**Описание функциональных характеристик, информация для установки и эксплуатации программного обеспечения «Модуль IT-Finance «Электронный документооборот»**

Оглавление

[**Функциональные характеристики программного обеспечения** 2](#_Toc76561004)

[**Установка программного обеспечения** 2](#_Toc76561005)

[**Требования к рабочему окружению** 2](#_Toc76561006)

[**Требуемые программные продукты в составе клиентской части** 2](#_Toc76561007)

[**Эксплуатация программного обеспечения** 3](#_Toc76561008)

[Руководство пользователей 3](#_Toc76561009)

1.[Наборы доп. документов 3](#_Toc76561010)

[1.1 Создание набора доп. документов 3](#_Toc76561011)

[1.2 Редактирование набора доп. документов 6](#_Toc76561012)

[1.3 Удаление набора доп. документов 6](#_Toc76561013)

2. [Пакеты документов 6](#_Toc76561014)

[2.1 Просмотр списка существующих пакетов документа 7](#_Toc76561015)

[2.2 Создание пакета документов 7](#_Toc76561016)

[2.3 Редактирование пакета документов 8](#_Toc76561017)

3. [Сценарии, алгоритмы 9](#_Toc76561018)

[3.1. Алгоритм автоматического назначения ответственного сотрудника в заявку (change\_assign\_user) 9](#_Toc76561019)

[3.2. Алгоритм сравнения сумм для правил (get\_lookup\_use\_summ\_rule) 13](#_Toc76561020)

[3.3. Алгоритм отвязки ответственного сотрудника 14](#_Toc76561021)

[3.4. Алгоритм привязки ответственного сотрудника (fasten\_request) 14](#_Toc76561022)

[3.5. Алгоритм закрепления сотрудника за компанией по ИНН (fasten\_bidder) 14](#_Toc76561023)

[4. Сценарии, алгоритмы 15](#_Toc76561024)

[4.1. Алгоритм получения кода налогового органа по данным заявки (get\_bidder\_nalog\_code) 15](#_Toc76561025)

[4.2. Основная цель(и), назначение 17](#_Toc76561026)

[4.3. Краткое описание и связь с другими историями 17](#_Toc76561027)

[5. Сценарии, алгоритмы 17](#_Toc76561028)

[5.1. Алгоритм генерации превью документов (preview-generator) 17](#_Toc76561029)

# **Функциональные характеристики программного обеспечения**

**Программное обеспечение Модуль IT-Finance «Электронный документооборот» (далее по тексту также программное обеспечение/ ПО/ Cистема)**

Программа обеспечивает автоматическую проверку документов соискателей и их комплектности.

Область применения: для организаций, использующих программу IT-Finance (свидетельство № 2018615047 от 23.04.2018).

Функциональные возможности:

Автоматическое заполнения справочных данных из различных сервисов;

Определение категории документов; Проверка комплектности документов по кредитной заявке; Настройка и определение маршрута согласования заявки, и отправка заявки на согласование; Хранение истории обработки заявки; Генерация печатных форм документов по шаблонам.

# **Установка программного обеспечения**

ПО распространяется в виде интернет-сервиса, специальные действия по установке ПО на стороне пользователя не требуются

# **Требования к рабочему окружению**

# **Требуемые программные продукты в составе клиентской части**

|  |  |  |
| --- | --- | --- |
| **Наименование** | **Версия** | **Официальный сайт продукта** |
| КриптоПро CSP | Действующие сертифицированные | <http://www.cryptopro.ru/> |
| КриптоПро ЭЦП Browser plug-in | <http://www.cryptopro.ru/> |
| **Браузеры (любой из):** |  |  |
| Firefox | Три последние официальные стабильные версии (вышедшие не позднее года на момент проверки версионности) | <https://www.mozilla.org/> |
| Google Chrome | <https://www.google.ru/chrome/> |

# **Эксплуатация программного обеспечения**

# Руководство пользователей

## 1.Наборы доп. документов

Данный справочник предназначен для создания и управления наборами типовых документов, необходимых для выдачи банковской гарантии. Функционал справочника позволяет создавать такие наборы для дальнейшего использования при подаче заявок. Количество типовых наборов не ограничено. Вы можете создать несколько наборов документов в зависимости от организационно-правовой формы соискателя гарантии (ЗАО, ООО, ИП и пр.) или, например, в зависимости от суммы гарантии.

* [1.1 Создание набора доп. документов](#scroll-bookmark-40)
* [1.2 Редактирование набора доп. документов](#scroll-bookmark-41)
* [1.3 Удаление набора доп. документов](#scroll-bookmark-42)

### 1.1 Создание набора доп. документов

Для создания нового набора типовых документов нажмите кнопку *"Создать"*в окне со списком и редактированием типовых наборов дополнительных документов (см. рис. 1.1.1)

