Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Федеральная государственная информационная система «Единая система предоставления государственных и муниципальных услуг (сервисов)»

Инструкция по настройке процессов предоставления государственных и муниципальных услуг

Москва, 2023

Содержание

Термины и сокращения 4
1. Введение 7
1.1. Область применения....7
1.2. Краткое описание возможностей Конструктора процессов предоставления услуг 7
1.3. Уровень подготовки специалистов 7
2. Условия применения Конструктора процессов предоставления услуг....9
3. Описание операций....10
3.1. Авторизация и переход к конструктору процессов. Список услуг 10
3.2. Создание услуги....13
3.3. Внесение основной информации об услуге 14
3.4. Процессы. Операции с процессами 19
3.5. Получение процесса импорта из XSD-схемы услуги....22
3.6. Просмотр и редактирование процесса импорта 34
3.6.1. Настраиваемый процесс импорта 34
3.6.2. Базовый импорт 41
3.7. Добавление и редактирование справочников. Добавление оснований для отказа в приеме документов и оснований для отказа в предоставлении услуги....45
3.8. Входные и выходные документы. Добавление видов входных документов для предоставления услуги....53
3.9. Добавление административных процедур услуги....57
3.10. Работа с межведомственными запросами для предоставления услуги....62
3.10.1. Добавление межведомственных запросов для предоставления услуги 62
3.10.2. Создание процесса межведомственного взаимодействия 67
3.10.3. Создание волн межведомственного взаимодействия (если необходимо) 77
3.11. Настройка проверок....85
3.12. Создание DMN-таблицы проверок 91
3.13. Создание процесса подготовки данных....96
3.14. Настройка экранных форм....106
3.14.1. Общие сведения об экранных формах. Операции с экранными формами 106
3.14.2. Описание основных компонентов кода экранной формы 113
3.14.3. Кодирование свойств атрибутов 116
3.15. Добавление шаблонов выходных документов для предоставления услуги 134
3.16. Создание пустых экранных форм для результирующих документов 140
3.17. Добавление результирующих документов и соответствующих им экранных форм в административные процедуры 143
3.18. Донастройка экранных форм для результирующих документов....146
3.19. Реестры решений 147
3.20. Настройка экранной формы очного приема 153
3.20.1. Создание и кодирование экранной формы очного приема 154

3.20.2. Документы, прикрепляемые к заявлению, формируемому при очном приеме.....166
3.21. Настройка криптопровайдера КриптоПро 166
3.22. Эмуляции процесса обработки XML-заявления в Конструкторе процессов 174
3.22.1. Эмуляция процесса импорта заявления 174
3.22.2. Эмуляция процесса выполнения межведомственных запросов 176
3.22.3. Эмуляция процесса выполнения проверок 180
Лист регистрации изменений.....183

Термины и сокращения

Термин, сокращение	Определение
Авторизация	Процедура предоставления субъекту определенных прав доступа
Административная процедура	Нормативно установленный последовательный порядок реализации административных действий уполномоченными субъектами в целях реализации их компетенции и оказания услуг
Административное действие	Предусмотренное административным регламентом действие должностного лица в рамках исполнения государственной функции (предоставления государственной услуги)
Администратор	Лицо, наделенное правами для осуществления деятельности в административной консоли ПГС
АРМ	Автоматизированное рабочее место
Аутентификация	Процедура проверки подлинности
Ведомства и организации	Федеральные органы исполнительной власти, государственные корпорации, которые в соответствии с федеральным законом наделены полномочиями по исполнению государственных функций и предоставлению государственных услуг в установленной сфере деятельности, органы государственных внебюджетных фондов, исполнительные органы государственной власти субъектов Российской Федерации, органы местного самоуправления, организации, участвующие в предоставлении государственных и муниципальных услуг
ВС	Вид сведений
ГЛР	Государственный лесной реестр
Государственная услуга, услуга	Деятельность по реализации функций федерального органа исполнительной власти, государственного внебюджетного фонда, исполнительного органа государственной власти субъекта Российской Федерации, а также органа местного самоуправления при осуществлении отдельных государственных полномочий, переданных федеральными законами и законами субъектов Российской Федерации, которая осуществляется по запросам заявителей в пределах установленных нормативными правовыми актами Российской Федерации и нормативными правовыми актами субъектов Российской Федерации полномочий органов, предоставляющих государственные услуги
Документ	Совокупность сведений, зафиксированных в установленной для них форме и подтверждающих юридически значимые факты
ЕГР ЗАГС	Единый государственный реестр записей актов гражданского состояния
ЕПГУ	Федеральная государственная информационная система «Единый портал государственных и муниципальных услуг (функций)»
ЕСИА	Федеральная государственная информационная система «Единая система идентификации и аутентификации в инфраструктуре, обеспечивающей информационно-технологическое взаимодействие информационных систем, используемых для предоставления государственных и муниципальных услуг в электронной форме»

Термин, сокращение	Определение
ИНН	Идентификационный номер налогоплательщика
Карточка	Представление информационного объекта заданного типа (услуга и т. п.) в ПГС, включая форму ввода (редактирования) данных об информационном объекте
Контекст процесса	Все данные, которые создаются, изменяются и сохраняются во время выполнения процесса
Машиночитаемый вид (форма) документа, машиночитаемый документ	Форма представления информации, доступная для интерпретации программным или аппаратным обеспечением ЭВМ
МВ	Межведомственное взаимодействие
МВЗ, МЗ	Межведомственный запрос
Муниципальная услуга	Деятельность по реализации функций органа местного самоуправления, осуществляемая по запросам заявителей в пределах полномочий органа, предоставляющего муниципальные услуги, по решению вопросов местного значения, установленных в соответствии с Федеральным законом от 06.10.2003 № 131-ФЗ «Об общих принципах организации местного самоуправления в Российской Федерации» и уставами муниципальных образований
НПА	Нормативный правовой акт
НСИ	Нормативно-справочная информация
Объект МЗ	Совокупность сведений (атрибутов), передаваемых адресату в МЗ
ОГРН	Основной государственный регистрационный номер
ПГС	Федеральная государственная информационная система «Единая система предоставления государственных и муниципальных услуг (сервисов)»
ПО	Программное обеспечение
ППД	Процесс подготовки данных
Процесс	Пошаговое совершение действий, направленных на оказание (предоставление) государственной (муниципальной) услуги
СМЭВ	Федеральная государственная информационная система «Единая система межведомственного электронного взаимодействия»
СМЭВ 2	Предыдущая по отношению к СМЭВ 3 версия федеральной государственной информационной системы «Единая система межведомственного электронного взаимодействия»
СМЭВ 3	Федеральная государственная информационная система «Единая система межведомственного электронного взаимодействия», функционирующая в соответствии с Методическими рекомендациями по работе с Единой системой межведомственного электронного взаимодействия версии 3.x
СНИЛС	Страховой номер индивидуального лицевого счёта

Термин, сокращение	Определение
Сущность	Любой конкретный или абстрактный объект, включая связи между объектами
Услуга	Государственная (муниципальная) услуга
Файл печатной формы	Файл преднастроенного шаблона типового документа для печати, часть атрибутов которого заполняется автоматически
ФИАС	Федеральная информационная адресная система
Человекочитаемый образ документа	Представление машиночитаемого документа в виде, пригодном для чтения человеком
Шаблон экранной формы шага процесса	Преднастроенная экранная форма шага процесса, на основе которой может быть создана уникальная форма шага процесса
Шаг процесса	Составная часть процесса, единичное действие, совершаемое при исполнении процесса
ЭВМ	Электронно-вычислительная машина
ЭП	Электронная подпись
ЭП-СП	Электронная подпись служебного пользования
BPMN	Business Process Model and Notation (нотация и модель бизнес-процессов) — система условных обозначений (нотация) и их описания на языке XML для моделирования бизнес-процессов
DMN	Decision Model and Notation — единая, понятная всем бизнес-пользователям нотация для моделирования процессов принятия решений
REST	Representational State Transfer (передача репрезентативного состояния) — архитектурный стиль взаимодействия компонентов распределенного приложения в сети
SOAP	Simple Object Access Protocol (простой протокол доступа к объектам) — протокол обмена структурированными сообщениями в распределенной вычислительной среде
URL	Uniform Resource Locator — система унифицированных адресов электронных ресурсов, или единообразный определитель местонахождения ресурса (файла)
XML	eXtensible Markup Language (расширяемый язык разметки) — язык для создания структурированных машиночитаемых документов
XML Schema	Язык описания структуры XML-документа
XSD	XML Schema Definition — документ, содержащий описание структуры XML-документа на языке XML Schema
XSL	eXtensible Stylesheet Language — семейство рекомендаций консорциума W3C, описывающее языки преобразования и визуализации XML-документов

1. Введение

1.1. Область применения

Настоящий документ предназначен для специалистов органов государственной власти, местного самоуправления и организаций, осуществляющих настройку процессов предоставления государственных и муниципальных услуг в федеральной государственной информационной системе «Единая система предоставления государственных и муниципальных услуг (сервисов)» (также используется наименование «Платформа государственных сервисов», сокр. «ПГС») с использованием встроенных инструментов конструирования процессов.

