - представители, адрес места жит.) КЛАДР
persons.isPdl	(Физ. лица - представители, ПДЛ) Является публичным должностным лицом
persons.pdlType	(Физ. лица - представители, ПДЛ) Тип публичного должностного лица
persons.moneyPdl	(Физ. лица - представители, ПДЛ) Источник происхождения денежных средств ПДЛ

7.2 Статусы подключаемого сервиса: dealApplicationStatuses

Используется для настроек статусной модели подключаемого сервиса

Значение поля dealApplicationStatus в статусной модели (здесь приведены только те значения, которые нужны именно для настройки)	Описание поля
REJECTED_BY_CLIENT	Отклонена клиентом
CLOSED_BY_SYSTEM	Закрыта системой
REJECTED_BY_SERVICE	Отклонена сервисом (банком)
CANCELLED_BY_CLIENT	Отозвана клиентом
PRODUCT_ISSUED	Выпущен продукт. Значение указывается для всех успешных терминальных статусов, присваиваемых заявке в сервисе
SENT_TO_SERVICE	Отправлена в сервис
null	При получении статуса от внешнего сервиса статус заявки в Goodfin остаётся текущим

7.3 Прикрепляемые документы: docs_types

id/ мнемокод	Описание документа	Параметризованные типы документов (в настройках указать параметр и значение), начиная с версии 1.11
FINANCE_REPORT_LAST_QUARTER	Бухгалтерская (финансовая) отчетность на последнюю календарную квартальную дату (форма «Бухгалтерский баланс»; форма «Отчет о финансовых результатах»)	
MANAGEMENT_BALANCE_FIN_PERF_STATEMENT	Бухгалтерская (финансовая) отчетность на предыдущую календарную квартальную дату (форма «Бухгалтерский баланс»; форма «Отчет о финансовых результатах»)	
EXTRACT_FROM_INDIVIDUAL_ENTREPRENEUR_REGISTRY	Выписка из ЕГРИП	
EXTRACT_FROM_LEGAL_PERSON_REGISTRY	Выписка из ЕГРЮЛ	
EXTRACT_FROM_REGISTER_SHAREHOLDERS	Выписка из реестра акционеров	
ANALYTICAL_BALANCE_LAST_REPORTING_YEAR	Годовая бухгалтерская (финансовая) отчетность (форма «Бухгалтерский баланс»; форма «Отчет о финансовых результатах»)	
ANNUAL_ACCOUNTING_FINANCIAL_STATEMENTS	Годовая бухгалтерская (финансовая) отчетность (форма «Бухгалтерский баланс»; форма «Отчет о	

id/ мнемокод	Описание документа	Параметризованные типы документов (в настройках указать параметр и значение), начиная с версии 1.11
	финансовых результатах») за отчётный год, предшествующий последнему завершённому	
SIGNER_POWER_ATTORNEY	Доверенность на представителя клиента и документ, удостоверяющий личность представителя клиента (при наличии)	
ACCOUNT_OPEN_AGREEMENT_PROCUREMENT_PARTICIPANT	Договор об открытии специального счета участника закупок	
LEASE_AGREEMENT_OWNERSHIP	Документы по фактическому адресу (договор аренды (субаренды); свидетельство о праве собственности / выписка из Единого государственного реестра недвижимости)	
LEGAL_ADDR_LEASE_AGREEMENT_OWNERSHIP	Документы по юридическому адресу (договор аренды (субаренды); свидетельство о праве собственности / выписка из Единого государственного реестра недвижимости)	
BIDDING_DOCUMENTATION	Конкурсная документация	
BENEFICIARY_IDENTITY_DOC	Копия паспорта бенефициарного владельца	

id/ мнемокод	Описание документа	Параметризованные типы документов (в настройках указать параметр и значение), начиная с версии 1.11
HEAD_PASSPORT_COPY	Копия паспорта руководителя	
LICENSE	Лицензия	
TAX_RETURN_3NDFL_LAST_TAX_PERIOD	Налоговая декларация 3-НДФЛ за последний налоговый период (год)	
TAX_DECLARATION_LAST_PERIOD	Налоговая декларация за последний налоговый период	
TAX_DECL_VALUE_ADDED_TAX_LAST_TAX_PERIOD	Налоговая декларация по налогу на добавленную стоимость за последний налоговый период (квартал)	
TAX_DECL_ORGANIZATION_INCOME_TAX_LAST_QUARTER	Налоговая декларация по налогу на прибыль на последнюю календарную дату (квартал)	
TAX_DECL_ORGANIZATION_INCOME_TAX_LAST_TAX_PERIOD	Налоговая декларация по налогу на прибыль организации за последний налоговый период (год)	
TURNOVER_BALANCE_SHEET_90_91	Оборотно-сальдовые ведомости в разбивке субсчетов по счетам отчета о финансовых результатах: сч. 90, сч. 91 за каждый квартал за пять завершённых кварталов	
TURNOVER_BALANCE_SHEET_58_60	Оборотно-сальдовые ведомости в	

id/ мнемокод	Описание документа	Параметризованные типы документов (в настройках указать параметр и значение), начиная с версии 1.11
	разрезе субсчетов по балансовым счетам: сч. 58 , сч. 60 , сч. 62, сч. 66 , сч. 67 , сч. 76 , сч. 86 , на последнюю отчетную дату, и на дату предыдущего года, аналогичную последней отчетной дате	
TURNOVER_BALANCE_SHEET_008_009	Оборотно-сальдовые ведомости по забалансовым счетам 008, 009 на последнюю отчетную дату	
TURNOVER_BALANCE_SHEET_COMMON	Общая оборотно-сальдовая ведомость за последние пять кварталов (Оборотно-сальдовые ведомости по счетам бухгалтерского учета в разбивке по контрагентам и/или субсчетам предоставляются при наличии оборотов и/или остатков)	
FINANCIAL_PERFORMANCE_STATEMENT	Отчет о финансовых результатах на пять последних квартальных отчетных дат (форма №2 бухгалтерской отчетности)	
PATENT	Патент	
SUMMING_UP_PROTOCOL	Протокол подведения итогов	
OTHER_DOCUMENT	Прочие документы	

id/ мнемокод	Описание документа	Параметризованные типы документов (в настройках указать параметр и значение), начиная с версии 1.11
TURNOVER_BALANCE_SHEET_02	Развернутая оборотно-сальдовая ведомость сч. 02 за последние пять кварталов	
BREAKDOWN_ACCOUNTS_RECEIVABLE_LAST_REPORTING_DATE	Расшифровка дебиторской задолженности на последнюю отчетную дату	
ACCOUNT_DETAILS_PROCUREMENT_PARTICIPANT	Реквизиты специального счета участника закупок	
HEAD_APPOINTMENT_PROTOCOL	Решение/протокол о назначении руководителя	
MAJOR_TRANSACTION_APPROVAL_PROTOCOL	Решение/Протокол уполномоченного органа клиента об одобрении крупной сделки, а также решение/протокол об избрании коллегиального органа клиента, одобряющего крупную сделку.	параметр FOR_SERVICE Возможные значения и описание банка: FOR_SERVICE_OTKRYTIE - ОТКРЫТИЕ; FOR_SERVICE_TENDERBANK - ТЕНДЕР-БАНК; FOR_SERVICE_MTSBANK - МТС БАНК; FOR_SERVICE_ITCREDIT - ITCREDIT; FOR_SERVICE_URALSIB - УРАЛСИБ; FOR_SERVICE_PKBAN - ПКБ; FOR_SERVICE_SBERBANK - Сбербанк; FOR_SERVICE_RGSBANK - РГС Банк.
CHARTER	Устав в действующей редакции, изменения и/или дополнения в Устав (при наличии)	
MANAGEMENT_COMPANY	Документы по Управляющей	

id/ мнемокод	Описание документа	Параметризованные типы документов (в настройках указать параметр и значение), начиная с версии 1.11
	компания/Управляющему	
ELECTION_BORD_OF_DIRECTORS	Решение/протокол об избрании Совета директоров/Наблюдательного совета	
TRANSPORT_UNLOAD	Транспортная выгрузка	

7.4 Электронные торговые площадки: etp_types

Справочник электронных торговых площадок (ЭТП), используется для типа продукта "Тендерный займ".

мнемокод	url	Описание
SBER_AST	http://www.sberbank-ast.ru	СбербанкАСТ
ROSELTORG	http://etp.roseltorg.ru	Единая электронная торговая площадка (Росэлторг)
RTS_TENDER	http://rts-tender.ru	РТС-Тендер
SBER_UTP	http://utp.sberbank-ast.ru/	Сбербанк-УТП
ETP_ETS	https://etp-ets.ru/	Национальная электронная площадка
FABRIKANT	https://www.fabrikant.ru/	Фабрикант
AGZRT	http://www.agzrt.ru/	АО "АГЗ РТ"
LOT_ONLINE	http://lot-online.ru	lot-online
ETPRF	http://etprf.ru	etprf.ru

7.5 Федеральные законы: federallaw_types

Справочник федеральных законов, которые могут быть указаны для типов продуктов, связанных с процедурами (торгами, тендерами).

мнемокод	Описание ФЗ, согласно которого проводится процедура (торги, тендер)
185FL615PP	185 ФЗ/615 ПП
223FL	223-ФЗ

мнемокод	Описание ФЗ, согласно которого проводится процедура (торги, тендер)
44FL	44-ФЗ
94FL	94-ФЗ
COMMERCIAL_BIDDING	Коммерческие торги

7.6 Типы прикрепляемых файлов: file_types

Справочник типов файлов, загрузка которых поддерживается системой.

мнемокод	mimetype
pdf	application/pdf
jpg	image/jpeg
tif	image/tiff
jpeg	image/jpeg
png	image/png
rtf	application/rtf
docx	application/vnd.openxmlformats-officedocument.wordprocessingml.document
doc	application/msword
zip	application/zip
xls	application/vnd.ms-excel
xlsx	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet

7.7 Поля финансовых показателей: finance_fields

Также см. [Формы бухгалтерской и финансовой отчетности: finance_forms](#)

Мнемокод	Наименование показателя	КНД набора форм	Форма по ОКУД	Код/варианты кодов показателей согласно отчетности
noMaterial	Нематериальные, финансовые и другие внеоборотные активы (включая результаты исследований и разработок, незавершенные вложения в нематериальные активы,	0710098	0710001	1110, 1120, 1130, 1170, 1180, 1190

Мнемокод	Наименование показателя	КНД набора форм	Форма по ОКУД	Код/варианты кодов показателей согласно отчетности
	исследования и разработки, отложенные налоговые активы)			
material	Материальные внеоборотные активы (включая основные средства, незавершенные капитальные вложения в основные средства)	0710098	0710001	1140, 1150, 1160
1210	Запасы	0710098	0710001	1210
finAssets	Финансовые и другие оборотные активы (включая дебиторскую задолженность)	0710098	0710001	1220, 1230, 1240, 1260
1250	Денежные средства и денежные эквиваленты	0710098	0710001	1250
capital	Капитал и резервы (Целевые средства, Фонд недвижимого и особо ценного движимого имущества и иные целевые фонды)	0710098	0710001	1310, 1320, 1340, 1350, 1360, 1370
1410	Долгосрочные заемные средства	0710098	0710001	1410
longTermDuties	Другие долгосрочные обязательства	0710098	0710001	1420, 1430, 1450
1510	Краткосрочные заемные средства	0710098	0710001	1510
1520	Кредиторская задолженность	0710098	0710001	1520
shortTermDuties	Другие краткосрочные обязательства	0710098	0710001	1530, 1540, 1550
1600	БАЛАНС	0710098	0710001	1600
1700	БАЛАНС	0710098	0710001	1700
2110	Выручка за минусом налога на добавленную стоимость, акцизов	0710098	0710002	2110
expenses	Расходы по обычной деятельности (включая себестоимость продаж, коммерческие и управленческие расходы)	0710098	0710002	2120, 2210, 2220
otherIncomes	Прочие доходы	0710098	0710002	2310, 2320, 2340
2330	Проценты к уплате	0710098	0710002	2330
2350	Прочие расходы	0710098	0710002	2350
2400	Чистая прибыль (убыток)	0710098	0710002	2400
incomeTaxes	Налоги на прибыль (доходы) (включая текущий налог на	0710098	0710002	2410, 2411, 2412, 2530