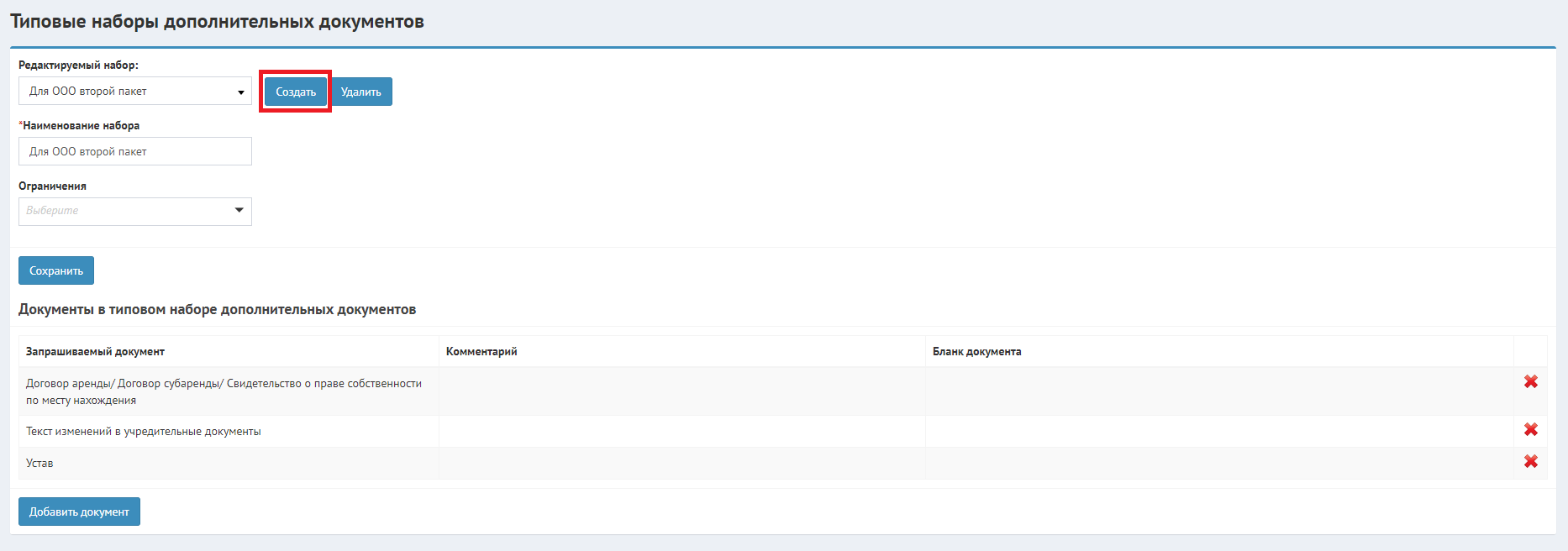


Рисунок 1.1.1 - Окно со списком и режимом редактирования наборов

Будет открыто окно создания нового набора. Для создания нового набора необходимо присвоить ему наименование и выбрать набор ограничений (из справочника *"Ограничения")*, с которыми будет доступен данный набор. После этого нажмите кнопку *"Сохранить"*(см. рис. 1.1.2)

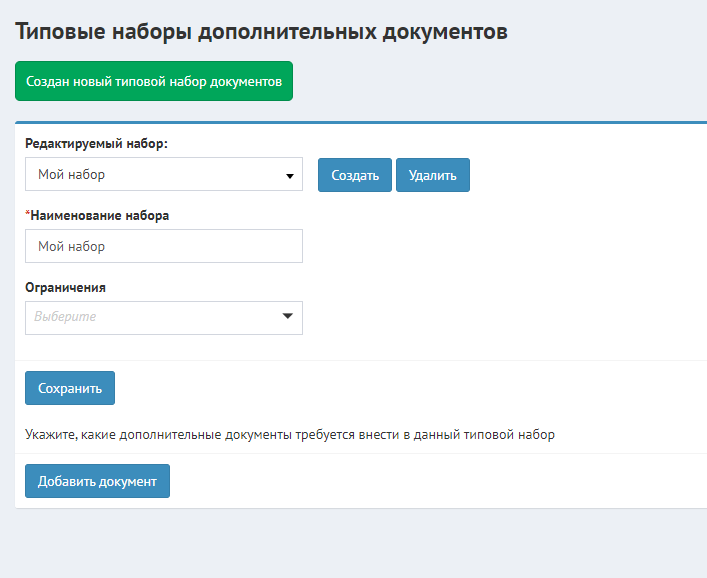


Рисунок 1.1.2 - Окно создания нового типового набора дополнительных документов

После сохранения вы увидите подтверждающее сообщение. Также станет доступна возможность *"Добавить документ"* в набор по нажатию соответствующей кнопки. Откроется окно с формой добавления нового документа, которое может иметь два представления (см. рис 1.1.3 - 1.1.4).  
***ПРИМЕЧАНИЕ:***В типовой набор может быть добавлено до 20 запрашиваемых **произвольных** документов.

* Представление *"Стандартный документ"*позволяет включить в набор один из стандартных документов, доступных из списка. Для этого нужно нажать кнопку рядом с соответствующим документом (1), по желанию добавить *"Сопроводительный комментарий"*(2) и нажать кнопку *"Применить"*(3) (см. рис. 9.1.3)

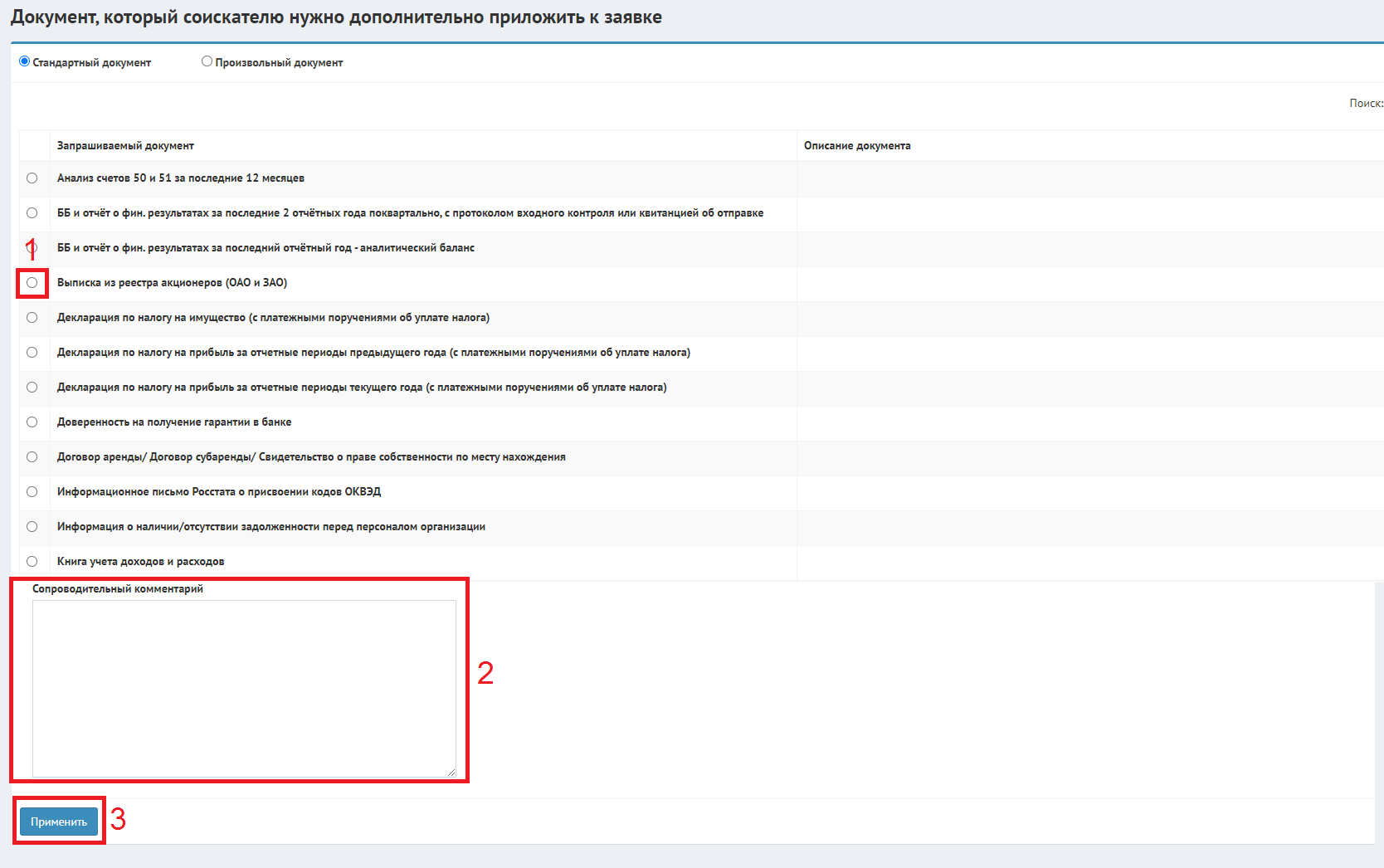


Рисунок 1.1.3 - Окно добавления документа из стандартного набора

* Представление "*Произвольный документ"*позволяет включить в набор уникальный документ, не описанный в системе. Для этого необходимо присвоить наименование добавляемому документу и прикрепить файл с бланком запрашиваемого документа, по желанию также добавить *"Сопроводительный комментарий"* и нажать кнопку *"Применить"*(см. рис 1.1.4)