1.2. Краткое описание возможностей Конструктора процессов предоставления услуг

Конструктор процессов оказания услуг ПГС:

– предоставляет сотрудникам ведомств и организаций инструмент настройки процессов, соответствующих целевым моделям оказания государственных (муниципальных) услуг, с использованием нотации BPMN 2.0;

– предоставляет ведомствам и организациям библиотеку шаблонов процессов для оказания государственных (муниципальных) услуг и исполнения связанных с их оказанием государственных (муниципальных) функций (с автоматическим межведомственным взаимодействием, автоматическим принятием решения, реестровой моделью).

1.3. Уровень подготовки специалистов

К квалификации специалистов, настраивающих процессы предоставления услуг, предъявляются следующие требования:

– навыки работы с персональным компьютером (рабочей станцией) и операционной системой Microsoft Windows или аналогичными (многооконный интерфейс) на уровне опытного пользователя;

– знания и навыки:

а) работы в сети Интернет при помощи веб-браузеров различных разработчиков;

б) работы с офисными программами;

в) работы с усиленной квалифицированной электронной подписью типа ЭП-СП;

г) написания простых скриптов на языках Java/Groovy/JavaScript;

д) экспертные знания предметной области оказываемых услуг;

– знание:

а) REST, протокола SOAP;

б) нотации BPMN 2.0;

в) форматов представления данных XML, XSD, XSL, JSON;

г) структурированных языков запросов к базам данных (ArangoDB Query Language — AQL);

д) принципов работы графовых баз данных.

2. Условия применения Конструктора процессов предоставления услуг

Для настройки процессов предоставления услуг допускается использование браузеров:

– Google Chrome версии 85 и выше;

– Yandex Browser версии 20 и выше.

Также на АРМ лица, настраивающего процессы предоставления услуг, должно быть установлено ПО:

– Camunda Modeler — для моделирования схем бизнес-процессов;

– криптопровайдер КриптоПро;

– КриптоПро ЭЦП Browser plug-in (CAdES-плагин);

– CryptoPro Extension for CAdES Browser Plug-in.

Кроме того, для настройки процессов предоставления услуг сотруднику необходимо иметь:

– учетную запись с предоставленными правами администратора услуг ПГС;

– настроенную ЭП на АРМ сотрудника;

– примеры генерируемых на ЕПГУ заявлений о предоставлении настраиваемых услуг в формате XML;

– НПА и иные документы, содержащие официальные сведения о данных услугах.

Перед созданием услуги в ПГС необходимо разработать (или получить от стороннего разработчика) и зарегистрировать в СМЭВ вид сведений (ВС), соответствующий данной услуге, включая XSD-схему услуги, а также Руководство пользователя, содержащее полный состав атрибутов услуги с их описанием. Если ВС был разработан и зарегистрирован в СМЭВ сторонним разработчиком, то соответствующая XSD-схема доступна на странице ВС на портале https://lkuv.gosuslugi.ru.

3. Описание операций

Для того чтобы создать услугу и настроить процесс ее предоставления в ПГС, необходимо последовательно выполнить действия, описанные в подразделах 3.1–3.21.

3.1. Авторизация и переход к конструктору процессов. Список услуг

Для того чтобы перейти к конструктору процессов в ПГС, необходимо:

1. В браузере перейти по URL-адресу ПГС.

2. Авторизоваться в ПГС с ролью администратора услуг (если пользователь уже авторизован, пропустить данный шаг): на открывшейся странице (рисунок 1) ввести данные учётной записи в поле «Имя пользователя» и «Пароль» и нажать кнопку «Вход».

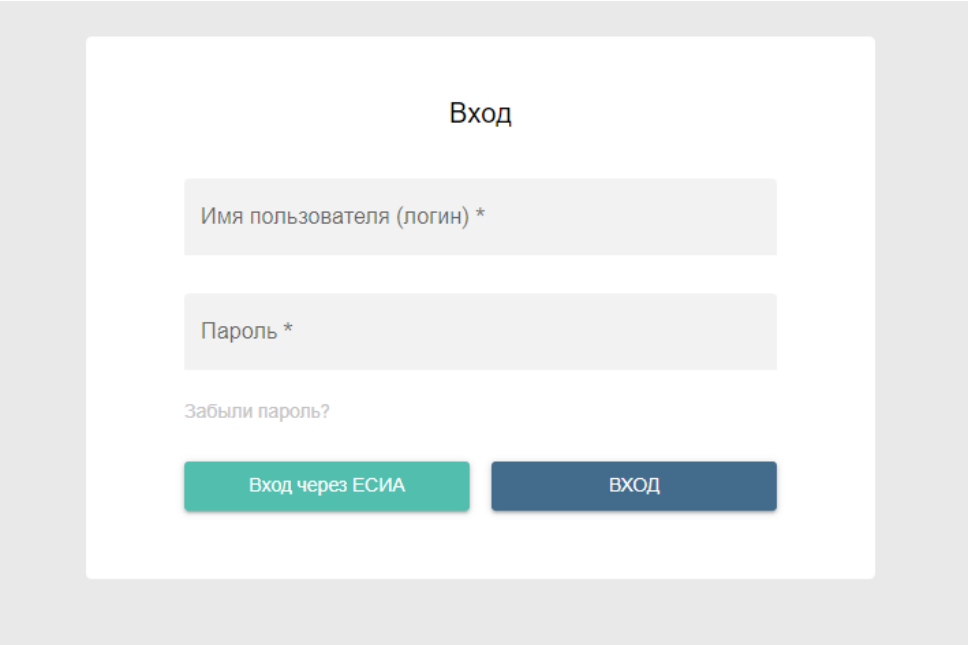


Рисунок 1 — Страница входа в ПГС

При этом откроется страница ПГС, представленная на рисунке 2.

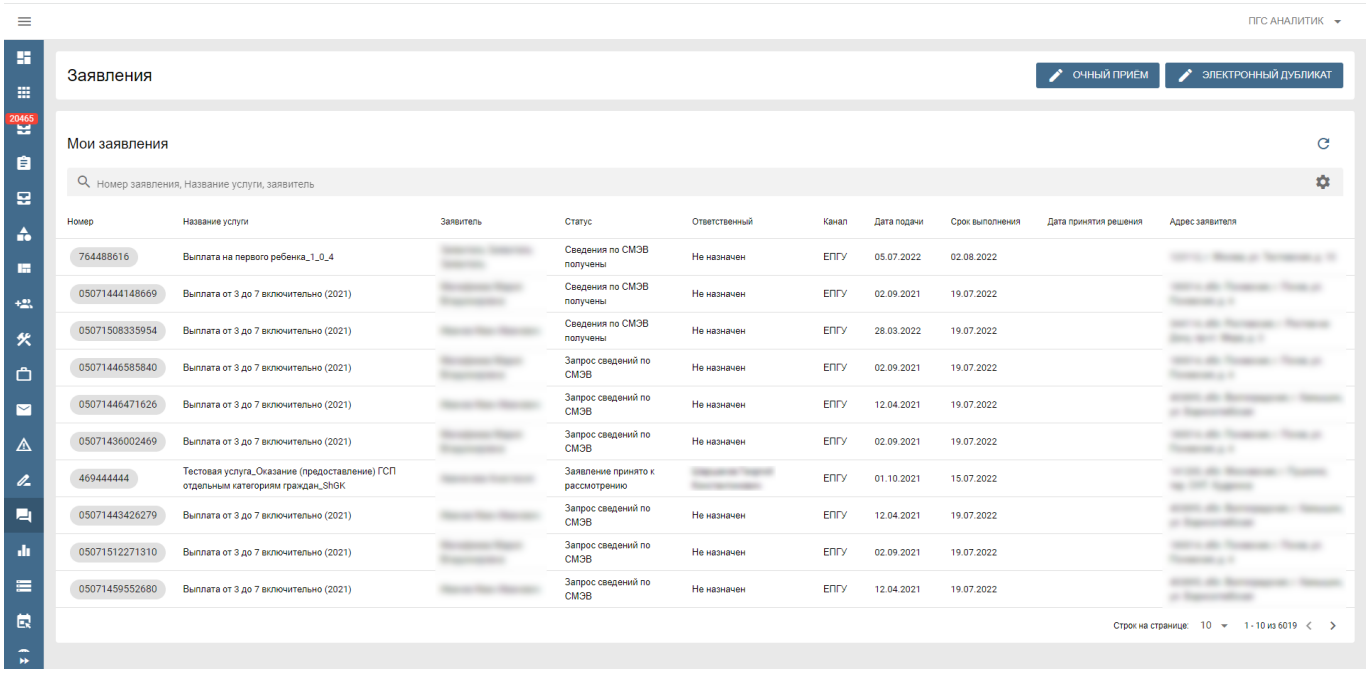


Рисунок 2 — Стартовая страница ПГС (раздел «Заявления»)

3. Перейти в раздел «Конструктор услуг».