Мнемокод	Наименование показателя	КНД набора форм	Форма по ОКУД	Код/варианты кодов показателей согласно отчетности
	прибыль, изменение отложенных налоговых обязательств и активов)			
1100	Итого по разделу I	0710099	0710001	1100
1110	Нематериальные активы	0710099	0710001	1110
1120	Результаты исследований и разработок	0710099	0710001	1120
1130	Нематериальные поисковые активы	0710099	0710001	1130
1140	Материальные поисковые активы	0710099	0710001	1140
1150	Основные средства	0710099	0710001	1150
1160	Доходные вложения в материальные ценности	0710099	0710001	1160
1170	Финансовые вложения	0710099	0710001	1170
1180	Отложенные налоговые активы	0710099	0710001	1180
1190	Прочие внеоборотные активы	0710099	0710001	1190
1200	Итого по разделу II	0710099	0710001	1200
1210	Запасы	0710099	0710001	1210
1220	Налог на добавленную стоимость по приобретенным ценностям	0710099	0710001	1220
1230	Дебиторская задолженность	0710099	0710001	1230
1240	Финансовые вложения (за исключением денежных эквивалентов)	0710099	0710001	1240
1250	Денежные средства и денежные эквиваленты	0710099	0710001	1250
1260	Прочие оборотные активы	0710099	0710001	1260
1300	Итого по разделу III	0710099	0710001	1300
1310	Уставный капитал (складочный капитал, уставный фонд, вклады товарищей)	0710099	0710001	1310
1320	Собственные акции, выкупленные у акционеров	0710099	0710001	1320
1340	Переоценка внеоборотных активов	0710099	0710001	1340
1350	Добавочный капитал (без переоценки)	0710099	0710001	1350

Мнемокод	Наименование показателя	КНД набора форм	Форма по ОКУД	Код/варианты кодов показателей согласно отчетности
1360	Резервный капитал	0710099	0710001	1360
1370	Нераспределенная прибыль (непокрытый убыток)	0710099	0710001	1370
1400	Итого по разделу IV	0710099	0710001	1400
1410	Заемные средства	0710099	0710001	1410
1420	Отложенные налоговые обязательства	0710099	0710001	1420
1430	Оценочные обязательства	0710099	0710001	1430
1450	Прочие обязательства	0710099	0710001	1450
1500	Итого по разделу V	0710099	0710001	1500
1510	Заемные средства	0710099	0710001	1510
1520	Кредиторская задолженность	0710099	0710001	1520
1530	Доходы будущих периодов	0710099	0710001	1530
1540	Оценочные обязательства	0710099	0710001	1540
1550	Прочие обязательства	0710099	0710001	1550
1600	БАЛАНС	0710099	0710001	1600
1700	БАЛАНС	0710099	0710001	1700
2100	Валовая прибыль (убыток)	0710099	0710002	2100
2110	Выручка за минусом налога на добавленную стоимость, акцизов	0710099	0710002	2110
2120	Себестоимость продаж	0710099	0710002	2120
2200	Прибыль (убыток) от продаж	0710099	0710002	2200
2210	Коммерческие расходы	0710099	0710002	2210
2220	Управленческие расходы	0710099	0710002	2220
2300	Прибыль (убыток) до налогообложения	0710099	0710002	2300
2310	Доходы от участия в других организациях	0710099	0710002	2310
2320	Проценты к получению	0710099	0710002	2320
2330	Проценты к уплате	0710099	0710002	2330
2340	Прочие доходы	0710099	0710002	2340
2350	Прочие расходы	0710099	0710002	2350

Мнемокод	Наименование показателя	КНД набора форм	Форма по ОКУД	Код/варианты кодов показателей согласно отчетности
2400	Чистая прибыль (убыток)	0710099	0710002	2400
2410	Текущий налог на прибыль	0710099	0710002	2410
2411	в т.ч. текущий налог на прибыль	0710099	0710002	2411
2412	отложенный налог на прибыль	0710099	0710002	2412
2460	Прочее	0710099	0710002	2460
2500	Совокупный финансовый результат периода	0710099	0710002	2500
2510	Результат от переоценки внеоборотных активов, не включаемый в чистую прибыль (убыток) периода	0710099	0710002	2510
2520	Результат от прочих операций, не включаемый в чистую прибыль (убыток) периода	0710099	0710002	2520
2530	Налог на прибыль от операций, результат которых не включается в чистую прибыль (убыток) периода	0710099	0710002	2530
2900	Справочно Базовая прибыль (убыток) на акцию	0710099	0710002	2900
2910	Разводненная прибыль (убыток) на акцию	0710099	0710002	2910

7.8 Формы бухгалтерской и финансовой отчетности: finance_forms

Наименование формы	КНД набора форм	Форма по ОКУД	Идентификатор в формате UUID (может использоваться в настройках правил применимости)	Примечание
Бухгалтерская (финансовая) отчетность с формой 1 (ОКУД 0710001) и формой 2(ОКУД 0710002)				
Бухгалтерский баланс (форма 1)	0710099	0710001	1b010932-3e5b-4992-861e-b3648f83a03d	Используется по умолчанию
Отчет о финансовых результатах (форма 2)	0710099	0710002	a48142d6-ded8-41c2-bfbf-57035a4fc773	Используется по умолчанию

Наименование формы	КНД набора форм	Форма по ОКУД	Идентификатор в формате UUID (может использоваться в настройках правил применимости)	Примечание
Упрощенная отчетность субъектов малого предпринимательства				
Бухгалтерский баланс (форма 1)	0710098	0710001	066a518c-7d17-4f84-9f21-350302b9342f	Временно не используется в Системе (отключен)
Отчет о финансовых результатах (форма 2)	0710098	0710002	009c267f-aad2-42d5-b716-51d40468998c	Временно не используется в Системе (отключен)

7.9 Документы, удостоверяющие личность: identity_doc_types

Справочник документов, удостоверяющих личность, которые поддерживаются при вводе сведений о физических лицах - представителях компании/ИП.

мнемокод	Описание документа, удостоверяющего личность
PASSPORT	Паспорт гражданина РФ
INTERNATIONAL_PASSPORT	Загранпаспорт гражданина РФ
FOREIGN_PASSPORT	Иностранный паспорт
MILITARY_ID	Военный билет
OFFICER_ID	Удостоверение личности офицера
MINMORFLOT_PASSPORT	Паспорт Минморфлота
DIPLOMATIC_PASSPORT	Дипломатический паспорт гражд. РФ
SEAMAN_PASSPORT	Паспорт моряка
MILITARY_RESERVE_OFFICER_ID	Военный билет офицера запаса
BIRTH_CERTIFICATE	Свидетельство о рождении
OTHER	Иной документ

7.10 Формы предпринимательской деятельности: legal_types

мнемокод	Описание
ORGANIZATION	Юридическое лицо, филиал ЮЛ и пр. типы организаций
IP	Индивидуальный предприниматель

7.11 Отчётные периоды: `period_types`

Мнемокод	Наименование отчетного периода
<code>currentReportingPeriod</code>	На отчетную дату отчетного периода (квартал)
<code>samePeriodPreviousYear</code>	За аналогичный период предыдущего года (квартал)
<code>previousPeriod</code>	На отчетную дату предыдущего отчетного периода (квартал)
<code>periodPrecedingPreviousOne</code>	На отчетную дату периода, предшест. предыдущему (квартал)
<code>periodPrecedingPreviousTwo</code>	На отчетную дату периода, предшест. двум предыдущим (квартал)
<code>periodPrecedingPreviousThree</code>	На отчетную дату периода, предшест. трем предыдущим (квартал)
<code>previousYear</code>	На 31 декабря предыдущего года
<code>yearPrecedingPrevious</code>	На 31 декабря года, предшествующего предыдущему

7.12 Способы размещения заказа: `placingway_types`

id/мнемокод способа размещения заказа	Способ размещения заказа
<code>E_AUCTION</code>	ЭА (Электронный аукцион)
<code>SOLE SUPPLIER</code>	Закупка у единственного поставщика (исполнителя, подрядчика)
<code>TWO_STAGE_TENDER</code>	ОК-Д (двухэтапный конкурс)
<code>RESTRICTED_TENDER</code>	ОК-ОУ (конкурс с ограниченным участием)
<code>OPEN_TENDER</code>	Открытый конкурс
<code>CLOSED_AUCTION</code>	ЗакА (закрытый аукцион)
<code>CLOSED_TWO_STAGE_TENDER</code>	ЗакК-Д (закрытый двухэтапный конкурс)
<code>CLOSED_RESTRICTED_TENDER</code>	ЗакК-ОУ (закрытый конкурс с ограниченным участием)
<code>CLOSED_TENDER</code>	ЗакК (закрытый конкурс)
<code>QUOTATIONS_REQUEST</code>	ЗК (запрос котировок)
<code>OFFER_REQUEST</code>	ЗП (запрос предложений)
<code>E_AUCTION_94FL</code>	АУКЦИОН В ЭЛЕКТРОННОЙ ФОРМЕ (ПО 94-ФЗ)
<code>TENDER_44FL</code>	Конкурс по 44-ФЗ

id/мнемокод способа размещения заказа	Способ размещения заказа
TENDER_223FL	Конкурс по 223-ФЗ
TENDER_185FL615PP	Конкурс по 185ФЗ/615ПП

7.13 Типы продуктов: product_types

id/мнемокод типа продукта	Наименование типа продукта	Тип продукта для ЮЛ, ИП	Префикс для номера сделки
ADVANCE_BG	Банковская гарантия на возврат аванса	True	БГ(А)
BG	Банковская гарантия на исполнение контракта	True	БГ(И)
CASH_SETTLEMENT_SERVICE	Расчетно-кассовое обслуживание	True	РКО
CONTRACT_EXEC_CREDIT	Кредит на исполнение контракта	True	КиК
EDS	Выпуск электронной подписи	True	ЭП
INDIVIDUAL_CREDIT	Потребительский кредит	False	Кредит
INVESTOR_LOAN	Заём от инвесторов	True	Заём(И)
PARTICIPATION_BG	Банковская гарантия на участие	True	БГ(У)
SMALL_BUSINESS_LOAN	Кредит для малого и среднего бизнеса	True	К(МСБ)
TENDER_LOAN	Тендерный займ	True	ТЗ
WARRANTY_BG	Банковская гарантия обеспечения гарантийных обязательств	True	БГ(О)

7.14 Поля в заявках по продуктам: product_types_fields

Типы продуктов, которым соответствуют поля по префиксам см. в [Типы продуктов: product_types](#)

id/мнемокод поля	Описание поля
ADVANCE_BG:productDealState.actualContractPrice	Фактическая цена контракта
ADVANCE_BG:productDealState.amount	Сумма БГ

id/мнемокод поля	Описание поля
ADVANCE_BG:productDealState.concludedContracts	Количество всех заключенных контрактов
ADVANCE_BG:productDealState.customer	Сведения о заказчике
ADVANCE_BG:productDealState.customer.bankAccountCorr	Корреспондентский счет
ADVANCE_BG:productDealState.customer.bankAccountMain	Номер расчетного счета заказчика
ADVANCE_BG:productDealState.customer.bankBik	БИК банка заказчика
ADVANCE_BG:productDealState.customer.bankName	Наименование банка заказчика
ADVANCE_BG:productDealState.customer.consolidatedRegNum	Код заказчика по сводному реестру
ADVANCE_BG:productDealState.customer.contactEmail	Контактный адрес электронной почты заказчика
ADVANCE_BG:productDealState.customer.contactFax	Контактный факс заказчика
ADVANCE_BG:productDealState.customer.contactFio	ФИО контактного лица заказчика
ADVANCE_BG:productDealState.customer.contactPhone	Контактный телефон заказчика
ADVANCE_BG:productDealState.customer.factAddress	Адрес заказчика (фактический)
ADVANCE_BG:productDealState.customer.iku	Идентификационный код заказчика
ADVANCE_BG:productDealState.customer.ikz	Идентификационный код закупки
ADVANCE_BG:productDealState.customer.inn	ИНН заказчика
ADVANCE_BG:productDealState.customer.kpp	КПП заказчика
ADVANCE_BG:productDealState.customer.name	Наименование заказчика
ADVANCE_BG:productDealState.customer.ogrn	ОГРН заказчика
ADVANCE_BG:productDealState.customer.oktmo	ОКТМО заказчика
ADVANCE_BG:productDealState.customer.postAddress	Адрес заказчика (юридический/почтовый)
ADVANCE_BG:productDealState.customer.region	Регион заказчика