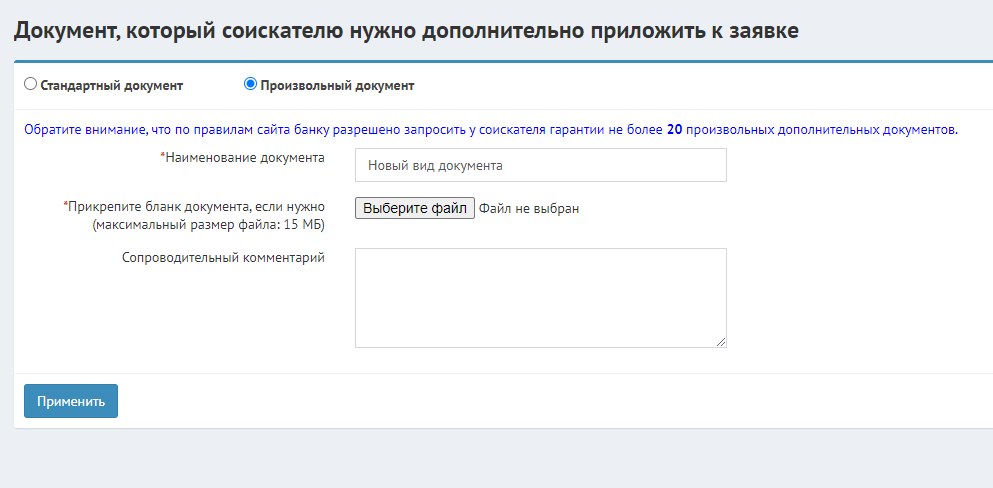
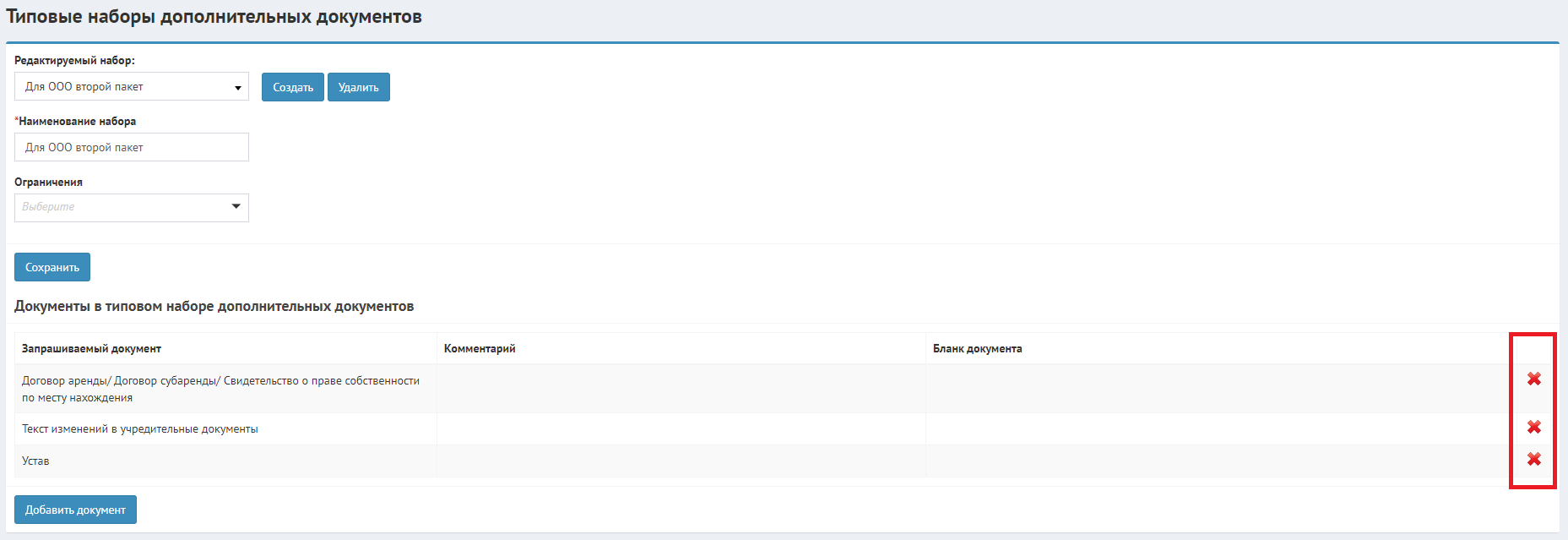


Рисунок 1.1.4 - Окно добавления произвольного документа

После того, как все необходимые документы добавлены в набор - нажмите кнопку "Сохранить" в окне редактирования набора (см. рис. 1.1.2).

### 1.2 Редактирование набора доп. документов

Окно редактирования открывается по умолчанию при выборе справочника *"Наборы доп. док-ов"*. Чтобы отредактировать конкретный набор - выберите его по наименованию в поле *"Редактируемый набор".*В окне редактирования доступно изменение наименования набора, выбор другого ограничения, а также добавление/удаление документов (см. рис. 1.2.1). Чтобы удалить документ из набора, нажмите кнопку *"Удаление" _scroll_external/attachments/worddav7c71a0e728ce89c6fbe819ffaaf3ede1-5a38c54266d9db1ff47766c00f78433379b6db3aa33abdd53970f6f616fa871b.png.*

**Рисунок 1.2.1 - Окно редактирования набора - удаление документов из набора

### 1.3 Удаление набора доп. документов

Чтобы удалить существующий типовой набор дополнительных документов - нажмите кнопку *"Удалить"*рядом с полем *"Редактируемый набор",*находясь в окне редактирования типового набора (см. рис. 1.3.1)