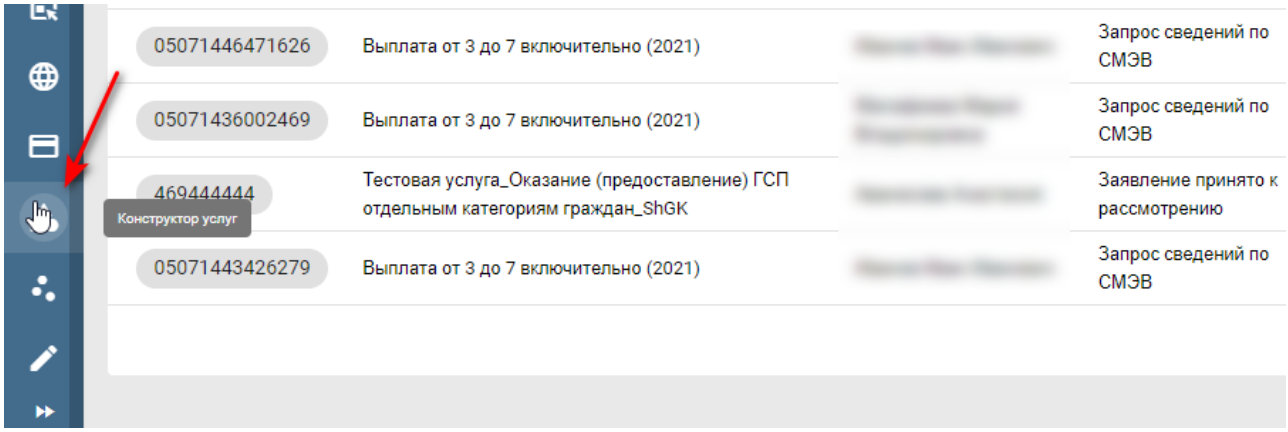


Рисунок 3 — Переход в раздел ПГС «Конструктор услуг»

На стартовой странице раздела «Конструктор услуг» представлен список услуг, созданных с помощью конструктора (рисунок 5). Для каждой услуги в списке отображаются:

– «Краткое имя» — наименование услуги, присвоенное в ПГС, позволяющее пользователям идентифицировать и находить ее;

– «Полное наименование» — наименование услуги, указанное в документации на услугу или на портале https://lkuv.gosuslugi.ru ;

– «Тип услуги» — значение вида request@urn://rostelekom.ru/<код услуги>/<номер версии услуги>.

Здесь и далее по тексту <код услуги> — код услуги, указанный в XSD-схеме услуги, <номер версии услуги> — номер текущей версии услуги, указанный в XSD-схеме услуги (см. раздел 3.3 и рисунок 4).

Пример: request@urn://rostelekom.ru/ExtractForestRegistry/1.0.3

Рисунок 4 — Код услуги и номер версии услуги в XSD-схеме услуги

– «Статус»:

а) «Черновик» — услуга создана, но еще не взята в работу, может быть удалена из ПГС;

б) «В работе» — услуга взята в работу для настройки процесса ее предоставления.

Доступны функции поиска услуг в списке по наименованию и статусу[1] (вызываются кнопкой — «Параметры»), обновления списка для отображения последних возможных изменений (выполняется кнопкой — «Обновить»).

Над списком услуг находятся кнопка создания новой услуги и иные.

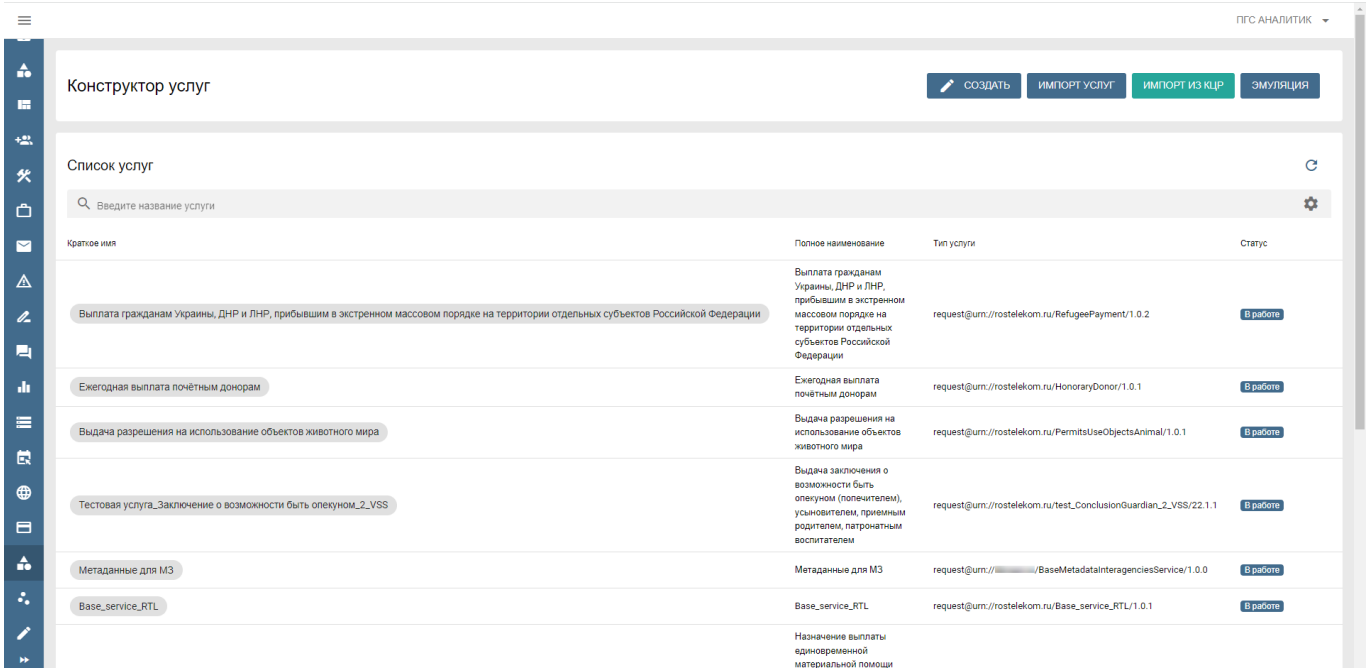


Рисунок 5 — Раздел ПГС «Конструктор услуг». Стартовая страница

[1] Прочие статусы, кроме «Черновик» и «В работе», не используются.

3.2. Создание услуги

Для создания новой услуги необходимо на стартовой странице раздела «Конструктор услуг» нажать кнопку «Создать» (см. рисунок 5 в подразделе 3.1), в открывшемся модальном окне задать краткое имя услуги (рисунок 6) и нажать кнопку «Создать». После этого услуга будет создана со статусом «Черновик» и откроется карточка услуги на вкладке «Основная информация» (рисунок 7).