id/мнемокод поля	Описание поля
ADVANCE_BG:productDealState.customer.regNum	Код в Сводном Перечне Заказчиков
ADVANCE_BG:productDealState.customer.siteUrl	Адрес сайта заказчика
ADVANCE_BG:productDealState.deadlineForGrant	Крайний срок получения БГ
ADVANCE_BG:productDealState.deliveryIsRequired	Требуется доставка оригиналов документов?
ADVANCE_BG:productDealState.endDate	Дата окончания действия БГ
ADVANCE_BG:productDealState.hasLoss	Наличие убытков
ADVANCE_BG:productDealState.isIndisputableWriteOff	Контрактом предусмотрено беспорное списание?
ADVANCE_BG:productDealState.isQuickDecision	Необходимо ускоренное принятие решения?
ADVANCE_BG:productDealState.loanCommitments	Сумма текущих кредитных обязательств (во всех кредитных организациях) и займов (руб.)
ADVANCE_BG:productDealState.startDate	Дата начала действия БГ
ADVANCE_BG:productDealState.tender	Сведения о процедуре (тендере)
ADVANCE_BG:productDealState.tender.closedLaw	Тендер по 275-ФЗ (исполнение гособоронзаказа)?
ADVANCE_BG:productDealState.tender.closedRegister	Закрытый реестр размещения гарантии?
ADVANCE_BG:productDealState.tender.currencyType	Валюта контракта
ADVANCE_BG:productDealState.tender.enforceAmount	Размер обеспечения исполнения контракта
ADVANCE_BG:productDealState.tender.federalLaw	Федеральный закон

id/мнемокод поля	Описание поля
ADVANCE_BG:productDealState.tender.initialContractPrice	Начальная (максимальная) цена контракта
ADVANCE_BG:productDealState.tender.isMultiCustomer	Совместные торги
ADVANCE_BG:productDealState.tender.isMultiLot	Мультилотовая процедура (тендер)
ADVANCE_BG:productDealState.tender.lotName	Предмет лота
ADVANCE_BG:productDealState.tender.lotNumber	Номер лота
ADVANCE_BG:productDealState.tender.name	Предмет процедуры (тендера)
ADVANCE_BG:productDealState.tender.okpds	Перечень ОКПД
ADVANCE_BG:productDealState.tender.okveds	Перечень ОКВЭД
ADVANCE_BG:productDealState.tender.placementType	Способ размещения заказа
ADVANCE_BG:productDealState.tender.publishDateTime	Дата/время публикации процедуры (тендера)
ADVANCE_BG:productDealState.tender.registryNumber	Реестровый номер
ADVANCE_BG:productDealState.tender.url	Ссылка на процедуру (тендер)
ADVANCE_BG:productDealState.tender.winnerDateTime	Дата/время протокола подведения итогов
ADVANCE_BG:productDealState.tender.winnerProtocol	Номер протокола подведения итогов
ADVANCE_BG:productDealState.termDays	Срок БГ (дней)
BG:productDealState.actualContractPrice	Фактическая цена контракта
BG:productDealState.amount	Сумма БГ
BG:productDealState.concludedContracts	Количество всех заключенных контрактов
BG:productDealState.customer	Сведения о заказчике
BG:productDealState.customer.bankAccountCorr	Корреспондентский счет
BG:productDealState.customer.bankAccountMain	Номер расчетного счета заказчика

id/мнемокод поля	Описание поля
BG:productDealState.customer.bankBik	БИК банка заказчика
BG:productDealState.customer.bankName	Наименование банка заказчика
BG:productDealState.customer.consolidatedRegNum	Код заказчика по сводному реестру
BG:productDealState.customer.contactEmail	Контактный адрес электронной почты заказчика
BG:productDealState.customer.contactFax	Контактный факс заказчика
BG:productDealState.customer.contactFio	ФИО контактного лица заказчика
BG:productDealState.customer.contactPhone	Контактный телефон заказчика
BG:productDealState.customer.factAddress	Адрес заказчика (фактический)
BG:productDealState.customer.iku	Идентификационный код заказчика
BG:productDealState.customer.ikz	Идентификационный код закупки
BG:productDealState.customer.inn	ИНН заказчика
BG:productDealState.customer.kpp	КПП заказчика
BG:productDealState.customer.name	Наименование заказчика
BG:productDealState.customer.ogrn	ОГРН заказчика
BG:productDealState.customer.oktmo	ОКТМО заказчика
BG:productDealState.customer.postAddress	Адрес заказчика (юридический/почтовый)
BG:productDealState.customer.region	Регион заказчика
BG:productDealState.customer.regNum	Код в Сводном Перечне Заказчиков
BG:productDealState.customer.siteUrl	Адрес сайта заказчика
BG:productDealState.deadlineForGrant	Крайний срок получения БГ
BG:productDealState.deliveryIsRequired	Требуется доставка оригиналов документов?

id/мнемокод поля	Описание поля
BG:productDealState.endDate	Дата окончания действия БГ
BG:productDealState.hasLoss	Наличие убытков
BG:productDealState.isIndisputableWriteOff	Контрактом предусмотрено беспорное списание?
BG:productDealState.isPrepayment	С авансом
BG:productDealState.isQuickDecision	Необходимо ускоренное принятие решения?
BG:productDealState.isWarrantyPeriod	Необходимо обеспечение гарантийного периода?
BG:productDealState.loanCommitments	Сумма текущих кредитных обязательств (во всех кредитных организациях) и займов (руб.)
BG:productDealState.prepaymentAmount	Сумма аванса
BG:productDealState.prepaymentPercent	Размер аванса (%)
BG:productDealState.startDate	Дата начала действия БГ
BG:productDealState.tender	Сведения о процедуре (тендере)
BG:productDealState.tender.closedLaw	Тендер по 275-ФЗ (исполнение гособоронзаказа)?
BG:productDealState.tender.closedRegister	Закрытый реестр размещения гарантии?
BG:productDealState.tender.currencyType	Валюта контракта
BG:productDealState.tender.enforceAmount	Размер обеспечения исполнения контракта
BG:productDealState.tender.federalLaw	Федеральный закон
BG:productDealState.tender.initialContractPrice	Начальная (максимальная) цена контракта
BG:productDealState.tender.isMultiCustomer	Совместные торги

id/мнемокод поля	Описание поля
BG:productDealState.tender.isMultiLot	Мультилотовая процедура (тендер)
BG:productDealState.tender.lotName	Предмет лота
BG:productDealState.tender.lotNumber	Номер лота
BG:productDealState.tender.name	Предмет процедуры (тендера)
BG:productDealState.tender.okpds	Перечень ОКПД
BG:productDealState.tender.okveds	Перечень ОКВЭД
BG:productDealState.tender.placementType	Способ размещения заказа
BG:productDealState.tender.publishDateTime	Дата/время публикации процедуры (тендера)
BG:productDealState.tender.registryNumber	Реестровый номер
BG:productDealState.tender.url	Ссылка на процедуру (тендер)
BG:productDealState.tender.winnerDateTime	Дата/время протокола подведения итогов
BG:productDealState.tender.winnerProtocol	Номер протокола подведения итогов
BG:productDealState.termDays	Срок БГ (дней)
CASH_SETTLEMENT_SERVICE:productDealState.city	Город присутствия офиса банка
CASH_SETTLEMENT_SERVICE:productDealState.email	E-mail
CASH_SETTLEMENT_SERVICE:productDealState.fullName	ФИО
CASH_SETTLEMENT_SERVICE:productDealState.phone	Номер телефона
CONTRACT_EXEC_CREDIT:productDealState.actualContractPrice	Фактическая цена контракта
CONTRACT_EXEC_CREDIT:productDealState.amount	Сумма кредита
CONTRACT_EXEC_CREDIT:productDealState.concludedContracts	Количество всех заключенных контрактов
CONTRACT_EXEC_CREDIT:productDealState.customer	Сведения о заказчике
CONTRACT_EXEC_CREDIT:productDealState.customer.bankAccountCoff	Корреспондентский счет банка заказчика

id/мнемокод поля	Описание поля
CONTRACT_EXEC_CREDIT:productDealState.customer.bankAccountMain	Номер основного расчетного счета заказчика
CONTRACT_EXEC_CREDIT:productDealState.customer.bankBik	БИК банка заказчика
CONTRACT_EXEC_CREDIT:productDealState.customer.bankName	Наименование банка заказчика
CONTRACT_EXEC_CREDIT:productDealState.customer.consolidatedRegNum	Код заказчика по сводному реестру
CONTRACT_EXEC_CREDIT:productDealState.customer.contactEmail	Контактный адрес электронной почты заказчика
CONTRACT_EXEC_CREDIT:productDealState.customer.contactFax	Контактный факс заказчика
CONTRACT_EXEC_CREDIT:productDealState.customer.contactFio	ФИО контактного лица заказчика
CONTRACT_EXEC_CREDIT:productDealState.customer.contactPhone	Контактный телефон заказчика
CONTRACT_EXEC_CREDIT:productDealState.customer.factAddress	Адрес заказчика (фактический)
CONTRACT_EXEC_CREDIT:productDealState.customer.inn	ИНН заказчика
CONTRACT_EXEC_CREDIT:productDealState.customer.kpp	КПП заказчика
CONTRACT_EXEC_CREDIT:productDealState.customer.name	Наименование заказчика
CONTRACT_EXEC_CREDIT:productDealState.customer.ogrn	ОГРН заказчика
CONTRACT_EXEC_CREDIT:productDealState.customer.oktmo	ОКТМО заказчика
CONTRACT_EXEC_CREDIT:productDealState.customer.postAddress	Адрес заказчика (юридический/почтовый)
CONTRACT_EXEC_CREDIT:productDealState.customer.region	Регион заказчика
CONTRACT_EXEC_CREDIT:productDealState.customer.regNum	Код по СПЗ
CONTRACT_EXEC_CREDIT:productDealState.customer.siteUrl	Адрес сайта заказчика
CONTRACT_EXEC_CREDIT:productDealState.endDate	Планируемая дата возврата кредита
CONTRACT_EXEC_CREDIT:productDealState.hasLoss	Наличие убытков
CONTRACT_EXEC_CREDIT:productDealState.isIndisputableWriteOff	Контрактом предусмотрено беспорное списание?

id/мнемокод поля	Описание поля
CONTRACT_EXEC_CREDIT:productDealState.isNeedPerformanceBg	Потребуется БГ на исполнение контракта?
CONTRACT_EXEC_CREDIT:productDealState.isPrepayment	В контракте оговорен аванс?
CONTRACT_EXEC_CREDIT:productDealState.loanCommitments	Сумма текущих кредитных обязательств (во всех кредитных организациях) и займов
CONTRACT_EXEC_CREDIT:productDealState.performanceBg.amount	Сумма БГ на исполнение
CONTRACT_EXEC_CREDIT:productDealState.performanceBg.endDate	Дата окончания действия БГ на исполнение (дд.мм.гггг)
CONTRACT_EXEC_CREDIT:productDealState.performanceBg.startDate	Дата начала действия БГ на исполнение (дд.мм.гггг)
CONTRACT_EXEC_CREDIT:productDealState.performanceBg.termDays	Срок БГ на исполнение (дней)
CONTRACT_EXEC_CREDIT:productDealState.prepaymentAmount	Сумма аванса
CONTRACT_EXEC_CREDIT:productDealState.prepaymentPercent	Размер аванса (%)
CONTRACT_EXEC_CREDIT:productDealState.startDate	Планируемая дата выдачи кредита
CONTRACT_EXEC_CREDIT:productDealState.tender	Сведения о процедуре (тендере)
CONTRACT_EXEC_CREDIT:productDealState.tender.currencyType	Валюта контракта
CONTRACT_EXEC_CREDIT:productDealState.tender.enforceAmount	Размер обеспечения исполнения контракта
CONTRACT_EXEC_CREDIT:productDealState.tender.federalLaw	Федеральный закон, регулирующий торги
CONTRACT_EXEC_CREDIT:productDealState.tender.initialContractPrice	Начальная (максимальная) цена контракта
CONTRACT_EXEC_CREDIT:productDealState.tender.isMultiCustomer	Процедура (тендер) с несколькими заказчиками

id/мнемокод поля	Описание поля
CONTRACT_EXEC_CREDIT:productDealState.tender.isMultiLot	Процедура (тендер) с несколькими лотами
CONTRACT_EXEC_CREDIT:productDealState.tender.lotName	Предмет лота
CONTRACT_EXEC_CREDIT:productDealState.tender.lotNumber	Номер лота
CONTRACT_EXEC_CREDIT:productDealState.tender.name	Предмет процедуры (тендера)
CONTRACT_EXEC_CREDIT:productDealState.tender.okpds	ОКПД (список кодов)
CONTRACT_EXEC_CREDIT:productDealState.tender.okveds	ОКВЭД (список кодов)
CONTRACT_EXEC_CREDIT:productDealState.tender.placementType	Способ размещения заказа
CONTRACT_EXEC_CREDIT:productDealState.tender.publishDateTime	Дата и время публикации процедуры (тендера) (дд.мм.гггг чч:мм:сс)
CONTRACT_EXEC_CREDIT:productDealState.tender.registryNumber	Реестровый номер процедуры (тендера)
CONTRACT_EXEC_CREDIT:productDealState.tender.url	Ссылка на процедуру (тендер)
CONTRACT_EXEC_CREDIT:productDealState.termDays	Срок кредита (дней)
EDS:productDealState.edsAddServices	Дополнительные услуги
EDS:productDealState.edsEmail	Е-mail, который будет записан в ЭП
EDS:productDealState.edsType	Вид ЭП
INDIVIDUAL_CREDIT:productDealState.additionalContactFirstName	Имя контактного лица
INDIVIDUAL_CREDIT:productDealState.additionalContactLastName	Фамилия контактного лица
INDIVIDUAL_CREDIT:productDealState.additionalContactMiddleName	Отчество контактного лица
INDIVIDUAL_CREDIT:productDealState.additionalContactPhone	Мобильный телефон контактного лица
INDIVIDUAL_CREDIT:productDealState.amount	Сумма кредита