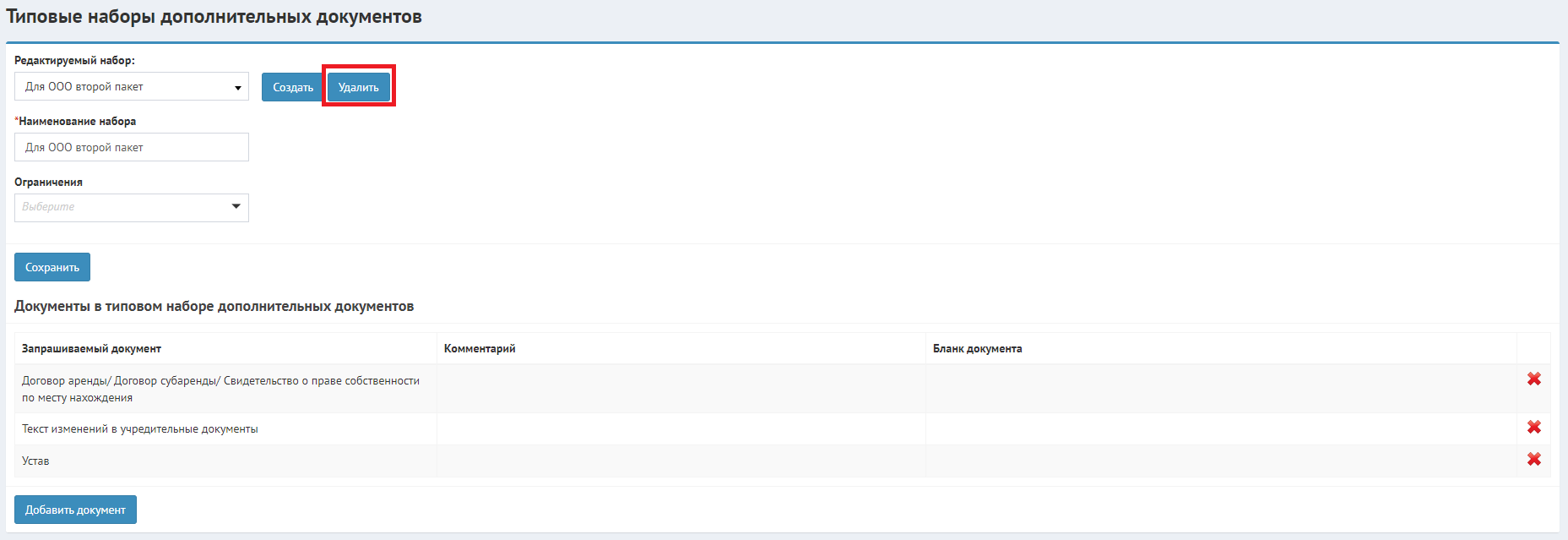


Рисунок 1.3.1 - Окно редактирование набора - удаление набора

## 2.Пакеты документов

Данный справочник предназначен для создания, просмотра и редактирования пакетов документов для различных организационно-правовых форм, которые будут запрошены при подаче заявки.

* [2.1 Просмотр списка существующих пакетов документа](#scroll-bookmark-52)
* [2.2 Создание пакета документов](#scroll-bookmark-53)
* [2.3 Редактирование пакета документов](#scroll-bookmark-54)

### 2.1 Просмотр списка существующих пакетов документа

Чтобы просмотреть список существующих пакетов, выберите пункт *"Пакеты документов"*из раздела *"Справочники".*Откроется окно со списком созданных пакетов документов (см. рис. 2.1.1).

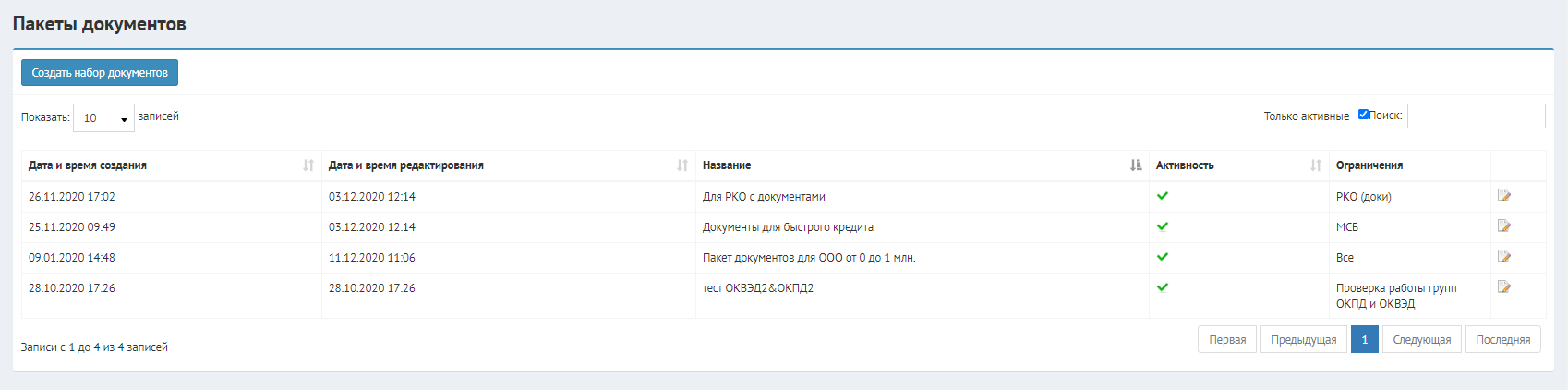


Рисунок 2.1.1 - Окно со списком пакетов документов (только активные)

По умолчанию отображаются только активные (т.е. доступные для применения) пакеты. Чтобы увидеть полный список созданных пакетов, снимите флажок "*Только активные"*в окне поиска (см. рис. 2.1.2)