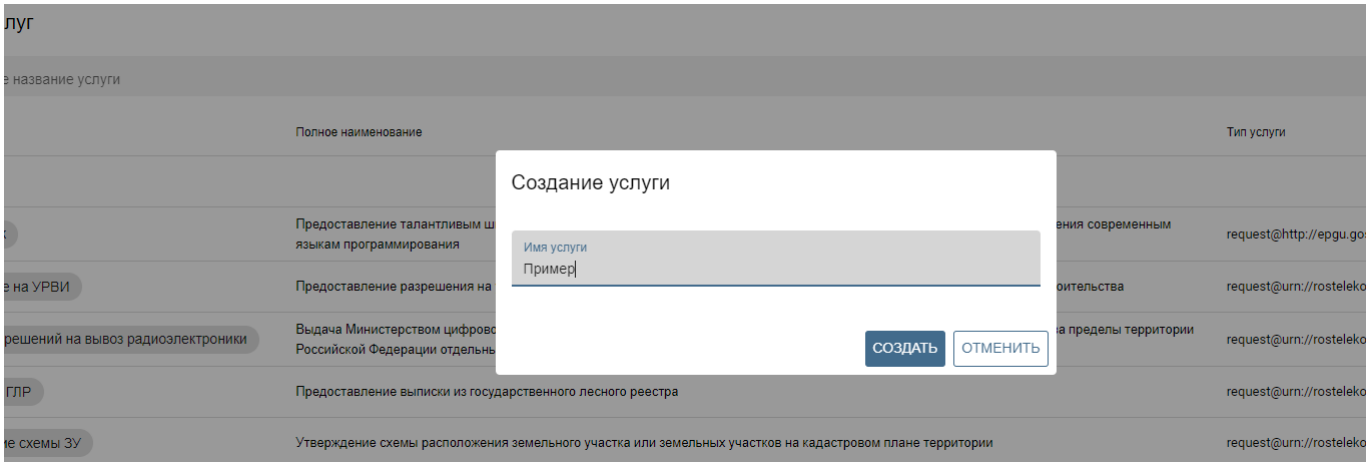


Рисунок 6 — Окно создания услуги

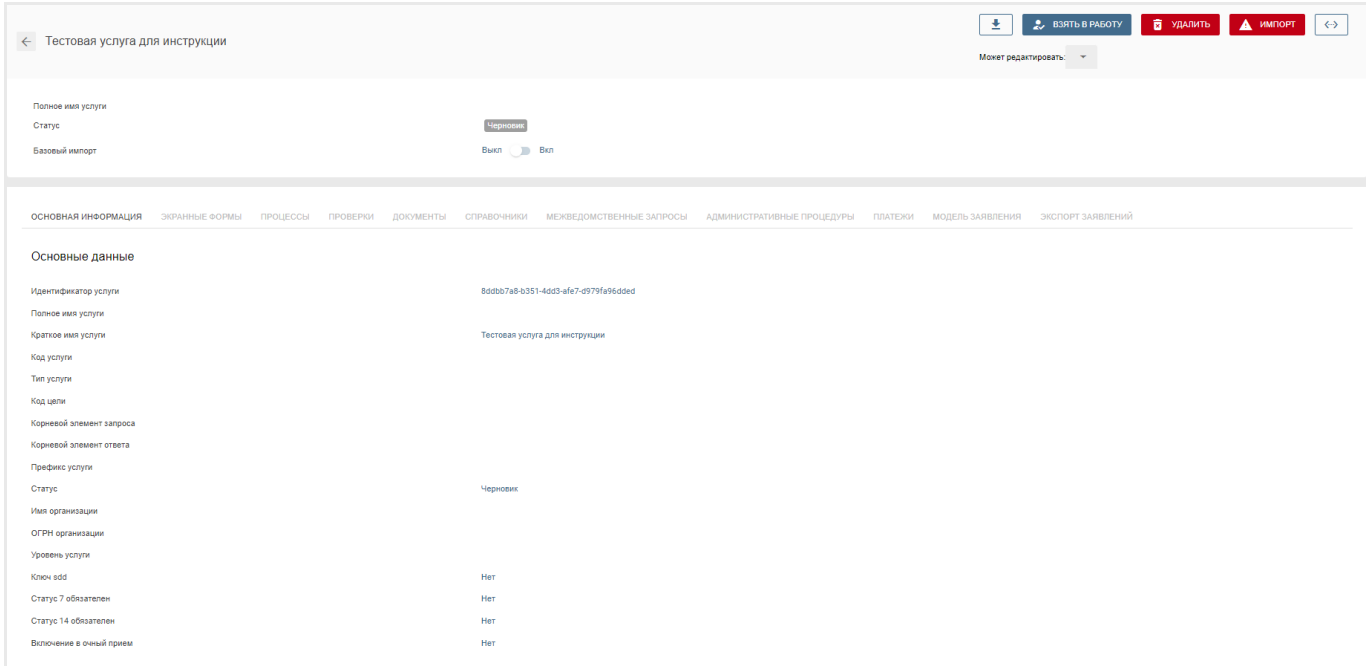


Рисунок 7 — Карточка услуги в статусе «Черновик». Вкладка «Основная информация»

Для того чтобы указать информацию об услуге и настроить процесс ее предоставления, необходимо изменить ее статус с «Черновик» на «В работе» нажатием на кнопку «Взять в работу».

До перевода услуги в статус «В работе» никакие действия по внесению информации и настройке процесса предоставления услуги невозможны, а услуга может быть удалена из ПГС при помощи кнопки «Удалить» в карточке услуги. После перевода услуги в статус «В работе»

становятся доступны внесение основной информации об услуге и настройка процесса ее предоставления, а удаление услуги из ПГС становится невозможным.

Также можно задавать список пользователей, которые смогут редактировать и настраивать услугу. Выбор производится посредством установки отметок напротив соответствующих пользователей в выпадающем списке, который появляется по нажатию на кнопку . Возможен выбор всех пользователей посредством выбора пункта «Все» (рисунок 8).

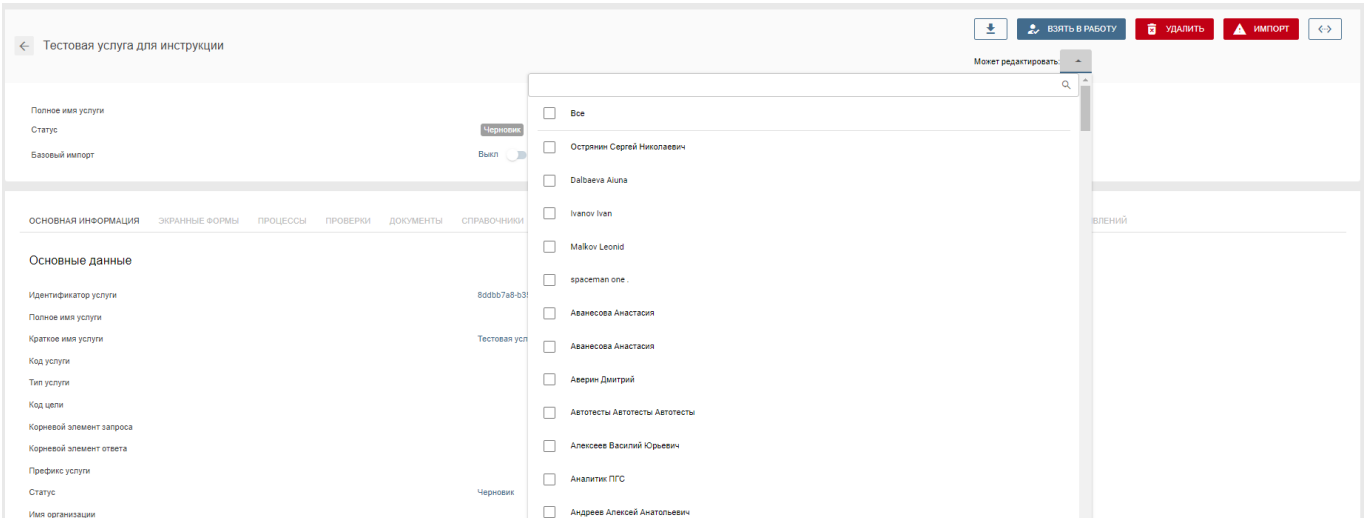


Рисунок 8 — Вкладка «Основная информация». Выбор пользователей, имеющих доступ к настройке услуги

3.3. Внесение основной информации об услуге

При наличии XML и XSD-схемы этапы, описанные в разделах 3.3–3.4, можно пропустить. Большинство данных будут автоматически сформированы при использовании функции «Импорт с ЕПГУ» (рисунок 20) и описаны в разделе 3.5.

Для того чтобы внести основную информацию об услуге, необходимо в карточке услуги на вкладке «Основная информация» нажать кнопку «Редактировать». После этого поля на вкладке становятся редактируемыми (рисунок 9).

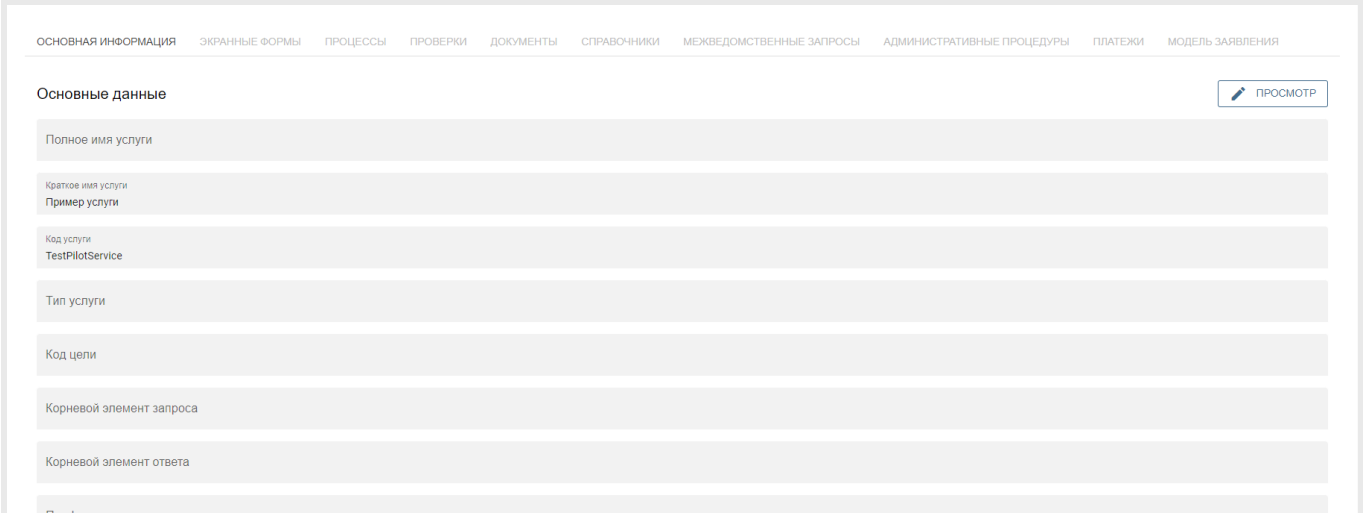


Рисунок 9 — Поля для внесения основной информации об услуге

Указываются сведения:

– «Полное имя услуги» — наименование услуги, указанное в документации на услугу или на портале https://lkuv.gosuslugi.ru ;

– «Краткое имя услуги» — наименование услуги, присваиваемое создателем или редактором услуги, позволяющее пользователям идентифицировать и находить ее;

– «Код услуги» — уникальное значение среди всех услуг в ПГС. Рекомендуется присваивать в виде <код услуги>_<номер версии услуги>.

Пример: ExtractForestRegistry_1_0_3

– «Тип услуги» — значение вида request@urn://rostelekom.ru/<код услуги>/<номер версии услуги>.

Пример: request@urn://rostelekom.ru/ExtractForestRegistry/1.0.3

– «Код цели» — значение, указанное в XML-файле заявления о предоставлении услуги, сгенерированном на ЕПГУ, в теге <tns:TargetId> (рисунок 10).