id/мнемокод поля	Описание поля
INDIVIDUAL_CREDIT:productDealState.amountAdditionalIncome	Сумма дополнительного дохода, руб
INDIVIDUAL_CREDIT:productDealState.amountFamilyIncome	Доход супруга/супруги
INDIVIDUAL_CREDIT:productDealState.amountIncome	Доход по основному месту работы или пенсия
INDIVIDUAL_CREDIT:productDealState.bankBranch	Отделение банка
INDIVIDUAL_CREDIT:productDealState.beneficiary	Тип выгодоприобретателя
INDIVIDUAL_CREDIT:productDealState.cardHolder	Имя и Фамилия на карте
INDIVIDUAL_CREDIT:productDealState.creditCityKladr	Город получения кредита
INDIVIDUAL_CREDIT:productDealState.creditDelivery	Метод получения кредита
INDIVIDUAL_CREDIT:productDealState.creditDocType	Способ подтверждения дохода
INDIVIDUAL_CREDIT:productDealState.creditIncomeType	Вид дополнительного дохода
INDIVIDUAL_CREDIT:productDealState.creditPurposeType	Цель кредита
INDIVIDUAL_CREDIT:productDealState.isAdditionalIncome	У Вас есть дополнительный доход?
INDIVIDUAL_CREDIT:productDealState.isBankrupt	Были ли Вы признаны банкротом в течение последних 5 лет?
INDIVIDUAL_CREDIT:productDealState.isBeneficiary	Наличие выгодоприобретателя
INDIVIDUAL_CREDIT:productDealState.isConsent	Согласие на обработку персональных данных
INDIVIDUAL_CREDIT:productDealState.isPDL	Являетесь ли Вы РПДЛ/МПДЛ?
INDIVIDUAL_CREDIT:productDealState.isRealty	Есть недвижимое имущество в собственности?

id/мнемокод поля	Описание поля
INDIVIDUAL_CREDIT:productDealState.isVehicle	Автомобиль в собственности?
INDIVIDUAL_CREDIT:productDealState.relationshipType	Тип контактного лица
INDIVIDUAL_CREDIT:productDealState.restructuring	В течение последних 8-и лет судом утверждался план реструктуризации Ваших долгов?
INDIVIDUAL_CREDIT:productDealState.secretCode	Кодовое слово
INDIVIDUAL_CREDIT:productDealState.termMonths	Срок кредита
INDIVIDUAL_CREDIT:productDealState.withPDL	Состоите ли Вы в родстве с РПДЛ/МПДЛ?
INVESTOR_LOAN:productDealState.actualContractPrice	Фактическая цена контракта
INVESTOR_LOAN:productDealState.amount	Максимальная сумма займа
INVESTOR_LOAN:productDealState.customer	Сведения о заказчике
INVESTOR_LOAN:productDealState.customer.bankAccountCorr	Корреспондентский счет банка заказчика
INVESTOR_LOAN:productDealState.customer.bankAccountMain	Номер основного расчетного счета заказчика
INVESTOR_LOAN:productDealState.customer.bankBik	БИК банка заказчика
INVESTOR_LOAN:productDealState.customer.bankName	Наименование банка заказчика
INVESTOR_LOAN:productDealState.customer.consolidatedRegNum	Код заказчика по сводному реестру
INVESTOR_LOAN:productDealState.customer.contactEmail	Контактный адрес электронной почты заказчика
INVESTOR_LOAN:productDealState.customer.contactFax	Контактный факс заказчика
INVESTOR_LOAN:productDealState.customer.contactFio	ФИО контактного лица заказчика
INVESTOR_LOAN:productDealState.customer.contactPhone	Контактный телефон заказчика

id/мнемокод поля	Описание поля
INVESTOR_LOAN:productDealState.customer.factAddress	Адрес заказчика (фактический)
INVESTOR_LOAN:productDealState.customer.inn	ИНН заказчика
INVESTOR_LOAN:productDealState.customer.kpp	КПП заказчика
INVESTOR_LOAN:productDealState.customer.name	Наименование заказчика
INVESTOR_LOAN:productDealState.customer.ogrn	ОГРН заказчика
INVESTOR_LOAN:productDealState.customer.oktmo	ОКТМО заказчика
INVESTOR_LOAN:productDealState.customer.postAddress	Адрес заказчика (юридический/почтовый)
INVESTOR_LOAN:productDealState.customer.region	Регион заказчика
INVESTOR_LOAN:productDealState.customer.regNum	Код по СПЗ
INVESTOR_LOAN:productDealState.customer.siteUrl	Адрес сайта заказчика
INVESTOR_LOAN:productDealState.isPrepayment	В контракте оговорен аванс?
INVESTOR_LOAN:productDealState.minAmount	Минимальная сумма займа
INVESTOR_LOAN:productDealState.note	Примечание
INVESTOR_LOAN:productDealState.prepaymentAmount	Сумма аванса
INVESTOR_LOAN:productDealState.prepaymentPercent	Размер аванса (%)
INVESTOR_LOAN:productDealState.purpose	Цель займа
INVESTOR_LOAN:productDealState.startDate	Планируемая дата получения займа
INVESTOR_LOAN:productDealState.tender	Сведения о процедуре (тендере)
INVESTOR_LOAN:productDealState.tender.currencyType	Валюта контракта
INVESTOR_LOAN:productDealState.tender.enforceAmount	Размер обеспечения исполнения контракта
INVESTOR_LOAN:productDealState.tender.federalLaw	Федеральный закон, регулирующий торги
INVESTOR_LOAN:productDealState.tender.initialContractPrice	Начальная (максимальная) цена контракта

id/мнемокод поля	Описание поля
INVESTOR_LOAN:productDealState.tender.isMultiCustomer	Процедура (тендер) с несколькими заказчиками
INVESTOR_LOAN:productDealState.tender.isMultiLot	Процедура (тендер) с несколькими лотами
INVESTOR_LOAN:productDealState.tender.lotName	Предмет лота
INVESTOR_LOAN:productDealState.tender.lotNumber	Номер лота
INVESTOR_LOAN:productDealState.tender.name	Предмет процедуры (тендера)
INVESTOR_LOAN:productDealState.tender.okpds	ОКПД (список кодов)
INVESTOR_LOAN:productDealState.tender.okveds	ОКВЭД (список кодов)
INVESTOR_LOAN:productDealState.tender.placementType	Способ размещения заказа
INVESTOR_LOAN:productDealState.tender.publishDateTime	Дата и время публикации процедуры (тендера) (дд.мм.гггг чч:мм:сс)
INVESTOR_LOAN:productDealState.tender.registryNumber	Реестровый номер процедуры (тендера)
INVESTOR_LOAN:productDealState.tender.url	Ссылка на процедуру (тендер)
INVESTOR_LOAN:productDealState.termMonths	Срок займа (мес.)
PARTICIPATION_BG:productDealState.actualContractPrice	Фактическая цена контракта
PARTICIPATION_BG:productDealState.amount	Сумма БГ на участие
PARTICIPATION_BG:productDealState.customer	Сведения о заказчике
PARTICIPATION_BG:productDealState.customer.bankAccountCorr	Корреспондентский счет банка заказчика
PARTICIPATION_BG:productDealState.customer.bankAccountMain	Номер основного расчетного счета заказчика
PARTICIPATION_BG:productDealState.customer.bankBik	БИК банка заказчика

id/мнемокод поля	Описание поля
PARTICIPATION_BG:productDealState.customer.bankName	Наименование банка заказчика
PARTICIPATION_BG:productDealState.customer.consolidatedRegNum	Код заказчика по сведному реестру
PARTICIPATION_BG:productDealState.customer.contactEmail	Контактный адрес электронной почты заказчика
PARTICIPATION_BG:productDealState.customer.contactFax	Контактный факс заказчика
PARTICIPATION_BG:productDealState.customer.contactFio	ФИО контактного лица заказчика
PARTICIPATION_BG:productDealState.customer.contactPhone	Контактный телефон заказчика
PARTICIPATION_BG:productDealState.customer.factAddress	Адрес заказчика (фактический)
PARTICIPATION_BG:productDealState.customer.iku	Идентификационный код заказчика
PARTICIPATION_BG:productDealState.customer.ikz	Идентификационный код закупки
PARTICIPATION_BG:productDealState.customer.inn	ИНН заказчика
PARTICIPATION_BG:productDealState.customer.kpp	КПП заказчика
PARTICIPATION_BG:productDealState.customer.name	Наименование заказчика
PARTICIPATION_BG:productDealState.customer.ogrn	ОГРН заказчика
PARTICIPATION_BG:productDealState.customer.oktmo	ОКТМО заказчика
PARTICIPATION_BG:productDealState.customer.postAddress	Адрес заказчика (юридический/почтовый)
PARTICIPATION_BG:productDealState.customer.region	Регион заказчика
PARTICIPATION_BG:productDealState.customer.regNum	Код по СПЗ
PARTICIPATION_BG:productDealState.customer.siteUrl	Адрес сайта заказчика
PARTICIPATION_BG:productDealState.deadlineForGrant	Крайний срок получения БГ на участие
PARTICIPATION_BG:productDealState.deliveryIsRequired	Требуется доставка оригиналов документов?
PARTICIPATION_BG:productDealState.endDate	Дата окончания действия БГ на участие

id/мнемокод поля	Описание поля
PARTICIPATION_BG:productDealState.hasLoss	Наличие убытков
PARTICIPATION_BG:productDealState.isIndisputableWriteOff	Контрактом предусмотрено беспорное списание?
PARTICIPATION_BG:productDealState.isNeedPerformanceBg	Потребуется БГ на исполнение контракта?
PARTICIPATION_BG:productDealState.isPrepayment	В контракте оговорен аванс?
PARTICIPATION_BG:productDealState.loanCommitments	Сумма текущих кредитных обязательств (во всех кредитных организациях) и займов
PARTICIPATION_BG:productDealState.performanceBg	Сведения о БГ на исполнение
PARTICIPATION_BG:productDealState.performanceBg.amount	Сумма БГ на исполнение
PARTICIPATION_BG:productDealState.performanceBg.deadlineForGrant	Крайний срок получения БГ на исполнение
PARTICIPATION_BG:productDealState.performanceBg.endDate	Дата окончания действия БГ на исполнение
PARTICIPATION_BG:productDealState.performanceBg.startDate	Дата начала действия БГ на исполнение
PARTICIPATION_BG:productDealState.performanceBg.termDays	Срок БГ на исполнение (дней)
PARTICIPATION_BG:productDealState.prepaymentAmount	Сумма аванса
PARTICIPATION_BG:productDealState.prepaymentPercent	Размер аванса (%)
PARTICIPATION_BG:productDealState.startDate	Дата начала действия БГ на участие
PARTICIPATION_BG:productDealState.tender	Сведения о процедуре (тендере)
PARTICIPATION_BG:productDealState.tender.closedLaw	Тендер по 275-ФЗ (исполнение гособоронзаказа)?
PARTICIPATION_BG:productDealState.tender.closedRegister	Закрытый реестр размещения гарантии?