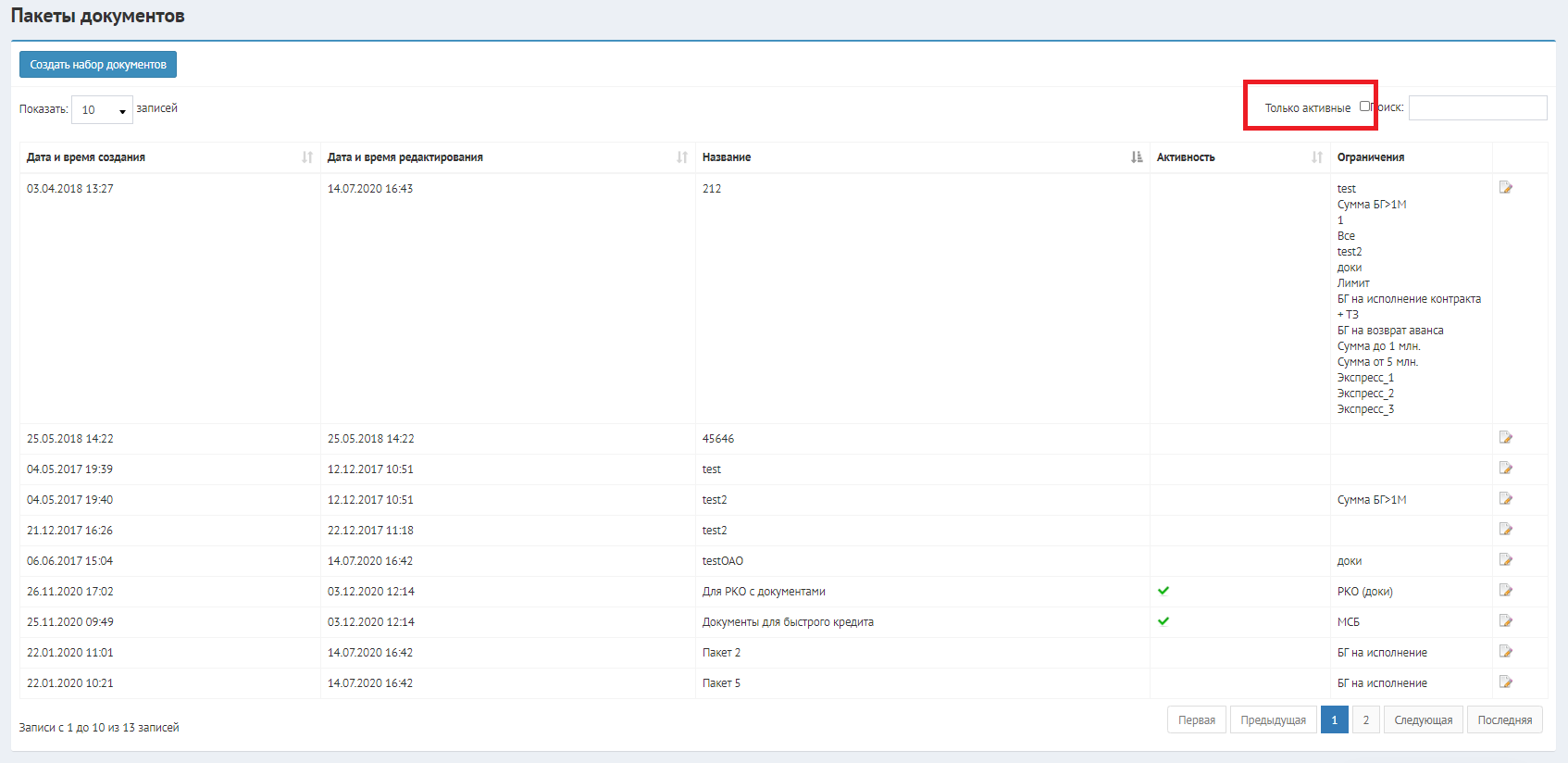


Рисунок 2.1.2 - Окно со списком пакетов документов (все)

### 2.2 Создание пакета документов

Чтобы создать новый пакет документов, нажмите кнопку *"Создать набор документов"*в окне со списком пакетов (см. рис. 13.1.2). Откроется оно для создания нового пакета документов (см. рис. 13.2.1).

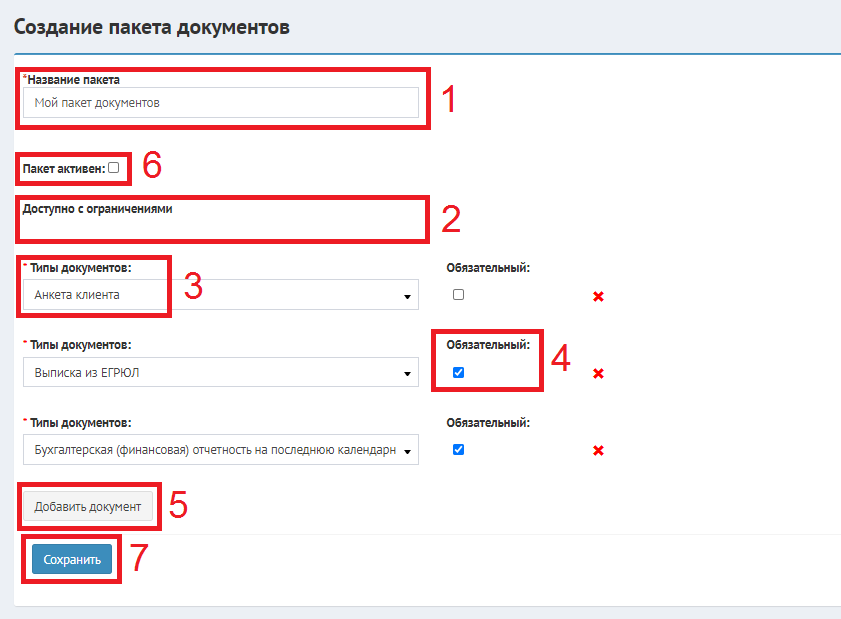


Рисунок 13.2.1 - Окно создания пакета документов

Для создания пакета необходимо:

* Заполнить поле *"Название пакета" (1)*
* Выбрать список ограничений (в качестве вариантов выбора используется содержимое справочника *"Ограничения"*), с которыми может быть применен данный пакет (2)
* Добавить хотя бы один документ в пакет. Добавление осуществляется путем выбора из списка доступных документов в поле *"Типы документов" (3).*а
* Чтобы требовать обязательного прикрепления документа при использовании пакета - установите флажок "*Обязательный" (4)* рядом с соответствующим документом
* Чтобы расширить количество документов в пакете нажмите кнопку *"Добавить документ" (5)*
* Для использования пакета документов в составе заявок необходимо активировать флажок *"Пакет активен" (6)*
* Чтобы удалить лишний документ нажмите кнопку *"Удаление" |_scroll_external/attachments/worddav7c71a0e728ce89c6fbe819ffaaf3ede1-94121d7a90362d56f11958d6beae2e1f8d8ceab0b316f6a6a148d4843542aa07.png*
* Для сохранения пакета нажмите кнопку *"Сохранить" (7)*

### 2.3 Редактирование пакета документов

Для редактирования существующего пакета документов нажмите кнопку "*Редактирование*"_scroll_external/attachments/worddavd81b8c697a19a11a1fbddc4ee1bccf42-6e5ab42e922b82cc1154667c157db2c08020da6e72082b5374be16be7ce53f0e.pngрядом с соответствующим пакетом документов в окне со списком доступных пакетов. Окно редактирование аналогично окну создания пакета (см. рис. 2.3.1). Для редактирования доступны все поля. Чтобы сохранить изменения, нажмите кнопку *"Сохранить".*