Пример: -10000115328

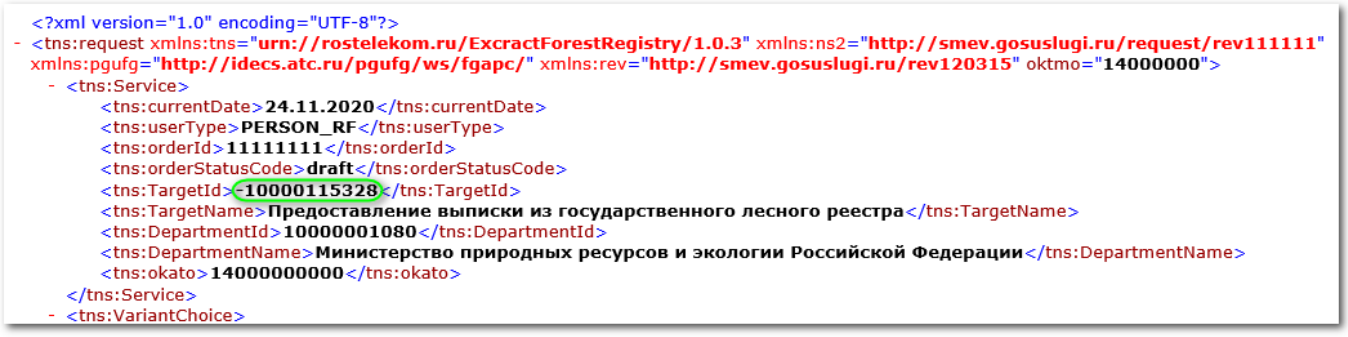


Рисунок 10 — Код цели в XML-файле заявления о предоставлении услуги

– «Корневой элемент запроса» — значение вида request@urn://rostelekom.ru/<код услуги>/<номер версии услуги>.

Пример: request@urn://rostelekom.ru/ExtractForestRegistry/1.0.3

– «Корневой элемент ответа» — значение вида FormResponse@urn://rostelekom.ru/<код услуги>/<номер версии услуги>.

Пример: FormResponse@urn://rostelekom.ru/ExtractForestRegistry/1.0.3

– «Префикс услуги» — любое значение, обозначающее услугу.

Используется в начале имен файлов и номеров, генерируемых при исполнении процесса предоставления услуги.

Рекомендуется указывать какую-либо аббревиатуру.

Пример: «ГЛР» для услуги «Предоставление выписки из государственного лесного реестра».

Поля «Имя организации», «ОГРН организации», «Уровень услуги», «Нормативно-правовые акты», «Жизненные ситуации», «Сферы» не заполняются.

Отметку «Ручные проверки обязательны» следует обязательно установить.

Отметки «hasSDD» и «Добавить форму заявления в очный прием» следует установить, если услуга предусматривает расчет доходов заявителя по предоставленным им сведениям для принятия решения по заявлению и очный прием заявителей с запросом о предоставлении услуги соответственно.

В таблицу «Список целей услуги» добавляются подуслуги, предусмотренные для услуги.

Для того добавления подуслуги, необходимо нажать кнопку «Добавить запись цели» (рисунок 11), в открывшемся окне указать код и название цели (полуслуги) и нажать кнопку «Добавить запись цели» (рисунок 12). Коды и наименования подуслуг содержатся в XSD-схеме услуги, в дочерних тегах <xsd:enumeration> и <xsd:documentation> тега <xsd:simpleType name=”GoalType”> соответственно (рисунок 13).

Данные действия необходимо выполнить для каждой подуслуги, имеющейся в услуге. Если у услуги отсутствуют подуслуги, то в качестве единственной подуслуги необходимо добавить саму услугу (полное наименование) с кодом «1».

Кнопки («Редактировать») и («Удалить») позволяют изменить данные о подуслугах и удалить подуслуги соответственно (рисунок 14).