id/мнемокод поля	Описание поля
PARTICIPATION_BG:productDealState.tender.currencyType	Валюта контракта
PARTICIPATION_BG:productDealState.tender.enforceAmount	Размер обеспечения исполнения контракта
PARTICIPATION_BG:productDealState.tender.federalLaw	Федеральный закон, регулирующий торги
PARTICIPATION_BG:productDealState.tender.initialContractPrice	Начальная (максимальная) цена контракта
PARTICIPATION_BG:productDealState.tender.isMultiCustomer	Процедура (тендер) с несколькими заказчиками
PARTICIPATION_BG:productDealState.tender.isMultiLot	Процедура (тендер) с несколькими лотами
PARTICIPATION_BG:productDealState.tender.lotName	Предмет лота
PARTICIPATION_BG:productDealState.tender.lotNumber	Номер лота
PARTICIPATION_BG:productDealState.tender.name	Предмет процедуры (тендера)
PARTICIPATION_BG:productDealState.tender.okpds	ОКПД (список кодов)
PARTICIPATION_BG:productDealState.tender.okveds	ОКВЭД (список кодов)
PARTICIPATION_BG:productDealState.tender.placementType	Способ размещения заказа
PARTICIPATION_BG:productDealState.tender.publishDateTime	Дата и время публикации процедуры (тендера)
PARTICIPATION_BG:productDealState.tender.registryNumber	Реестровый номер процедуры (тендера)
PARTICIPATION_BG:productDealState.tender.url	Ссылка на процедуру (тендер)
PARTICIPATION_BG:productDealState.termDays	Срок БГ на участие (дней)
SMALL_BUSINESS_LOAN:productDealState.amount	Сумма кредита
SMALL_BUSINESS_LOAN:productDealState.deposit	Возможность предоставления залога

id/мнемокод поля	Описание поля
SMALL_BUSINESS_LOAN:productDealState.endDate	Планируемая дата возврата кредита
SMALL_BUSINESS_LOAN:productDealState.guarantee	Поручительство
SMALL_BUSINESS_LOAN:productDealState.hasLoss	Наличие убытков
SMALL_BUSINESS_LOAN:productDealState.insurance	Страхование кредита
SMALL_BUSINESS_LOAN:productDealState.loanCommitments	Сумма текущих кредитных обязательств (во всех кредитных организациях) и займов
SMALL_BUSINESS_LOAN:productDealState.loanType	Вид кредита
SMALL_BUSINESS_LOAN:productDealState.startDate	Планируемая дата выдачи кредита
SMALL_BUSINESS_LOAN:productDealState.termDays	Срок кредита (дней)
TENDER_LOAN:productDealState.amount	Сумма тендерного займа
TENDER_LOAN:productDealState.customer	Сведения о заказчике
TENDER_LOAN:productDealState.customer.bankAccountCurr	Корреспондентский счет банка заказчика
TENDER_LOAN:productDealState.customer.bankAccountMain	Номер основного расчетного счета заказчика
TENDER_LOAN:productDealState.customer.bankBik	БИК банка заказчика
TENDER_LOAN:productDealState.customer.bankName	Наименование банка заказчика
TENDER_LOAN:productDealState.customer.consolidatedRegNum	Код заказчика по сводному реестру
TENDER_LOAN:productDealState.customer.contactEmail	Контактный адрес электронной почты заказчика
TENDER_LOAN:productDealState.customer.contactFax	Контактный факс заказчика
TENDER_LOAN:productDealState.customer.contactFio	ФИО контактного лица заказчика
TENDER_LOAN:productDealState.customer.contactPhone	Контактный телефон заказчика

id/мнемокод поля	Описание поля
TENDER_LOAN:productDealState.customer.factAddress	Адрес заказчика (фактический)
TENDER_LOAN:productDealState.customer.inn	ИНН заказчика
TENDER_LOAN:productDealState.customer.kpp	КПП заказчика
TENDER_LOAN:productDealState.customer.name	Наименование заказчика
TENDER_LOAN:productDealState.customer.ogrn	ОГРН заказчика
TENDER_LOAN:productDealState.customer.oktmo	ОКТМО заказчика
TENDER_LOAN:productDealState.customer.postAddress	Адрес заказчика (юридический/почтовый)
TENDER_LOAN:productDealState.customer.region	Регион заказчика
TENDER_LOAN:productDealState.customer.regNum	Код по СПЗ
TENDER_LOAN:productDealState.customer.siteUrl	Адрес сайта заказчика
TENDER_LOAN:productDealState.deadlineForApplication	Дата окончания подачи заявок
TENDER_LOAN:productDealState.deliveryIsRequired	Требуется доставка оригиналов документов?
TENDER_LOAN:productDealState.endDate	Планируемая дата возврата тендерного займа
TENDER_LOAN:productDealState.etp	Наименование площадки
TENDER_LOAN:productDealState.etpAccount	Счет для перечисления займа
TENDER_LOAN:productDealState.etpAccount.type	Выберите тип счета для перечисления займа
TENDER_LOAN:productDealState.etpAccount[PAYMENT_ACCOUNT].actualAddress	Фактический адрес банка
TENDER_LOAN:productDealState.etpAccount[PAYMENT_ACCOUNT].bankName	Наименование банка
TENDER_LOAN:productDealState.etpAccount[PAYMENT_ACCOUNT].bank	БИК банка
TENDER_LOAN:productDealState.etpAccount[PAYMENT_ACCOUNT].corrAccount	Корреспондентский счет
TENDER_LOAN:productDealState.etpAccount[PAYMENT_ACCOUNT].inn	ИНН банка

id/мнемокод поля	Описание поля
TENDER_LOAN:productDealState.etpAccount[PAYMENT_ACCOUNT].kpp	КПП банка
TENDER_LOAN:productDealState.etpAccount[PAYMENT_ACCOUNT].legalAddress	Юридический адрес банка
TENDER_LOAN:productDealState.etpAccount[PAYMENT_ACCOUNT].paymentAccount	Расчетный счет (специальный счёт участника закупок)
TENDER_LOAN:productDealState.etpAccount[PERSONAL_ACCOUNT].bankName	Название банка
TENDER_LOAN:productDealState.etpAccount[PERSONAL_ACCOUNT].bik	Банковский идентификационный код
TENDER_LOAN:productDealState.etpAccount[PERSONAL_ACCOUNT].corrAccount	Корреспондентский счет
TENDER_LOAN:productDealState.etpAccount[PERSONAL_ACCOUNT].paymentAccount	Расчётный счёт
TENDER_LOAN:productDealState.etpAccount[PERSONAL_ACCOUNT].personalAccount	Лицевой счет участника на площадке
TENDER_LOAN:productDealState.hasLoss	Наличие убытков
TENDER_LOAN:productDealState.isIndisputableWriteOff	Контрактом предусмотрено беспорное списание?
TENDER_LOAN:productDealState.isNeedPerformanceBg	Потребуется БГ на исполнение контракта?
TENDER_LOAN:productDealState.isPrepayment	В контракте оговорен аванс?
TENDER_LOAN:productDealState.loanCommitments	Сумма текущих кредитных обязательств (во всех кредитных организациях) и займов
TENDER_LOAN:productDealState.performanceBg	Сведения о БГ на исполнение
TENDER_LOAN:productDealState.performanceBg.amount	Сумма БГ на исполнение
TENDER_LOAN:productDealState.performanceBg.deadlineForGrant	Крайний срок получения БГ на исполнение
TENDER_LOAN:productDealState.performanceBg.endDate	Дата окончания действия БГ на исполнение

id/мнемокод поля	Описание поля
TENDER_LOAN:productDealState.performanceBg.startDate	Дата начала действия БГ на исполнение
TENDER_LOAN:productDealState.performanceBg.termDays	Срок БГ на исполнение (дней)
TENDER_LOAN:productDealState.prepaymentAmount	Сумма аванса
TENDER_LOAN:productDealState.prepaymentPercent	Размер аванса (%)
TENDER_LOAN:productDealState.startDate	Планируемая дата выдачи тендерного займа
TENDER_LOAN:productDealState.tender	Сведения о процедуре (тендере)
TENDER_LOAN:productDealState.tender.currencyType	Валюта контракта
TENDER_LOAN:productDealState.tender.enforceAmount	Размер обеспечения исполнения контракта
TENDER_LOAN:productDealState.tender.federalLaw	Федеральный закон, регулирующий торги
TENDER_LOAN:productDealState.tender.initialContractPrice	Начальная (максимальная) цена контракта
TENDER_LOAN:productDealState.tender.isMultiCustomer	Процедура (тендер) с несколькими заказчиками
TENDER_LOAN:productDealState.tender.isMultiLot	Процедура (тендер) с несколькими лотами
TENDER_LOAN:productDealState.tender.lotName	Предмет лота
TENDER_LOAN:productDealState.tender.lotNumber	Номер лота
TENDER_LOAN:productDealState.tender.name	Предмет процедуры (тендера)
TENDER_LOAN:productDealState.tender.okpds	ОКПД (список кодов)
TENDER_LOAN:productDealState.tender.okveds	ОКВЭД (список кодов)
TENDER_LOAN:productDealState.tender.placementType	Способ размещения заказа
TENDER_LOAN:productDealState.tender.publishDateTime	Дата и время публикации

id/мнемокод поля	Описание поля
	процедуры (тендера)
TENDER_LOAN:productDealState.tender.registryNumber	Реестровый номер процедуры (тендера)
TENDER_LOAN:productDealState.tender.url	Ссылка на процедуру (тендер)
TENDER_LOAN:productDealState.termDays	Срок тендерного займа, дней
WARRANTY_BG:productDealState.actualContractPrice	Фактическая цена контракта
WARRANTY_BG:productDealState.amount	Сумма БГ
WARRANTY_BG:productDealState.concludedContracts	Количество всех заключенных контрактов
WARRANTY_BG:productDealState.customer	Сведения о заказчике
WARRANTY_BG:productDealState.customer.bankAccountCorr	Корреспондентский счет
WARRANTY_BG:productDealState.customer.bankAccountMain	Номер расчетного счета заказчика
WARRANTY_BG:productDealState.customer.bankBik	БИК банка заказчика
WARRANTY_BG:productDealState.customer.bankName	Наименование банка заказчика
WARRANTY_BG:productDealState.customer.consolidatedRegNum	Код заказчика по сводному реестру
WARRANTY_BG:productDealState.customer.contactEmail	Контактный адрес электронной почты заказчика
WARRANTY_BG:productDealState.customer.contactFax	Контактный факс заказчика
WARRANTY_BG:productDealState.customer.contactFio	ФИО контактного лица заказчика
WARRANTY_BG:productDealState.customer.contactPhone	Контактный телефон заказчика
WARRANTY_BG:productDealState.customer.factAddress	Адрес заказчика (фактический)
WARRANTY_BG:productDealState.customer.iku	Идентификационный код заказчика
WARRANTY_BG:productDealState.customer.ikz	Идентификационный код закупки

id/мнемокод поля	Описание поля
WARRANTY_BG:productDealState.customer.inn	ИНН заказчика
WARRANTY_BG:productDealState.customer.kpp	КПП заказчика
WARRANTY_BG:productDealState.customer.name	Наименование заказчика
WARRANTY_BG:productDealState.customer.ogrn	ОГРН заказчика
WARRANTY_BG:productDealState.customer.oktmo	ОКТМО заказчика
WARRANTY_BG:productDealState.customer.postAddress	Адрес заказчика (юридический/почтовый)
WARRANTY_BG:productDealState.customer.region	Регион заказчика
WARRANTY_BG:productDealState.customer.regNum	Код в Сводном Перечне Заказчиков
WARRANTY_BG:productDealState.customer.siteUrl	Адрес сайта заказчика
WARRANTY_BG:productDealState.deadlineForGrant	Крайний срок получения БГ
WARRANTY_BG:productDealState.deliveryIsRequired	Требуется доставка оригиналов документов?
WARRANTY_BG:productDealState.endDate	Дата окончания действия БГ
WARRANTY_BG:productDealState.hasLoss	Наличие убытков
WARRANTY_BG:productDealState.isIndisputableWriteOff	Контрактом предусмотрено беспорное списание?
WARRANTY_BG:productDealState.isPrepayment	С авансом
WARRANTY_BG:productDealState.isQuickDecision	Необходимо ускоренное принятие решения?
WARRANTY_BG:productDealState.loanCommitments	Сумма текущих кредитных обязательств (во всех кредитных организациях) и займов (руб.)
WARRANTY_BG:productDealState.prepaymentAmount	Сумма аванса
WARRANTY_BG:productDealState.prepaymentPercent	Размер аванса (%)
WARRANTY_BG:productDealState.startDate	Дата начала действия БГ

id/мнемокод поля	Описание поля
WARRANTY_BG:productDealState.tender	Сведения о процедуре (тендере)
WARRANTY_BG:productDealState.tender.closedLaw	Тендер по 275-ФЗ (исполнение гособоронзаказа)?
WARRANTY_BG:productDealState.tender.closedRegister	Закрытый реестр размещения гарантии?
WARRANTY_BG:productDealState.tender.currencyType	Валюта контракта
WARRANTY_BG:productDealState.tender.enforceAmount	Размер обеспечения исполнения контракта
WARRANTY_BG:productDealState.tender.federalLaw	Федеральный закон
WARRANTY_BG:productDealState.tender.initialContractPrice	Начальная (максимальная) цена контракта
WARRANTY_BG:productDealState.tender.isMultiCustomer	Совместные торги
WARRANTY_BG:productDealState.tender.isMultiLot	Мультилотовая процедура (тендер)
WARRANTY_BG:productDealState.tender.lotName	Предмет лота
WARRANTY_BG:productDealState.tender.lotNumber	Номер лота
WARRANTY_BG:productDealState.tender.name	Предмет процедуры (тендера)
WARRANTY_BG:productDealState.tender.okpds	Перечень ОКПД
WARRANTY_BG:productDealState.tender.okveds	Перечень ОКВЭД
WARRANTY_BG:productDealState.tender.placementType	Способ размещения заказа
WARRANTY_BG:productDealState.tender.publishDateTime	Дата/время публикации процедуры (тендера)
WARRANTY_BG:productDealState.tender.registryNumber	Реестровый номер
WARRANTY_BG:productDealState.tender.url	Ссылка на процедуру (тендер)
WARRANTY_BG:productDealState.tender.winnerDateTime	Дата/время протокола подведения итогов
WARRANTY_BG:productDealState.tender.winnerProtocol	Номер протокола подведения итогов

id/мнемокод поля	Описание поля
WARRANTY_BG:productDealState.termDays	Срок БГ (дней)