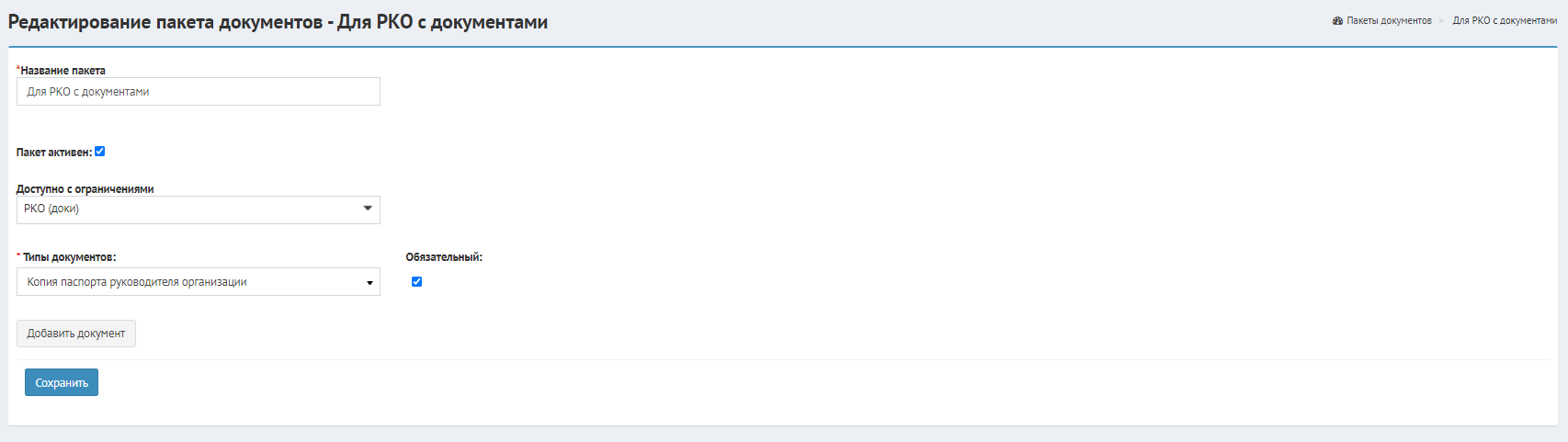


Рисунок 2.3.1 - Окно редактирования пакета документов

## 3. Сценарии, алгоритмы

### 3.1. Алгоритм автоматического назначения ответственного сотрудника в заявку (change\_assign\_user)

#### 3.1.1. Основной сценарий

1) Сначала смотрим, кого должны назначить для повторной заявки:

По истории назначения ОС по заявкам (ebb\_historyrequestsassigneduser) ищем самую свежую запись (по assigned\_date), удовлетворяющую одновременно следующим условиям, как в текущей заявке:

* + Если указан тендер (сопоставляем по request\_id.tender\_id и req.tender\_id):
    - Номером тендера (number)
    - Номером лота (numberLot)
    - ИНН заказчика (customerInn)
  + ИНН соискателя (request\_id → inn=req.inn)
  + Статусом назначения (assigned\_status\_id IN req.status\_id)

И в которой указан пользователь (user\_id IS NOT NULL), для которого существует правило назначения (ebb\_assigneduserrules.assigned\_user\_id=user\_id), удовлетворяющее следующим условиям:

* + Активное (active=True)
  + Стоит признак 'Назначать ответственным в повторной заявке' (assign\_when\_repeat=True)
  + Без указания типов банковских продуктов (product\_types\_id IS NULL) ИЛИ с указанием банковского продукта текущей заявки (product\_types\_id IN req.product\_id)
  + Без признака 'Проверять по сумме БГ для повторной заявки' (assign\_when\_repeat\_check\_sum\_bg=False) ИЛИ с признаком (assign\_when\_repeat\_check\_sum\_bg=True) И (подходящей суммой (см. ниже [Алгоритм сравнения сумм для правил (get\_lookup\_use\_summ\_rule)](#scroll-bookmark-33)) ИЛИ с нулевыми суммами (min\_sum\_bg=0 AND max\_sum\_bg=0)).

Ну и берем самое первое активное правило для этого пользователя с признаком 'Назначать ответственным в повторной заявке' (rule = ebb\_assigneduserrules(assigned\_user\_id=user\_id, assign\_when\_repeat=True, active=True).

2) Если мы не нашли такое правило

2.1) Смотрим, может не стоит менять ответственного сотрудника и подходит уже назначенный:

Смотрим, существует ли в правилах назначения правило, одновременно удовлетворяющее следующим условиям:

* + - Активное (active=True)
    - Принадлежит уже назначенному ОС (assigned\_user\_id=req.bidders\_requestsadditionaldata.assigned\_user\_id)
    - Сам сотрудник не удален (ebb\_extendedusers(id = assigned\_user\_id).wasRemoved=False)
    - Заявка может назначаться на ОС в этом статусе (statuses\_sum\_bg IN req.status\_id)
    - Без указания типов банковских продуктов (product\_types\_id IS NULL) ИЛИ с указанием банковского продукта текущей заявки (product\_types\_id IN req.product\_id)

Если существует, значит менять ничего не надо и выходим из алгоритма.

2.2) Иначе, пытаемся найти правила для назначения, одновременно удовлетворяющие следующим условиям:

* + - Активное (active=True)
    - Сам сотрудник не удален (ebb\_extendedusers(id = assigned\_user\_id).wasRemoved=False)
    - Заявка может назначаться на ОС в этом статусе (statuses\_sum\_bg IN req.status\_id)
    - Без указания типов банковских продуктов (product\_types\_id IS NULL) ИЛИ с указанием банковского продукта текущей заявки (product\_types\_id IN req.product\_id)
    - Подходящее по сумме (см. ниже [Алгоритм сравнения сумм для правил (get\_lookup\_use\_summ\_rule)](#scroll-bookmark-33) ИЛИ с нулевыми суммами (min\_sum\_bg=0 AND max\_sum\_bg=0))
    - Если выполняем данную проверку первый раз (в рамках текущего алгоритма возможен переход к этому пункту из п.3, т.е. если мы выполняемся при переходе из него, то этот фильтр не нужен):   
      Подходящее по скорингу заявки (scorings\_id IN req.bidders\_requestsadditionaldata.used\_scoring\_id)

Для каждого такого правила вычисляем:

* + - Приоритет роли из настроек (priority\_max) = MAX(priority из ebb\_assignedusersettings где role\_id есть в списке ролей пользователя правила (ebb\_userroles(user\_id=assigned\_user\_id)))
    - Приоритет суммы или привязки сотрудника к компании по ИНН из настроек распределения для ролей (filter\_by\_summ) = MAX(use\_sum\_bg\_or\_assigned\_bidder\_priority из ebb\_assignedusersettings где role\_id есть в списке ролей пользователя правила (ebb\_userroles(user\_id=assigned\_user\_id)))
    - Собственно, определяем признак - есть привязка компании к сотруднику по ИНН или нет (filter\_by\_bidder) = 1, если есть привязка (EXISTS ebb\_extendedusersassignedinnkpp(user\_id=assigned\_user\_id, inn = req.inn)), иначе 0

и дополнительно смотрим, если в заявке указана годовая выручка (берем самый свежий годовой показатель 2110 из ФП → Форма 2 (в сис.переменных это было бы 2110^y), то берем его, умножаем на 1000 и проверяем, что правило соответствует условию

COALSECE('Годовая выручка от (руб.)' (min\_fp\_2110\_year), -INF) <= 2110^y \* 1000 <= COALESCE('Годовая выручка до (руб.)' (max\_fp\_2110\_year), INF)

Далее, для более равномерного распределения по сотрудникам нам необходимо посчитать, кому сколько заявок уже назначено по настройкам правил.