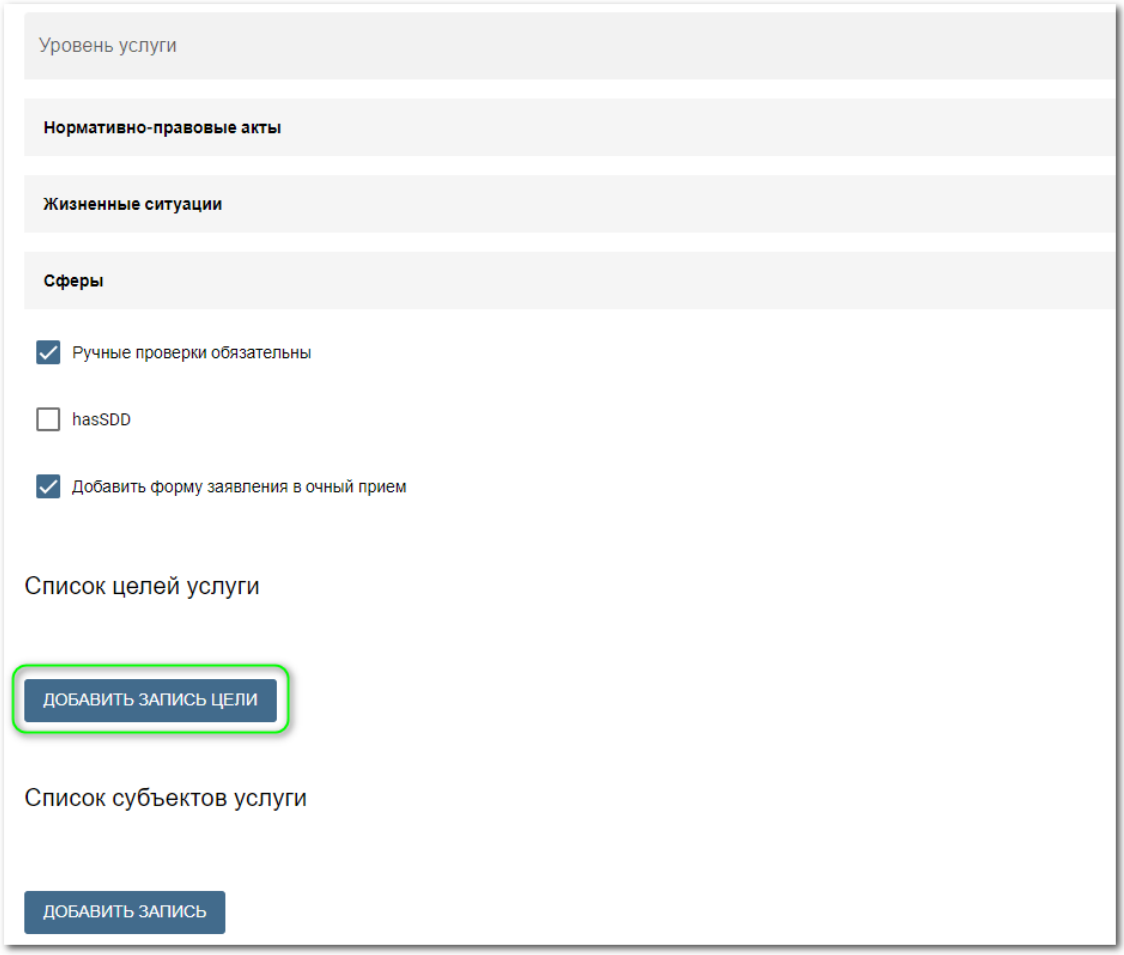


Рисунок 11 — Кнопка добавления подуслуги

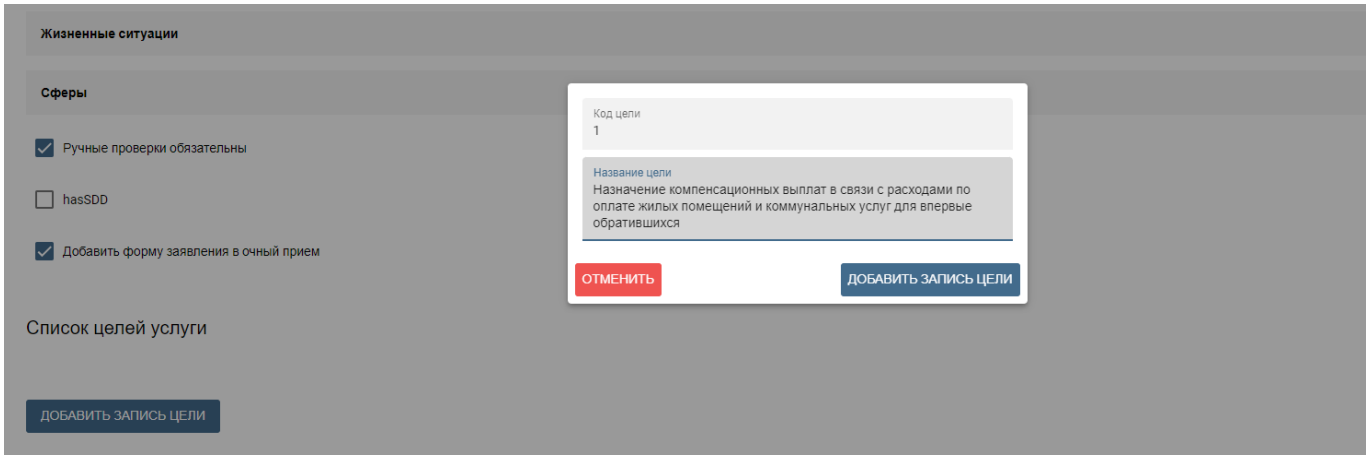


Рисунок 12 — Данные о подуслуге в окне добавления подуслуги

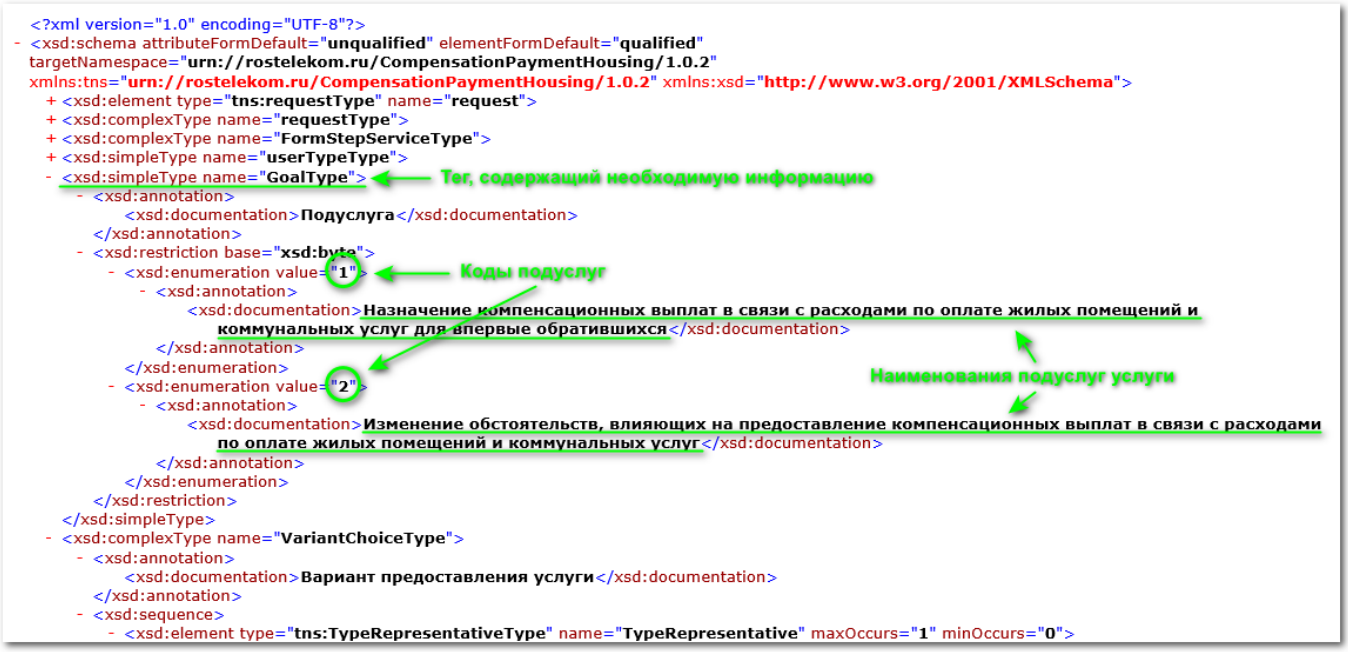


Рисунок 13 — Данные о подуслугах в XSD-схеме услуги

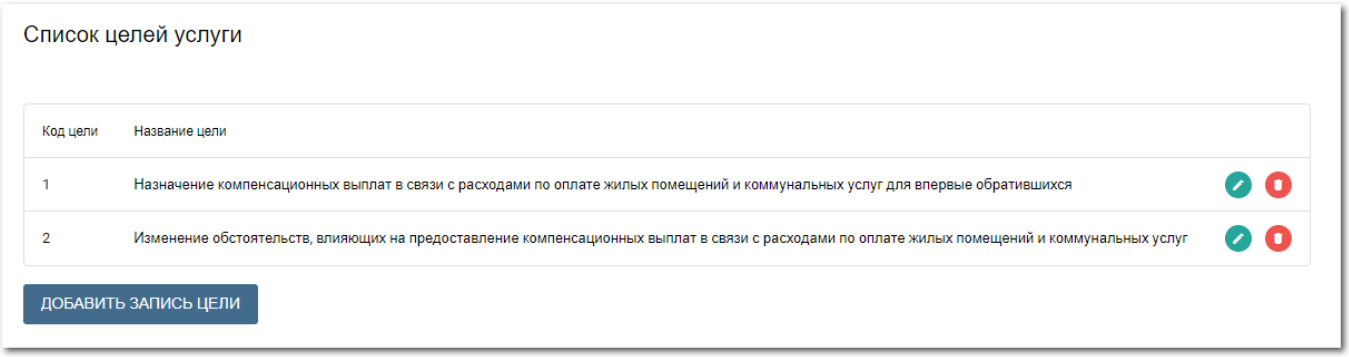


Рисунок 14 — Таблица целей (подуслуг) услуги

В таблицу «Список субъектов услуги» (рисунок 15) добавляются все субъекты, для которых процессом предоставления услуги предусмотрены проверки (см. подраздел 3.11). Для любой услуги субъектом проверки является заявитель. Наличие иных субъектов проверок зависит от настраиваемой услуги.

Действия по добавлению, редактированию и удалению субъектов услуги аналогичны действиям по добавлению, редактированию и удалению целей услуги.

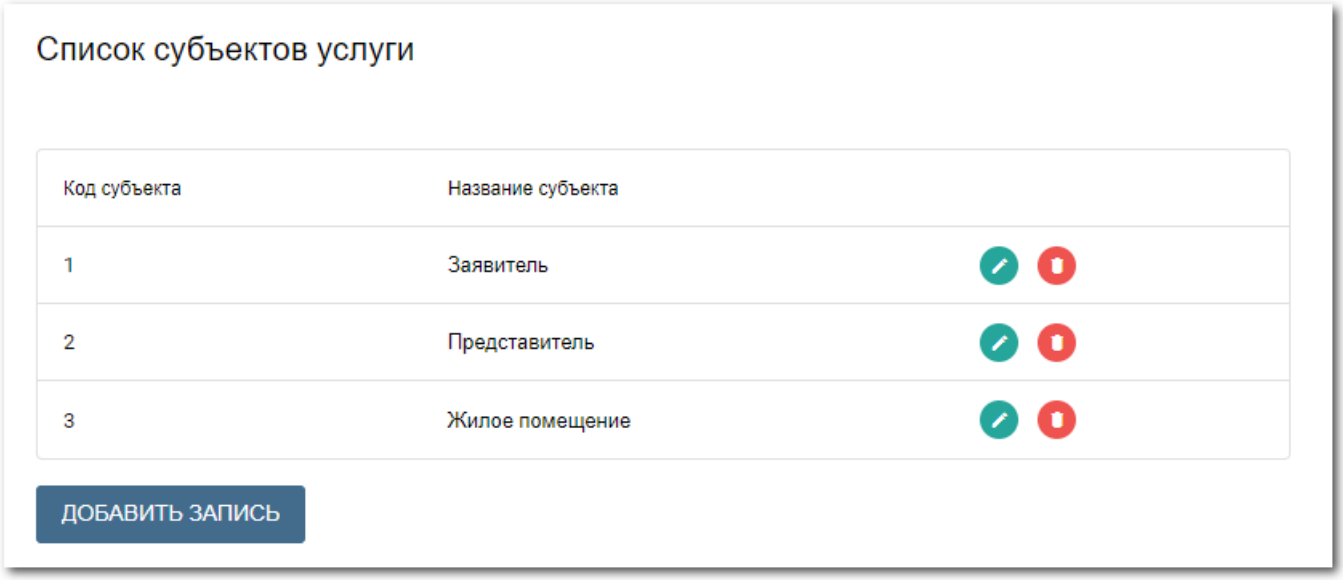


Рисунок 15 — Таблица субъектов услуги

По окончании внесения/редактирования информации следует нажать кнопку («Сохранить») для сохранения изменений.

Примечание. Кнопку («Сохранить») следует нажимать после добавления/изменения любых объектов на любой вкладке карточки услуги перед переходом на другую вкладку.

Примечание. Идентификатор услуги, указанный в одноименном поле (рисунок 7), присваивается ПГС автоматически и не допускает изменения.

3.4. Процессы. Операции с процессами

Во время исполнения процесса предоставления услуги выполняются отдельные процессы, которые создаются на вкладке карточки услуги «Процессы». В ПГС используются процессы трех стандартных типов[1]:

– процесс импорта (подразделы 3.5, 3.6);

– процесс межведомственного взаимодействия (подраздел 3.10.2);

– процесс подготовки данных (подраздел 3.13).

Процесс импорта — это процесс в рамках процесса предоставления услуги, который обрабатывает поступающие в ПГС заявления о предоставлении услуги в формате XML и заносит все данные из документа XML в ПГС для последующей работы с этими данными. Также процесс отвечает за маршрутизацию заявлений в надлежащее ведомство.

[1] В ПГС также предусмотрены, но не используются стандартные типы процессов «Базовый» и «Маршрутизация».

Процесс межведомственного взаимодействия (МВ) необходим для сборки запросов в ведомства, виды сведений которых зарегистрированы в СМЭВ.

Процесс подготовки данных (ППД) используется для осуществления автоматических проверок и формирования финальной таблицы проверок. Процесс должен быть обязательно добавлен в услугу, поскольку без него будет невозможно сформировать таблицу проверок, а также осуществить автоматические проверки пользователя о его принадлежности к категории лиц, которые имеют право на получение услуги.

Помимо стандартных, могут создаваться процессы пользовательских типов.

Процессы представляют собой бизнес-процессы в нотации BPMN 2.0. Процесс импорта может быть получен из XSD-схемы услуги (см. подразделы 3.5, 3.6). Прочие процессы (см. подразделы 3.10.2, 3.13) создаются как бизнес-процессы в нотации BPMN 2.0 во внешнем конструкторе (рекомендован Camunda Modeler) и загружаются в ПГС формате XML.

Добавление процессов в услугу осуществляется на вкладке карточки услуг «Процессы» (рисунок 16) при помощи кнопки «Новый процесс».