7.15 Поставщики сервиса: services

Системное имя сервиса	Наименование банковской/кредитной организации - поставщика сервиса
bg.uralsib.ru	ПАО «БАНК УРАЛСИБ»
bin.itfinance.io	ПАО Банк «ФК Открытие»
ca.goodfin.ru	Удостоверяющий центр
itcredit.io	IT-Credit (займы и кредиты)
mts.itfinance.io	ПАО "МТС-Банк"
pb.itfinance.io	АО "Почта Банк"
psbank.itfinance.io	ПАО "Промсвязьбанк"
rgsbank.itfinance.io	ПАО "РГС Банк"
sdm.itfinance.io	ПАО "СДМ-Банк"
ABSOLUTBANK	АКБ "Абсолют Банк" (ПАО)
AKBARS_FARZOOM	АК "Барс Банк" (ПАО)
ALFABANK	АО "АЛЬФА-БАНК"
EKSPOBANK	ООО "Экспобанк"
FAIRFIN	ООО "Феир Финанс"
LUCROS	Lucros
METIB	ПАО АКБ "Металлинвестбанк"
MODULBANK	АО КБ "Модульбанк"
MONEYFRIENDS	ООО "Мани Френдс"
MSPBANK	АО "МСП Банк"
ROSSIABANK	АО "АБ "Россия"
RUSNARBANK	АО КБ "РУСНАРБАНК"
SBERBANK	ПАО "Сбербанк"
SBIBANK	ООО "Эс-Би-Ай Банк"
SOVCOMBANK	ПАО "Совкомбанк"
TINKOFF	АО "Тинькофф Банк"
TOWNMONEY	ООО "Город Денег"
VNESHFINBANK	ООО КБ "ВНЕШФИНБАНК"

TRANSSTROJBANK_LIKEBG	АО АКБ Трансстройбанк
PROMSELXOZBANK_LIKEBG	ООО Промсельхозбанк

7.16 Типы тарифов: tariff_cell_types

Единица измерения	Описание единицы измерения
ANNUAL_PERCENT	% годовых. Указывается именно процент. Например, 4.55 , а не доля 0.0455
ONE_TIME_PERCENT	Разовый %. Указывается именно процент. Например, 4.55 , а не доля 0.0455
RUB	Рубли
TEXT	Текст

7.17 Типы расчётных счетов: types_banking_details

id/мнемокод	Описание типа расчетного счета
ACCOUNT_RUB	Расчетный счет (в рублях)
ACCOUNT_EUR	Расчетный счет (в иностранной валюте, евро)
ACCOUNT_USD	Расчетный счет (в иностранной валюте, доллары)
SUBACCOUNT_RUB	Расчетный субсчет (в рублях)
SUBACCOUNT_EUR	Расчетный субсчет (в иностранной валюте, евро)
SUBACCOUNT_USD	Расчетный субсчет (в иностранной валюте, доллары)
DEPOSIT_ACCOUNT_RUB	Счет по вкладу (депозит, в рублях)
BUDGET_ACCOUNT_INCOME_RUB	Бюджетный счет (доходный, в рублях)
BUDGET_ACCOUNT_EXPENSE_RUB	Бюджетный счет (расходный, в рублях)
TRUST_MANAGEMENT_ACCOUNT_RUB	Счет доверительного управления (в рублях)
CORR_ACCOUNT_CREDIT_ORG_RUB	Корреспондентский счет для кредитной организации (в рублях)
CORR_SUBACCOUNT_CREDIT_ORG_BRANCH_RUB	Корреспондентский субсчет для филиала кредитной организации (в рублях)

id/мнемокод	Описание типа расчетного счета
SPECIAL_ACCOUNT_BANK_AGENT_RUB	Специальный банковский счет банковского платежного агента (в рублях)
SPECIAL_ACCOUNT_BANK_SUBAGENT_RUB	Специальный банковский счет банковского платежного субагента (в рублях)
SPECIAL_ACCOUNT_AGENT_RUB	Специальный банковский счет платежного агента (в рублях)
SPECIAL_ACCOUNT_SUPPLIER_RUB	Специальный банковский счет поставщика (в рублях)
TRADING_BANK_ACCOUNT_RUB	Торговый банковский счет (в рублях)
CLEARING_BANK_ACCOUNT_RUB	Клиринговый банковский счет (в рублях)
ACCOUNT_GUARANTEE_FUND_PAYSYSTEM_RUB	Счет гарантийного фонда платежной системы (в рублях)
NOMINAL_ACCOUNT_RUB	Номинальный счет (в рублях)
ESCROW_ACCOUNT_RUB	Счет эскроу (в рублях)
COLLATERAL_ACCOUNT_RUB	Залоговый счет (в рублях)
SPECIAL_BANK_ACCOUNT_DEBTOR_RUB	Специальный банковский счет должника (в рублях)
DEPOSIT_ACCOUNT_COURT_RUB	Депозитный счет суда (в рублях)
DEPOSIT_ACCOUNT_BAILIFF	Депозитный счет подразделения службы судебных приставов (в рублях)
DEPOSIT_ACCOUNT_LAW_ENFORCEMENT	Депозитный счет правоохранительного органа (в рублях)
DEPOSIT_ACCOUNT_NOTARY	Депозитный счет нотариуса (в рублях)

7.18 Пользовательские действия: userActions

Используется для настройки статусной модели подключаемого сервиса. На текущий момент поддерживаются такие действия:

- CLIENT_EXTERNAL_SIGN (ссылка для перехода в сервис по определенному URL клиента).
- AGENT_GET_EXTERNAL_SIGN_URL (ссылка для передачи клиенту его агентом, оформившим заявку).
- CLIENT_INTERNAL_SIGN (подписание заявок одной кнопкой).
- AGENT_GET_INTERNAL_SIGN_URL (ссылка для подписания заявок клиентом одной кнопкой).

- CORRECT_REQUEST (ссылка для внесения исправлений в заявку в сервисе).
- GO_TO_SERVICE (ссылка для перехода в сервис).

APPLICATION_ID - идентификатор заявки в сервисе, которая указывается в составе URL. Должен быть передан в Goodfin в составе сообщения при получении заявки (см. пример в [ШАГ 4.4. Отправка сообщения о присвоении номера заявке в сервисе](#)).

EXT_APPLICATION_NUMBER - идентификатор, который приходит в сообщении от сервиса (тип сообщения: UpdateApplicationParams атрибут сообщения: applicationNumber).

Примеры объектов в составе userActions (по требованиям поставщика сервиса возможно доработка и расширение поддерживаемых действий)	Описание назначения
<pre>{ "actionType": "CLIENT_EXTERNAL_SIGN", "label": "Принять", "description": "Перейти по ссылке для принятия предложения" "template": "https://exampleService.com/указание на страницу с принятием предложения/{EXT_APPLICATION_NUMBER}?idp=системное имя банка" }</pre>	<p>Показан пример, который будет выводить клиенту кнопку "Принять" с всплывающей подсказкой "Перейт и по ссылке для принятия предложения". При нажатии на кнопку клиент перейдет по ссылке, сгенерированной на основе значения поля template.</p>
<pre>{ "actionType": "AGENT_GET_EXTERNAL_SIGN_URL", "label": "Получить ссылку", "description": "Получить ссылку для принятия предложения" "template": "https://exampleService.com/указание на страницу для передачи клиенту с предложением без необходимости авторизации/{APPLICATION_ID}" }</pre>	<p>Показан пример, который будет выводить агенту кнопку "Получить ссылку" с всплывающей подсказкой "Получить ссылку для принятия предложения". При нажатии на кнопку агент увидит модальное окно, в котором будет показана ссылка, сгенерированная на основе значения поля template. Агент сможет скопировать ссылку в буфер и далее передать клиенту любым удобным им способом.</p>
<pre>{ "actionType": "CLIENT_INTERNAL_SIGN", "label": "Подписать", "description": "Подписать заявку" "documentsToSignUrlTemplate": "http://указать страницу с</pre>	<p>Показан пример, который будет выводить клиенту кнопку "Подписать все ({количество заявок})". При</p>

Примеры объектов в составе userActions (по требованиям поставщика сервиса возможно доработка и расширение поддерживаемых действий)	Описание назначения
<pre>документами на подписание}" }</pre>	<p>нажатии на кнопку клиент перейдёт на страницу Подписания всех заявок в статусе "Готова к подписанию" средствами Goodfin. Результат подписания будет передан в банковские сервисы, с которыми выполнена интеграция по подписанию заявок на стороне Goodfin. Примечание: На текущий момент работает для банковских сервисов на базе платформ IT-Finance, ITCredit.</p>
<pre>{ "actionType": "AGENT_GET_INTERNAL_SIGN_URL", "label": "Получить ссылку", "description": "Получить ссылку для подписания клиентом" }</pre>	<p>Показан пример, который будет выводить агенту кнопку "Подписать все ()". При нажатии на кнопку агент скопирует ссылку для отправки клиенту на подписание всех заявок в статусе "Готова к подписанию". Агент может отправить ссылку клиенту любым удобным способом.</p> <p>Клиент по полученной ссылке перейдёт на страницу Подписания всех заявок в статусе "Готова к подписанию" средствами Goodfin. Результат подписания будет передан в</p>

Примеры объектов в составе userActions (по требованиям поставщика сервиса возможно доработка и расширение поддерживаемых действий)	Описание назначения
	<p>банковские сервисы, с которыми выполнена интеграция по подписанию заявок на стороне Goodfin. Примечание: На текущий момент работает для банковских сервисов на базе платформ IT-Finance</p>
<pre>{ actionType: "CORRECT_REQUEST" label: "Исправить заявку в сервисе" description: "Перейти по ссылке для внесения исправлений в заявку в сервисе" template: "https://exampleService.com/указание на страницу для исправления заявки{APPLICATION_ID}/" }</pre>	<p>Если во внешнем сервисе есть возможность исправить заявку, то можно показать пользователю действие "Исправить заявку". Показан пример, который будет выводить пользователю кнопку "Исправить заявку в сервисе" с всплывающей подсказкой "Перейт и по ссылке для внесения исправлений в заявку в сервисе". При нажатии на кнопку пользователь перейдет по указанному URL для редактирования заявки в сервисе.</p> <p>Если нет специального URL, который сразу переведет пользователя на редактирование заявки в сервисе, то можно указать тот, который переведет на страницу сервиса с кнопкой "Внести изменения"</p>

Примеры объектов в составе userActions (по требованиям поставщика сервиса возможно доработка и расширение поддерживаемых действий)	Описание назначения
	внутри самого сервиса.
<pre>{ actionType: "GO_TO_SERVICE" label: "Перейти в сервис ..." description: "Перейти по ссылке в сервис ..." template: "https://.../bidders/requests/{EXT_APPLICATION_NUMBER}?idp=системное имя банка" }</pre>	<p>Показан пример, который будет выводить текст со ссылкой для перехода в сервис. При нажатии на ссылку, сгенерированной на основе значения поля template, пользователь перейдет в свой профиль в данном сервисе.</p>

7.19 Типы валидаторов: validators

Описание валидатора	Валидатор
Одно из физ. лиц компании должно иметь роль "Руководитель/директор"	type: "COMPANY_PERSON_ROLE_REQUIRED" role: "DIRECTOR"
Одно из физ. лиц компании должно иметь роль "Главный бухгалтер"	type: "COMPANY_PERSON_ROLE_REQUIRED" role: "BOOKER"
Должна присутствовать информация о соучредителях	type: "COMPANY_FOUNDER_REQUIRED"
Поле "Дата окончания полномочий" является обязательным для роли "Является единоличным исполнительным органом"	type: "DIRECTOR_ENDDATE_REQUIRED"
Для паспорта РФ указать "Код подразделения"	type: "PASSPORT_CODE_POD_REQUIRED"
Обязательный тип документа в карточке клиента должен	type: "COMPANY_REQUIRED_DOC_FILE_TYPE" fileTypes:

Описание валидатора	Валидатор
содержать только те типы файлов, которые указаны в параметрах валидатора при настройке продукта.	- "pdf"
Обязательный тип документа в карточке клиента должен содержать количество файлов, которое меньше и равно максимальному количеству, указанному в параметрах валидатора при настройке продукта.	type: "COMPANY_REQUIRED_DOC_FILE_COUNT" maxFilesCount: 2
Обязательный тип документа по предмету сделки должен содержать только те типы файлов, которые указаны в параметрах валидатора при настройке продукта.	type: "PRODUCT_REQUIRED_DOC_FILE_TYPE" fileTypes: - "doc" - "xlsx"
Обязательный тип документа по предмету сделки должен содержать количество файлов, которое меньше и равно максимальному количеству, указанному в параметрах валидатора при настройке продукта.	type: "PRODUCT_REQUIRED_DOC_FILE_COUNT" maxFilesCount: 1
Обязательно автоматическое заполнение итогов финансовых показателей Убран в версии 1.23	type: "COMPANY_REQUIRED_CALCULATED_INDICATOR_NOT_REWRITED"

7.20 Правила применимости: productApplicabilityRule, applicabilityRule

7.20.1 Оглавление

- [1 Примеры скриптов правил применимости](#)
- [2 Примеры получения значений полей по данным заявки для использования в правилах применимости](#)

Примеры правил применимости, которые можно использовать в правиле применимости продукта в целом или для пакетов документов в составе продукта.