Поэтому для каждого правила считаем кол-во заявок, назначенных сотруднику (count\_req), при этом вообще исключаем правила, если они относяться к уже назначенному сотрудинку в заявке (assigned\_user\_id=req.bidders\_requestsadditionaldata.assigned\_user\_id), т.к. проверка по нему уже была в предыдущем пункте и значит тут нам он уже не подходит:

count\_req = Выбираем кол-во заявок, исключая текущую, где назначен сотрудник из правила (bidders\_requests ← bidders\_requestsadditionaldata.assigned\_user\_id=assigned\_user\_id) и одновременно выполняются следующие условия

* + - * Заявка в статусе 'Статусы для определения количества назначенных заявок' (bidders\_requests.status\_id IN max\_current\_assigned\_requests\_statuses\_id)
      * Продукт заявки в списке 'Типы банковских продуктов для определения количества назначенных заявок' (bidders\_requests.product\_id IN current\_assigned\_requests\_product\_types\_id), если список указан, иначе все подходят

И оставляем только правила, подходящие под условие

* + - Стоит признак 'Учитывать количество назначенных заявок' (use\_max\_current\_assigned\_requests\_count) И 'Максимальное количество назначенных заявок' (max\_current\_assigned\_requests\_count) > count\_req

ИЛИ

* + - Не стоит признак 'Учитывать количество назначенных заявок'

ИЛИ

* + - 'Максимальное количество назначенных заявок'=0

Далее берем первое подходящее правило с учетом сортировки (если какое-то из полей сортировки NULL, то считаем, что оно = 0)

* + - priority\_max DESC
    - filter\_by\_summ DESC
    - filter\_by\_bidder DESC
    - priority DESC
    - count\_req ASC
    - На всякий случай, добавим элемент рандомности ) RANDOM ASC

3) Если на предыдущих этапах мы так и не нашли нужное правило, то всё равно упорно стараемся найти ОС, поэтому ищем правило, у которого стоит признак 'Назначать, если не найдено других сотрудников', т.е. выбираем правила, которые соответствуют следующим условиям одновременно

* + Активное (active=True)
  + Сам сотрудник не удален (ebb\_extendedusers(id = assigned\_user\_id).wasRemoved=False)
  + Заявка может назначаться на ОС в этом статусе (statuses\_sum\_bg IN req.status\_id)
  + Стоит признак 'Назначать, если не найдено других сотрудников' (assign\_request\_if\_no\_any\_user)
  + Без указания типов банковских продуктов (product\_types\_id IS NULL) ИЛИ с указанием банковского продукта текущей заявки (product\_types\_id IN req.product\_id)
  + Если выполняем данную проверку первый раз (в рамках текущего алгоритма возможен ещё раз переход к данному пункту, т.е. если мы выполняемся повторно, то этот фильтр не нужен):   
    Подходящее по скорингу заявки (scorings\_id IN req.bidders\_requestsadditionaldata.used\_scoring\_id)

Далее, определяем для каждого правила приоритет по роли

priority\_max = MAX(priority из ebb\_assignedusersettings где role\_id есть в списке ролей пользователя правила (ebb\_userroles(user\_id=assigned\_user\_id)))

И берем первое правило с учетом сортировки

* + priority\_max DESC
  + priority DESC

Если правило найдено - УРА!!! идем дальше, иначе пытаемся ещё раз найти сотрудника, но уже без учета привязки к скорингу, т.е. переходим к п. 2.2), а если уже пытались, то УВЫ, бесполезно искать и выходим.

4) Если в результате всех мытарств на предыдущих шагах правило найдено, то начинаем следующие манипуляции:

* + Отвязываем ОС (см. ниже [Алгоритм отвязки ответственного сотрудника (unfasten\_request)](#scroll-bookmark-34))
  + Записываем в историю по заявке событие 'Открепление ответственного сотрудника' (action = ebb\_lovs(lovName\_id=41, pid=31)
  + Привязываем нового ОС (см. ниже [Алгоритм привязки ответственного сотрудника (fasten\_request)](#scroll-bookmark-35))
  + Записываем в историю по заявке событие 'Назначение ответственного сотрудника' (action = ebb\_lovs(lovName\_id=41, pid=31)

Ну и, если у соискателя по ИНН из заявки нет закрепленного сотрудника (NOT EXISTS ebb\_extendedusersassignedinnkpp(inn=req.inn)) И в правиле стоит признак 'Привязывать к Принципалу без ответственного' (can\_assign\_for\_free\_bidder), то привязываем к компании сотрудника (см. ниже [Алгоритм закрепления сотрудника за компанией по ИНН (fasten\_bidder)](#scroll-bookmark-36)).

### 3.2. Алгоритм сравнения сумм для правил (get\_lookup\_use\_summ\_rule)

#### 3.2.1. Предисловие

Для продуктов (req.product\_id → pid)

* 8, Расчетно-кассовое обслуживание
* 11, РКО (открыть по видеосвязи с документами)

возвращаем NULL.

#### 3.2.2. Основной сценарий

В зависимости от правила 'Правило выбора суммы' (use\_summ\_rule\_id → pid) выбираем сумму по текущей заявке (**TO\_DO - правила выбора сумм используются во многих местах, но пока не описаны**)

* 1: product\_liabiliti\_summ
* 2: get\_product\_summ\_bg
* 3: summ\_bg\_from\_judgment\_or\_request
* 4: summBg

и возвращаем результат сравнения

'Сумма БГ от (руб.)' (min\_sum\_bg) <= полученная сумма <= 'Сумма БГ до (руб.)' (max\_sum\_bg)

### 3.3. Алгоритм отвязки ответственного сотрудника

#### 3.3.1. Основной сценарий

1) Запоминаем текущего ОС из заявки (old\_assigned\_user\_id = req.bidders\_requestsadditionaldata.assigned\_user\_id)

2) Убираем привязку к ОС в заявке (req.bidders\_requestsadditionaldata.assigned\_user\_id = NULL)

3) Берём последнюю запись по id из истории привязок бывшего ОС к заявке (ebb\_historyrequestsassigneduser(user\_id=old\_assigned\_user\_id , request\_id = req.id))

4) Обновляем эту запись

* 'Дата снятия' (removal\_date) = CURRENT\_TIMESTAMP
* 'Статус снятия' (removal\_status\_id) = текущий статус заявки

### 3.4. Алгоритм привязки ответственного сотрудника (fasten\_request)

#### 3.4.1. Основной сценарий

1) Запоминаем текущего ОС из заявки (old\_assigned\_user\_id = req.bidders\_requestsadditionaldata.assigned\_user\_id)