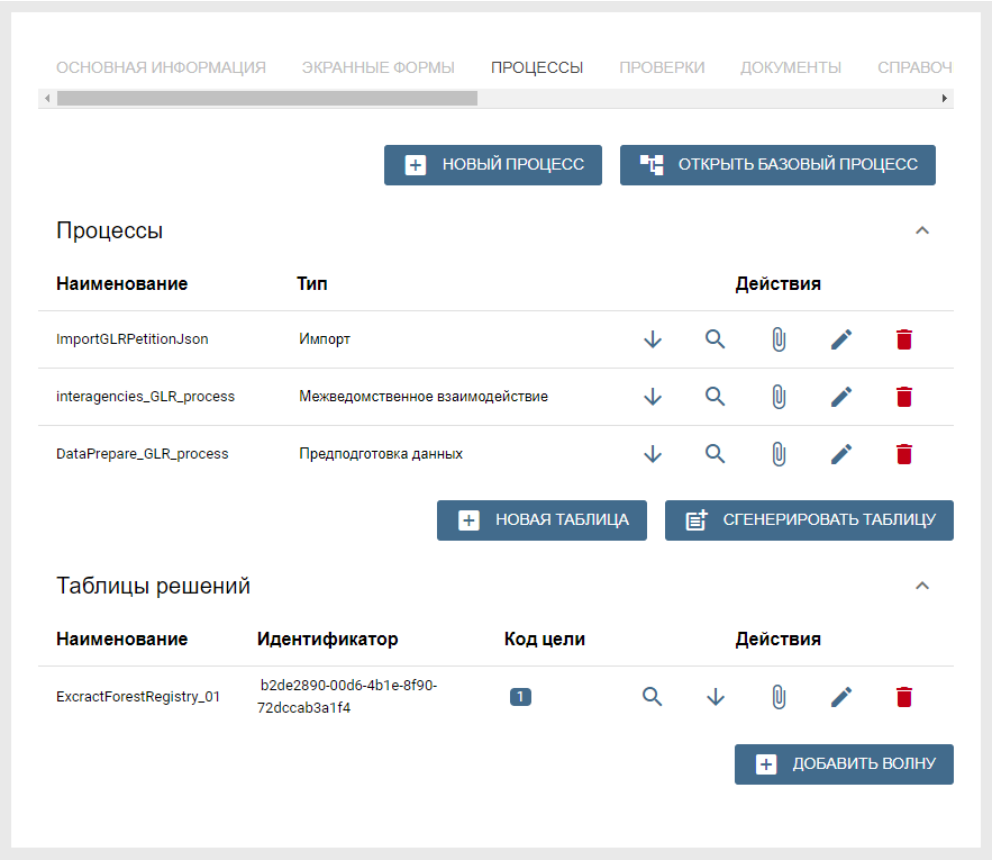


Рисунок 16 — Процессы на вкладке карточки услуги «Процессы»

При нажатии на данную кнопку открывается окно добавления процесса, представленное на рисунке 17, в котором:

– указывается имя процесса;

– выбирается стандартный (в выпадающем списке «Тип») или задается пользовательский тип процесса (переход от выпадающего списка типов процессов к полю для ввода пользовательского типа осуществляется при помощи кнопки «Свой»);

– загружается XML-файл процесса либо выбирается требуемый процесс среди хранящихся в ПГС.

Рисунок 17 — Окно добавления процесса в услугу

На вкладке «Процессы» возможны также следующие действия с процессами:

– скачивание процесса в формате XML — кнопка (↓) («Скачать»);

– просмотр и редактирование процесса во встроенном визуальном BPMN-конструкторе (рисунок 39) — кнопка (🔍) («Открыть в редакторе BPMN»);

– повторная загрузка XML-файла процесса в услугу (рисунок 18) — кнопка (📎) («Загрузить файл процесса»);

– редактирование процесса в формате XML в текстовом редакторе (рисунок 19) — кнопка (✎) («Редактировать»);

– удаление процесса из услуги — кнопка (🗑) («Удалить»).

Примечание. При редактировании процесса в формате XML (рисунок 19) не следует изменять имя и тип процесса.

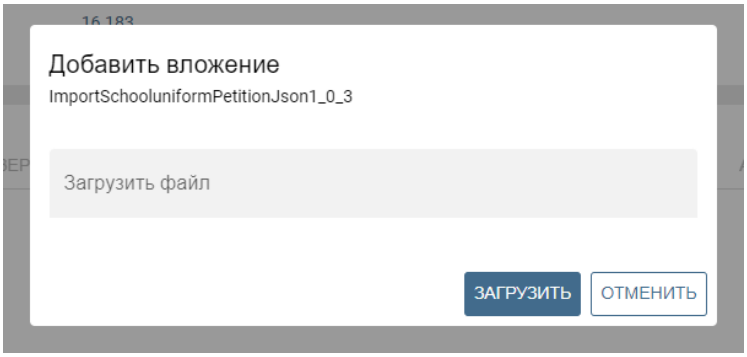


Рисунок 18 — Окно загрузки XML-файла процесса (с расширением .bpmn) в ПГС

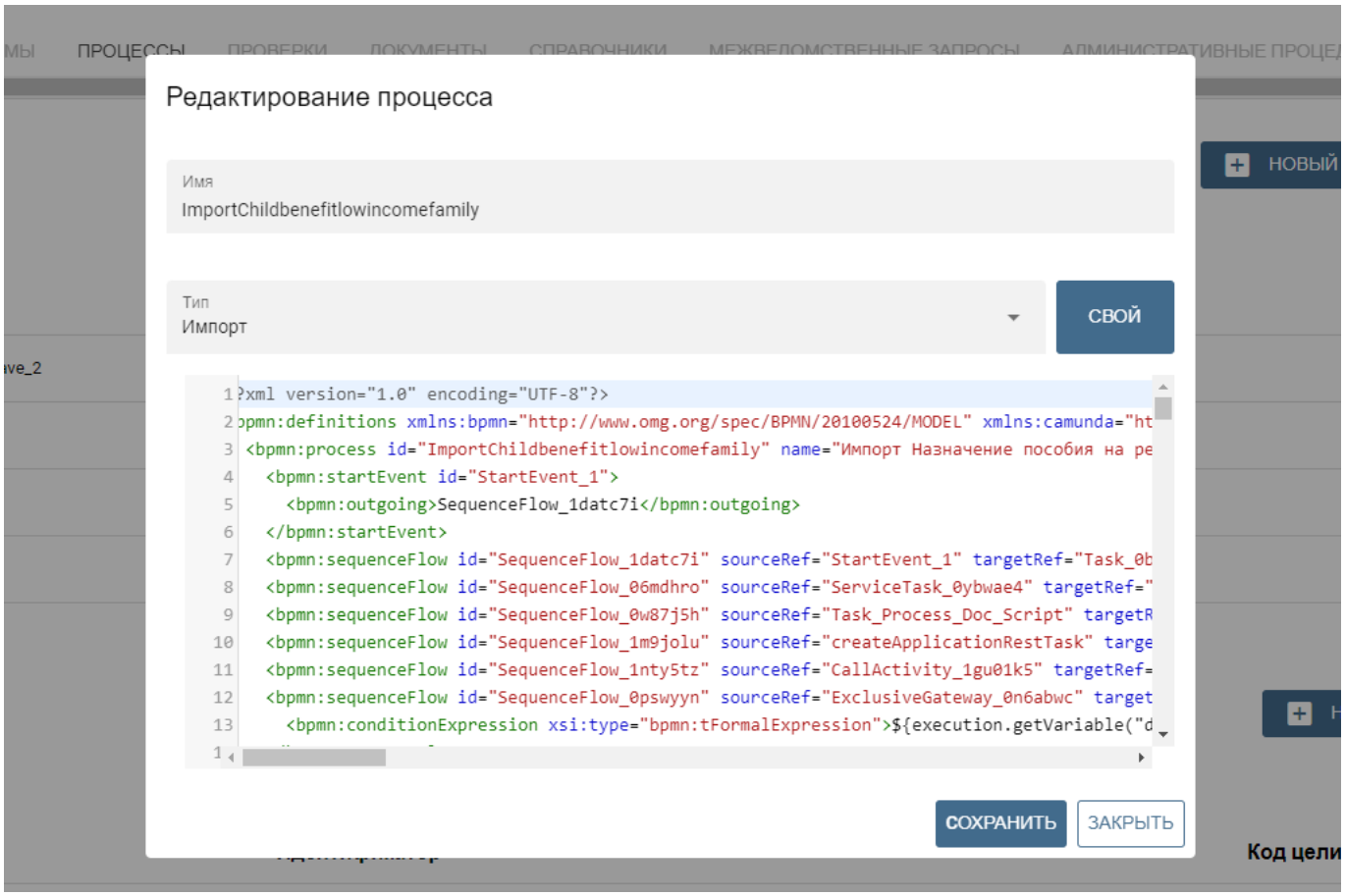


Рисунок 19 — Окно редактирования XML-кода процесса

3.5. Получение процесса импорта из XSD-схемы услуги

Для того чтобы автоматически получить основную информацию по услуге (см. подраздел 3.3) и процесс импорта (см. подраздел 3.4) из XSD-схемы услуги, необходимо в карточке услуги нажать кнопку («Импортировать из ЕПГУ») справа от заголовка карточки услуги (рисунок 20). После этого откроется страница, представленная на рисунке 21.