Вы можете по аналогии написать собственные правила.

Примеры сформированных тестовых заявок можно посмотреть и скачать в документе [1. Подготовительные работы по интеграции с применением API](#)

7.20.2 1 Примеры скриптов правил применимости

Описание правила	Пример правила применимости, script
<p>для ИП (пример синтаксиса действует с версии 1.9.0)</p>	<pre data-bbox="544 1021 1326 1368">function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); if (clientCompany == null clientCompany.legalType == null) { return null; } return clientCompany.legalType.id == 'IP'; }</pre> <p data-bbox="504 1458 1361 1518">где для указания значения <code>clientCompany.legalType.id</code> используется справочник Формы предпринимательской деятельности: legal types</p>
<p>ЮЛ с суммой гарантии больше 10 млн (пример синтаксиса действует с версии 1.9.0)</p>	<pre data-bbox="544 1570 1326 1944">function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); if (clientCompany == null clientCompany.legalType == null dealApplicationData.result.productDealState.amount == null) { return null; } return clientCompany.legalType.id == 'ORGANIZATION' &&</pre>

Описание правила	Пример правила применимости, script
	<pre>dealApplicationData.result.productDealState.amount > 10000000; }</pre>
<p>ЮЛ с фактической ценой контракта до 100 тысяч</p> <p>(пример синтаксиса действует с версии 1.9.0)</p>	<pre>function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); if (clientCompany == null clientCompany.legalType == null dealApplicationData.result.productDealState.actualContr actPrice == null) { return null; } return clientCompany.legalType.id == 'ORGANIZATION' && dealApplicationData.result.productDealState.actualContr actPrice < 100000; }</pre>
<p>ЮЛ со сроком гарантии меньше 30 дней</p> <p>(пример синтаксиса действует с версии 1.9.0)</p>	<pre>function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); if (clientCompany == null clientCompany.legalType == null dealApplicationData.result.productDealState.termDays == null) { return null; } return clientCompany.legalType.id == 'ORGANIZATION' && dealApplicationData.result.productDealState.termDays < 30; }</pre>

Описание правила	Пример правила применимости, script
<p>ЮЛ, финансовая (FIN) или некоммерческая организации (UNCOM)</p> <p>(пример синтаксиса действует с версии 1.9.0)</p>	<pre data-bbox="544 277 1326 763"> function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); var clientOrgTypeId = clientCompany.orgTypeId; var orgTypes = ['FIN', 'UNCOM']; if(clientCompany == null clientOrgTypeId == null clientCompany.legalType == null) { return null; } return orgTypes.indexOf(clientOrgTypeId) !== -1 && clientCompany.legalType.id == 'ORGANIZATION'; } </pre> <p data-bbox="504 853 1270 936">где orgTypes может быть 'COM' - коммерческая организация, 'UNCOM' - некоммерческая организация, 'FIN' - финансовая организация</p>
<p>ЮЛ, регион заказчика Камчатский край или Чукотский АО</p> <p>(пример синтаксиса действует с версии 1.9.0)</p>	<pre data-bbox="544 994 1342 1532"> function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); var customerRegion = dealApplicationData.result.productDealState.customer.re gion.id; var customerRegions = ['Камчатский край', 'Чукотский АО']; if (clientCompany == null clientCompany.legalType == null customerRegion == null) { return null; } return clientCompany.legalType.id == 'ORGANIZATION' && customerRegions.indexOf(customerRegion) !== -1; } </pre>

Описание правила	Пример правила применимости, script
<p>ИП с типом налогообложения УСН, Сельхоз, ЕНВД</p> <p>(пример синтаксиса действует с версии 1.9.0)</p>	<pre>function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); if (clientCompany == null) { return null; } var taxSystemDto = dealApplicationData.getFetchField('TaxSystemCatalog', clientCompany.taxSystem); if (clientCompany.legalType == null taxSystemDto == null) { return null; } var taxSystems = ['USN', 'ENVD', 'ESHN']; return taxSystems.indexOf(taxSystemDto.code) !== -1 && clientCompany.legalType.id == 'IP'; }</pre>
<p>ЮЛ, дата государственной регистрации после 2003-06-05</p> <p>(пример синтаксиса действует с версии 1.9.0)</p>	<pre>function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); if (clientCompany == null clientCompany.legalType == null clientCompany.regDate == null) { return null; } return clientCompany.legalType.id == 'ORGANIZATION' && clientCompany.regDate > '2003-06-05'; }</pre>
<p>ЮЛ, Сумма БГ до 2 млн.руб. - срок деятельности клиента от 180 дней. Сумма БГ от 2 млн.руб. срок деятельности клиента - от 365 дней.</p> <p>(пример синтаксиса действует с версии 1.9.0)</p>	<pre>function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); if (clientCompany == null clientCompany.legalType == null clientCompany.regDate == null dealApplicationData.result.productDealState.amount == null) { return null; } now = new Date(); rD = new Date(clientCompany.regDate); termClient = (now - rD)/(60 * 60 * 24 * 1000); return clientCompany.legalType.id == 'ORGANIZATION' && ((termClient >= 180 && dealApplicationData.result.productDealState.amount <= 2000000) (termClient >= 365 &&</pre>

Описание правила	Пример правила применимости, script
	<pre>dealApplicationData.result.productDealState.amount > 2000000)); }</pre>
<p>ЮЛ, базовый ОКВЭД 86.23 - Стоматологическая практика или 86.90.9 - Деятельность в области медицины прочая, не включенная в другие группировки</p> <p>(пример синтаксиса действует с версии 1.9.0)</p>	<pre>function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); if (clientCompany == null) { return null; } var clientBaseOkved = dealApplicationData.getFetchField('OkvedCatalog', clientCompany.baseOkved); if (clientCompany.legalType == null clientBaseOkved == null) { return null; } var baseOkveds = ['86.23', '86.90.9']; return baseOkveds.indexOf(clientBaseOkved.code) !== -1 && clientCompany.legalType.id == 'ORGANIZATION'; }</pre>
<p>ЮЛ с организационно-правовыми формами АО (код 12200), ПАО (код 12247), НАО (код 12267)</p> <p>(пример синтаксиса действует с версии 1.9.0)</p>	<pre>function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); if (clientCompany == null) { return null; } var okopfDto = dealApplicationData.getFetchField('OkopfCatalog', clientCompany.okopf); if(okopfDto == null clientCompany.legalType == null) { return null; } var okopfTypes = ['12200', '12247', '12267']; return clientCompany.legalType.id == 'ORGANIZATION' && okopfTypes.indexOf(okopfDto.code) !== -1; }</pre>

Описание правила	Пример правила применимости, script
<p>ЮЛ, 223-ФЗ или 44-ФЗ</p> <p>(пример синтаксиса действует с версии 1.9.0)</p>	<pre data-bbox="544 277 1342 763">function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); var federalLaw = dealApplicationData.result.productDealState.tender.fede ralLaw.id; if (clientCompany == null clientCompany.legalType == null federalLaw == null) { return null; } var federalLaws = ['223FL', '44FL']; return federalLaws.indexOf(federalLaw) != -1 && clientCompany.legalType.id == 'ORGANIZATION'; }</pre> <p data-bbox="504 853 1166 909">где для указания значения federalLaws используется справочник Федеральные законы: federalLaw types</p>
<p>Продукт применим только для головных компаний (ЮЛ), филиалы исключаются</p> <p>(пример синтаксиса действует с версии 1.9.0)</p>	<pre data-bbox="544 960 1342 1637">function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); if(clientCompany == null) { return null; } var okopfDto = dealApplicationData.getFetchField('OkopfCatalog', clientCompany.okopf); if(okopfDto == null clientCompany.legalType == null) { return null; } var sourceInfo = clientCompany.sourceInfo; var okopfTypes = ['30000', '30001', '30002', '30003', '30004', '30005', '30006', '30008']; return clientCompany.legalType.id == 'ORGANIZATION' && okopfTypes.indexOf(okopfDto.code) === -1 && sourceInfo.headCompany === true && sourceInfo.hasBranches === false; }</pre>

Описание правила	Пример правила применимости, script
<p>ЮЛ, регион фактического местоположения компании Камчатский край или Тульская область</p> <p>(пример синтаксиса действует с версии 1.14.0)</p>	<pre>function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); if (clientCompany == null) { return null; } var factAddressDto = dealApplicationData.getFetchField('CompanyAddress', clientCompany.factAddress); if (clientCompany.legalType == null factAddressDto == null) { return null; } var companyRegions = ['Камчатский край', 'Tульская область']; return companyRegions.indexOf(factAddressDto.region) !== -1 && clientCompany.legalType.id == 'ORGANIZATION'; }</pre>
<p>ЮЛ, стоп-регионы по юридическому адресу: Адыгея, Республика Ингушетия, Чеченская республика, Республика Дагестан, республика Крым, Севастополь</p> <p>(пример синтаксиса действует с версии 1.14.0)</p>	<pre>function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); var LegalAddress = dealApplicationData.getFetchField('CompanyAddress', clientCompany.legalAddress); if (clientCompany == null clientCompany.legalType == null LegalAddress.region == null) { return null; } var regex = new RegExp('адыгея ингушет чечен дагестан крым севастоп', 'i'); var clientLegalReg = regex.test(LegalAddress.region); return !clientLegalReg && clientCompany.legalType.id == 'ORGANIZATION'; }</pre>

Описание правила	Пример правила применимости, script
<p>ЮЛ, среди собственников есть юр. лица</p> <p>(пример синтаксиса действует с версии 1.14.0)</p>	<pre>function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); if (clientCompany == null clientCompany.legalType == null) { return null; } var legalFounders = dealApplicationData.getFetchFieldAsMap('LegalFounder'); var keyArray = legalFounders.keySet().toArray(); if(keyArray.length == 0) { return false; } return clientCompany.legalType.id == 'ORGANIZATION'; }</pre>
<p>ЮЛ, среди собственников только физ. лица</p> <p>(пример синтаксиса действует с версии 1.14.0)</p>	<pre>function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); if (clientCompany == null clientCompany.legalType == null) { return null; } var legalFounders = dealApplicationData.getFetchFieldAsMap('LegalFounder'); var keyArray = legalFounders.keySet().toArray(); if(keyArray.length != 0) { return false; } return clientCompany.legalType.id == 'ORGANIZATION'; }</pre>

Описание правила	Пример правила применимости, script
<p>ЮЛ, запрет тендеров по 275-ФЗ</p> <p>(пример синтаксиса действует с версии 1.16.0)</p>	<pre>function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); var tenderClosedLaw = dealApplicationData.result.productDealState.tender.clos edLaw; if (clientCompany == null clientCompany.legalType == null) { return null; } if (tenderClosedLaw === true) { return false; } return clientCompany.legalType.id == 'ORGANIZATION'; }</pre>
<p>ЮЛ, валюта контракта - рубли</p> <p>(пример синтаксиса действует с версии 1.9.0)</p>	<pre>function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); var tenderCurrencyType = dealApplicationData.result.productDealState.tender.curr encyType; var currencyTypeDto = dealApplicationData.getFetchField('CurrencyTypeCatalog' , tenderCurrencyType); if (clientCompany == null clientCompany.legalType == null) { return null; } var currencyTypes = ['RUB']; return clientCompany.legalType.id == 'ORGANIZATION' && currencyTypes.indexOf(currencyTypeDto.code) !== -1; }</pre>

Описание правила	Пример правила применимости, script
<p>ИП, БГ(У), где сумма БГ на исполнение не превышает 10 млн.</p> <p>(пример синтаксиса действует с версии 1.16.0)</p>	<pre>function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); var isNeedPerformanceBgDeal = dealApplicationData.result.productDealState.isNeedPerfo rmanceBg; if (isNeedPerformanceBgDeal === true) { var amountBG = dealApplicationData.result.productDealState.performance Bg.amount; return amountBG <= 10000000; } if (clientCompany == null clientCompany.legalType == null) { return null; } return clientCompany.legalType.id == 'IP'; }</pre>
<p>Не заполнены фин. показатели.</p> <p>(пример синтаксиса действует с версии 1.20.0)</p> <p>Ниже приведен аналог проверки заполнения фин. показателей для версии 1.19.0</p>	<pre>function isAvailableForDealApplicationData (dealApplicationData) { return !isFinanceIndicatorFilledForAllPeriods (dealApplicationD ata, 'FORM_1', '1600') !isFinanceIndicatorFilledForAllPeriods (dealApplicationD ata, 'FORM_1', '1700') !isFinanceIndicatorFilledForAllPeriods (dealApplicationD ata, 'FORM_2', '2110'); }</pre>

Сумма БГ меньше 1млн И не заполнены фин. показатели.