2) Заменяем привязку к ОС в заявке (req.bidders\_requestsadditionaldata.assigned\_user\_id = id нового сотрудника)

3) Берём последнюю запись по id из истории привязок бывшего ОС к заявке (ebb\_historyrequestsassigneduser(user\_id=old\_assigned\_user\_id , request\_id = req.id))

4) Обновляем эту запись

* 'Дата снятия' (removal\_date) = CURRENT\_TIMESTAMP
* 'Статус снятия' (removal\_status\_id) = текущий статус заявки

5) Добавляем в историю привязок ОС по заявке  (ebb\_historyrequestsassigneduser) запись

* 'Ответственный сотрудник' (user\_id) = id нового сотрудника
* 'Дата назначения' (assigned\_date) = CURRENT\_TIMESTAMP
* 'Статус назначения' (assigned\_status) = текущий статус заявки

### 3.5. Алгоритм закрепления сотрудника за компанией по ИНН (fasten\_bidder)

#### 3.5.1. Основной сценарий

1) Ищем старую привязку к сотруднику (ebb\_extendedusersassignedinnkpp(inn=req.inn))

2) Запоминаем старого сотрудника (old\_assigned\_user\_id)

3) Если привязка есть, то:

3.1) Обновляем в ней

* + 'Ответственный сотрудник' (user\_id) = id нового сотрудника

3.2) Берём последнюю запись по id из истории закрепленных сотрудников по ИНН (ebb\_historyextendedusersassignedinnkpp(inn=inn, user\_id=old\_assigned\_user\_id)) и обновляем в ней

* + 'Дата снятия' (removal\_date) = CURRENT\_TIMESTAMP
  + 'Статус снятия' (removal\_status\_id) = текущий статус заявки
  + 'Открепляющий сотрудник' (unfasten\_user\_id) = id текущего пользователя

4) Если привязки нет, то создаём её (ebb\_extendedusersassignedinnkpp)

* + 'ИНН' (inn) = ИНН
  + 'Ответственный сотрудник' (user\_id) = id нового сотрудника

5) Создаём в истории закрепленных сотрудников по ИНН (ebb\_historyextendedusersassignedinnkpp) запись

* 'ИНН' (inn) = ИНН
* 'Ответственный сотрудник' (user\_id) =  id нового сотрудника
* 'Дата назначения' (assigned\_date) = CURRENT\_TIMESTAMP
* 'Статус назначения' (assigned\_status) = текущий статус заявки

## 4. Сценарии, алгоритмы

### 4.1. Алгоритм получения кода налогового органа по данным заявки (get\_bidder\_nalog\_code)

#### 4.1.1. Основной сценарий

1)  Получение кода ИФНС  
Если в заявке указан КПП (bidders\_requests.kpp) - берем первые 4 цифры КПП (bidders\_requests.kpp)- это и есть код налогового органа, который нам нужен;

Иначе - делаем запрос в к шлюзу [Шлюз Api Gateway (AGW)](https://docs.srvhub.tools/pages/viewpage.action?pageId=66650114) по ИНН участника торгов  (bidders\_requests.inn) метод  egrDetails и получаем из ответа сведения о постановке на учет в налоговом органе - **nalogRegBody.nalogCode** - код налогового органа.

* + если возвращается код, то переходим к следующему пункту;
  + если данных нет - возвращаем пустой результат.

2) Получение реквизитов ИФНС

По коду ИФНС делаем запрос к  [Шлюз Api Gateway (AGW)](https://docs.srvhub.tools/pages/viewpage.action?pageId=66650114)  методом **get\_ifns\_requisites\_by\_code** [PKG04.UCS1 - Сервис загрузки и получения реквизитов ИФНС (ifns\_requisites)](https://docs.srvhub.tools/pages/viewpage.action?pageId=78709936)

Из полученного положительного ответа выбираем следующие данные:

* + ИНН получателя (receiver\_inn)
  + КПП получателя (receiver\_kpp)
  + Счет № (bank\_account\_number)
  + Получатель платежа (payment\_receiver)
  + БИК (bank\_account\_bik)

3) Если мы получили данные  
3.1) Сохраняем JSON в поле tax\_qrcode в таблицу it2b\_investmentpaymentschedule

3.2) Формируем QR-Код по алгоритму  [PKG05.09.UCS1 - Формирование QR-кода для оплаты](https://docs.srvhub.tools/pages/viewpage.action?pageId=54984793)   со следующими данными:

Схема QR-кода:

|  |
| --- |
| {        "name":   { payment\_receiver },      "personal\_acc": { bank\_account\_number },  "bank\_name" : { bank\_name },      "bic": { bank\_account\_bik },  "corresp\_acc" : { bank\_account\_corr },  "OKTMO" : { OKTMO\_code },  "sum" : { str(investmentpaymentschedule.paid\_percentage\_amount\_tax) в формате # ###.## рублей},    "kpp": { receiver\_kpp },      "payee\_inn": { receiver\_inn },  "payer\_inn" : { req.inn },    "cbc": { "18210102010011000110" },    "purpose": { "Налог на доходы физических лиц за {it2b\_investmentpaymentschedule.payment\_date в формате январь 2021 } года" }      } |

Code Block 1 Схема QR-кода

4) Если не получены реквизиты ИФНС - продолжаем работу.

5) Формируем письмо с QR-кодом или без  - [PKG05.05.01.S29 -Шаблон письма с QR-кодом для оплаты НДФЛ за инвесторов](https://docs.srvhub.tools/pages/viewpage.action?pageId=79954997)

6) Отправляем письмо заемщику с помощью вызова функции send\_msg с темой "Уведомление об оплате НДФЛ за инвесторов". На адрес bidders\_requests.bidderEmail

### 4.2. Основная цель(и), назначение

Генерация превью загружаемых пользователем документов.

### 4.3. Краткое описание и связь с другими историями

Сервис в дальнейшем будет использоваться для отображения миниатюр на [PKG01.01.UCS1 - Страница личного кабинета инвестора IT2B](https://docs.srvhub.tools/pages/viewpage.action?pageId=79957773)

## 5. Сценарии, алгоритмы

### 5.1. Алгоритм генерации превью документов (preview-generator)

#### 5.1.1. Основной сценарий

Принимаем какой-либо из следующих параметров:

1. Только высоту
2. Только ширину
3. Высоту и ширину одновременно

Тело исходника получаем в body  (обязательное).

1. Если получена только высота или ширина вычисляем второй размер.
2. Если переданы оба параметра, изображение должно быть помещено по центру полотна этого размера, и выполнена подрезка сторон без искажений.

На выходе получаем файл с заданным расширением и код 200.  
Или устанавливаем код ошибки 400, 500 и в detail пишем исключение.