(пример синтаксиса действует с версии 1.19.0)

```
function
isAvailableForDealApplicationData (dealApplicationData)
{
    if
    (dealApplicationData.result.productDealState.amount >
1000000) {
        return false;
    }
    if
    (dealApplicationData.getFetchFieldAsMap('FinanceIndicator') == null) {
        return true;
    }
    var allIndicators =
dealApplicationData.getFetchFieldAsMap('FinanceIndicator').values();
    var idOf1600 = allIndicators.stream()
        .filter(function(e) {
            return '1600' == e.code
        })
        .findFirst()
        .map(function(e){ return e.id})
        .orElse(null);
    var idOf1700 = allIndicators.stream()
        .filter(function(e) {
            return '1700' == e.code
        })
        .findFirst()
        .map(function(e){ return e.id})
        .orElse(null);
    var idOf2110 = allIndicators.stream()
        .filter(function(e) {
            return '2110' == e.code
        })
        .findFirst()
        .map(function(e){ return e.id})
        .orElse(null);
    if (idOf1600 == null || idOf1700 == null ||
idOf2110 == null) {
        return true;
    }
    var form1Values =
dealApplicationData.result.financeForms.stream()
        .filter(function(e) {
            return '1b010932-3e5b-4992-861e-
b3648f83a03d' == e.formId
        })
        .findFirst()
        .get();
    var form2Values =
dealApplicationData.result.financeForms.stream()
        .filter(function(e) {
            return 'a48142d6-ded8-41c2-bfbf-
57035a4fc773' == e.formId
        })
        .findFirst()
        .get();
    if (form1Values == null || form2Values == null)
    {
        return true;
    }
    for (var j = 0; j < form1Values.periods.size();
j++) {
```

Описание правила	Пример правила применимости, script
	<pre> if (form1Values.values[form1Values.getIndicators().indexOf(idOf1600)][j] == null form1Values.values[form1Values.getIndicators().indexOf(idOf1700)][j] == null) { return true; } for (var j = 0; j < form2Values.periods.size(); j++) { if (form2Values.values[form2Values.getIndicators().indexOf(idOf2110)][j] == null) { return true; } } return false; } </pre>
<p>ЮЛ, если все физ. лица компании с документом "Паспорт гражданина РФ"</p> <p>(пример синтаксиса действует с версии 1.19.0)</p>	<pre> function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); var allPersonsHasPassport = true; var personsMap = dealApplicationData.getFetchFieldAsMap('Person'); if (personsMap !== null) { var persons = personsMap.values(); allPersonsHasPassport = persons.stream().allMatch(function (person) { return person.docType != null && person.docType.id == 'PASSPORT'; }); } return clientCompany.legalType.id == 'ORGANIZATION' && allPersonsHasPassport; } </pre>

Описание правила	Пример правила применимости, script
<p>ЮЛ, если прикреплены документы: "Бухгалтерская (финансовая) отчетность на последнюю календарную квартальную дату" И "Годовая бухгалтерская (финансовая) отчетность"</p> <p>(пример синтаксиса действует с версии 1.19.0)</p>	<pre> function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); if (clientCompany == null) { return null; } // Для проверки каждого документа требуются: // 1. переменная, хранящая признак прикрепления документа //----- var financeReportExists = false; //----- var analyticalBalanceExists = false; var documentsMap = dealApplicationData.getFetchFieldAsMap('ApplicationDocu ment'); if (documentsMap !== null) { var documents = documentsMap.values(); // 2. осуществить поиск прикрепленного документа (не архивного) нужного типа; записать результат (true/false) в переменную //----- financeReportExists = documents.stream().anyMatch(function (doc) { return doc.docType != null && doc.docType.id == 'FINANCE_REPORT_LAST_QUARTER' && doc.docState != null && doc.docState.id == 'ACTUAL'; }); //----- analyticalBalanceExists = documents.stream().anyMatch(function (doc) { return doc.docType != null && doc.docType.id == 'ANALYTICAL_BALANCE_LAST_REPORTING_YEAR' && doc.docState != null && doc.docState.id == 'ACTUAL'; }); } // 3. Если требуются все документы, то в return указать "! (признак_прикрепления_документа_1 && ... && признак_прикрепления_документа_N)" // Если требуется прикрепить один из документов, то в return указать "! (признак_прикрепления_документа_1 ... признак_прикрепления_документа_N)" //----- return clientCompany.legalType.id == 'ORGANIZATION' && !(financeReportExists && analyticalBalanceExists); //----- } </pre>

Описание правила	Пример правила применимости, script
<p>ЮЛ, стоп по ИНН заказчика: 1660049283, 1653014843.</p> <p>(пример синтаксиса действует с версии 1.6.0)</p>	<pre>function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); var customerInn = dealApplicationData.result.productDealState.customer.inn; if (clientCompany == null clientCompany.legalType == null) { return null; } var stopCustomerInn = ['1660049283', '1653014843']; return clientCompany.legalType.id == 'ORGANIZATION' && stopCustomerInn.indexOf(customerInn) == -1; }</pre>
<p>ЮЛ, стоп по ОКПД</p> <p>(пример синтаксиса действует с версии 1.19.0)</p>	<pre>function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); var okpdMap = dealApplicationData.getFetchFieldAsMap('OkpdCatalog'); var okpdIsValid = true; if (okpdMap != null) { var okpds = okpdMap.values(); okpdIsValid = okpds.stream().allMatch(function(okpd) { return okpd.code != null && okpd.code != '42.11.20.000'; }); } return clientCompany.legalType.id == 'ORGANIZATION' && okpdIsValid; }</pre>

Описание правила	Пример правила применимости, script
ИП, стоп по группам ОКПД (пример синтаксиса действует с версии 1.19.0)	<pre> function isAvailableForDealApplicationData (dealApplicationData) { var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); var okpdMap = dealApplicationData.getFetchFieldAsMap('OkpdCatalog'); var okpdIsValid = true; var regexOkpd = new RegExp('^ (06 19 41.2 42 43.99 64 65 66 68 99)', 'i'); if (okpdMap !== null) { var okpds = okpdMap.values(); okpdIsValid = okpds.stream().allMatch(function (okpd) { var legalOkpd = regexOkpd.test(okpd.code); return okpd.code !== null && !legalOkpd; }); } return clientCompany.legalType.id == 'IP' && okpdIsValid; } </pre>

7.20.3 2 Примеры получения значений полей по данным заявки для использования в правилах применимости

Поле согласно структуре заявки с учетом вложенности	Описание	Как объявить и получить значение в java
result.client	Объект компания клиента/ИП	<pre> var clientCompany = dealApplicationData.getFetchField('Comp any', dealApplicationData.result.client.id); </pre>
result.Company.legalType	Категория клиента (организация, ИП, физ. лицо)	<pre> var clientCompany = dealApplicationData.getFetchField('Comp any', dealApplicationData.result.client.id); var companyLegalType = clientCompany.legalType.id; </pre>
result.productDealState.amount	Сумма продукта	<pre> var amountProductDealState = dealApplicationData.result.productDeals tate.amount; </pre>
result.productDealState.actualContractPrice	Сумма ФЦК продукта	<pre> var priceProductDealState = dealApplicationData.result.productDeals tate.actualContractPrice; </pre>

Поле согласно структуре заявки с учетом вложенности	Описание	Как объявить и получить значение в java
result.productDealState.termDays	Срок гарантии	<pre>var termDaysProductDealState = dealApplicationData.result.productDealState.termDays;</pre>
result.Company.orgType	Тип организации	<pre>var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); var clientOrgTypeId = clientCompany.orgType.id;</pre>
result.productDealState.customer.region	Местонахождение (регион) заказчика	<pre>var customerRegion = dealApplicationData.result.productDealState.customer.region.id;</pre>
result.Company.taxSystem	Система налогообложения	<pre>var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); var taxSystemDto = dealApplicationData.getFetchField('TaxSystemCatalog', clientCompany.taxSystem);</pre>
result.Company.regDate	Дата государственной регистрации	<pre>var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); var companyRegDate = clientCompany.regDate;</pre>
result.Company.baseOkved	Базовый ОКВЭД	<pre>var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); var clientBaseOkved = dealApplicationData.getFetchField('OkvedCatalog', clientCompany.baseOkved);</pre>
result.Company.okopf	Организационно-правовая форма	<pre>var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); var okopfDto = dealApplicationData.getFetchField('OkopfCatalog', clientCompany.okopf);</pre>
result.productDealState.tender.federalLaw	Федеральный закон	<pre>var federalLaw = dealApplicationData.result.productDealState.tender.federalLaw.id;</pre>

Поле согласно структуре заявки с учетом вложенности	Описание	Как объявить и получить значение в java
fetchFields.CompanySourceInfo	Головная компания	<pre>var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); var sourceInfo = clientCompany.sourceInfo;</pre>
fetchFields.CompanyAddress.region	Регион фактического местоположения компании	<pre>var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); var factAddressDto = dealApplicationData.getFetchField('CompanyAddress', clientCompany.factAddress);</pre>
fetchFields.CompanyAddress.region	Регион юридического адреса компании	<pre>var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); var legalAddress = dealApplicationData.getFetchField('CompanyAddress', clientCompany.legalAddress);</pre>
fetchFields.LegalFounder	Собственники (юр лицо)	<pre>var legalFounders = dealApplicationData.getFetchFieldAsMap('LegalFounder'); var keyArray = legalFounders.keySet().toArray();</pre>
result.Company.regDate	Срок деятельности клиента	<pre>var clientCompany = dealApplicationData.getFetchField('Company', dealApplicationData.result.client.id); now = new Date(); rD = new Date(clientCompany.regDate); termClient = (now - rD)/(60 * 60 * 24 * 1000);</pre>
result.productDealState.tender.closedLaw	Тендер по 275-ФЗ	<pre>var tenderClosedLaw = dealApplicationData.result.productDealState.tender.closedLaw;</pre>
result.productDealState.tender.currencyType	Валюта контракта	<pre>var tenderCurrencyType = dealApplicationData.result.productDealState.tender.currencyType; var currencyTypeDto = dealApplicationData.getFetchField('CurrencyTypeCatalog', tenderCurrencyType);</pre>

Поле согласно структуре заявки с учетом вложенности	Описание	Как объявить и получить значение в java
result.productDealState.performanceBg.amount	Сумма БГ на исполнении вне типа продукта БГ(И)	<pre>var amountBG = dealApplicationData.result.productDealState.performanceBg.amount;</pre>
result.productDealState.isNeedPerformanceBg	Потребуется БГ на исполнении контракта?	<pre>var isNeedPerformanceBgDeal = dealApplicationData.result.productDealState.isNeedPerformanceBg;</pre>
result.productDealState.customer.inn	ИНН заказчика	<pre>var customerInn = dealApplicationData.result.productDealState.customer.inn;</pre>

8 Branches

8.1 All Branches in this Space

Name	Created	By	
There are no branches yet.			

8.2 Confirm bulk merging

[scroll-bookmark-137](#)

Do you really want to merge all selected branches?Version of pages (optional)Merge and trash
Close