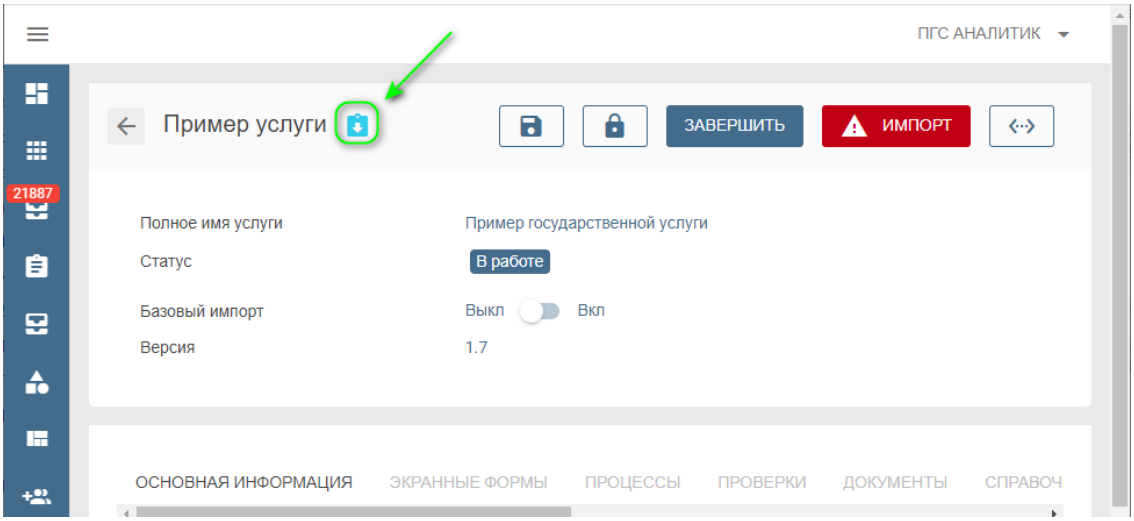


Рисунок 20 — Кнопка импорта из XSD-схемы услуги в карточке услуги

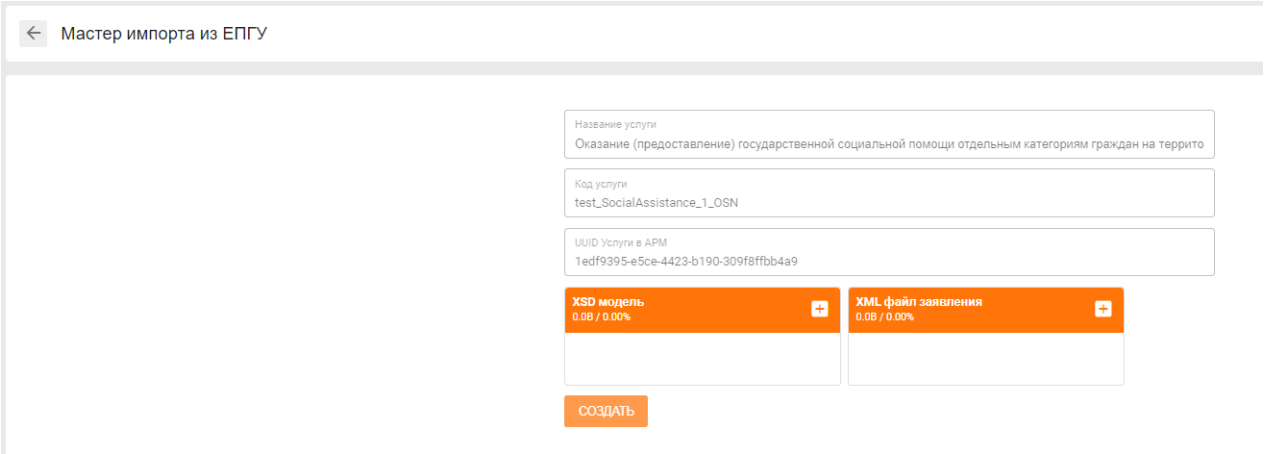


Рисунок 21 — Страница мастера импорта данных из ЕПГУ

Здесь необходимо нажать кнопку в блоке «XSD модель» и выбрать файл XSD-схемы услуги на компьютере. В случае успешной загрузки файла блок «XSD модель» примет вид, как на рисунке 22.

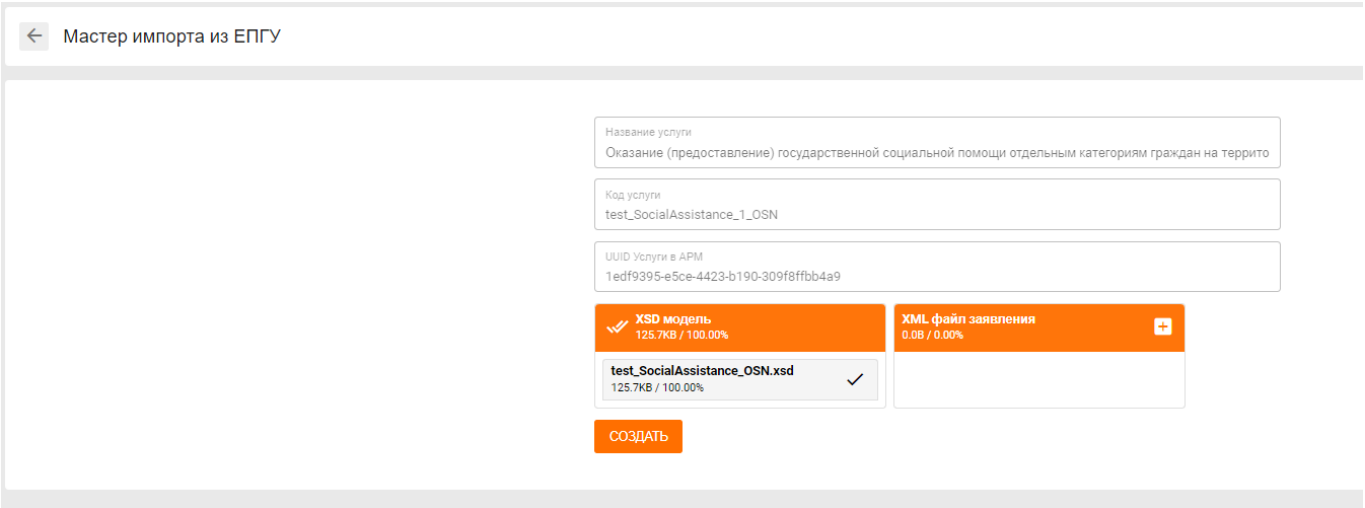


Рисунок 22 — Страница импорта данных из ЕПГУ. Загруженная XSD-схема услуги

Далее необходимо нажать кнопку в блоке «XML файл заявления» и выбрать XML-файл заявления на компьютере (рисунок 22). В случае успешной загрузки файла блок «XML файл заявления» примет вид, как на рисунке 23.

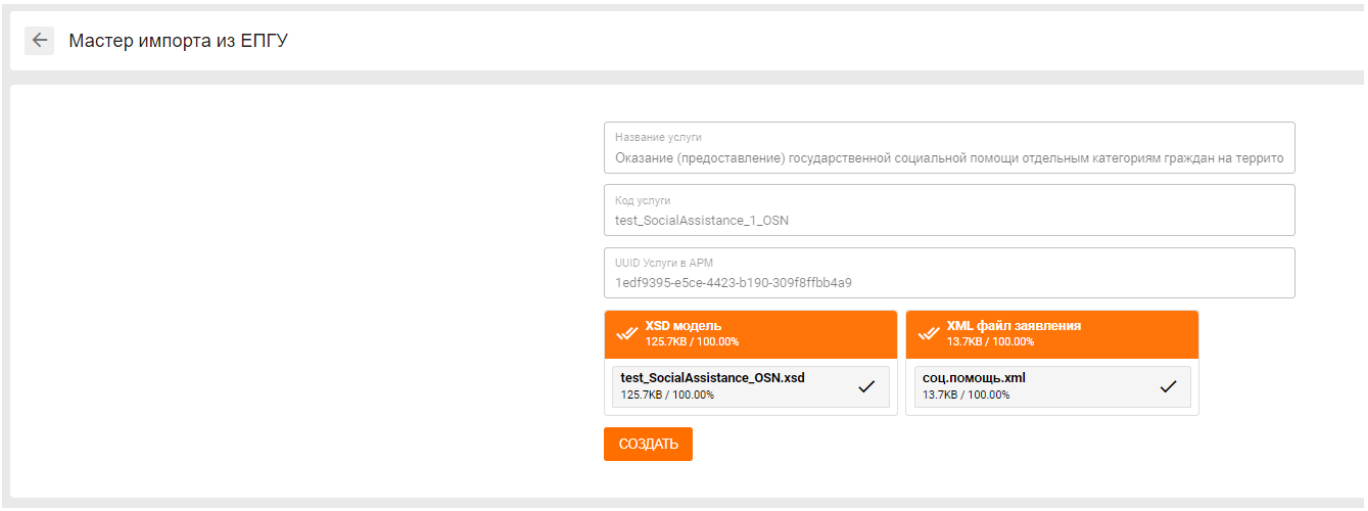


Рисунок 23 — Страница импорта данных из ЕПГУ. Загруженный XML файл заявления

Далее следует нажать кнопку «Создать». После этого отобразится страница, представленная на рисунке 24.

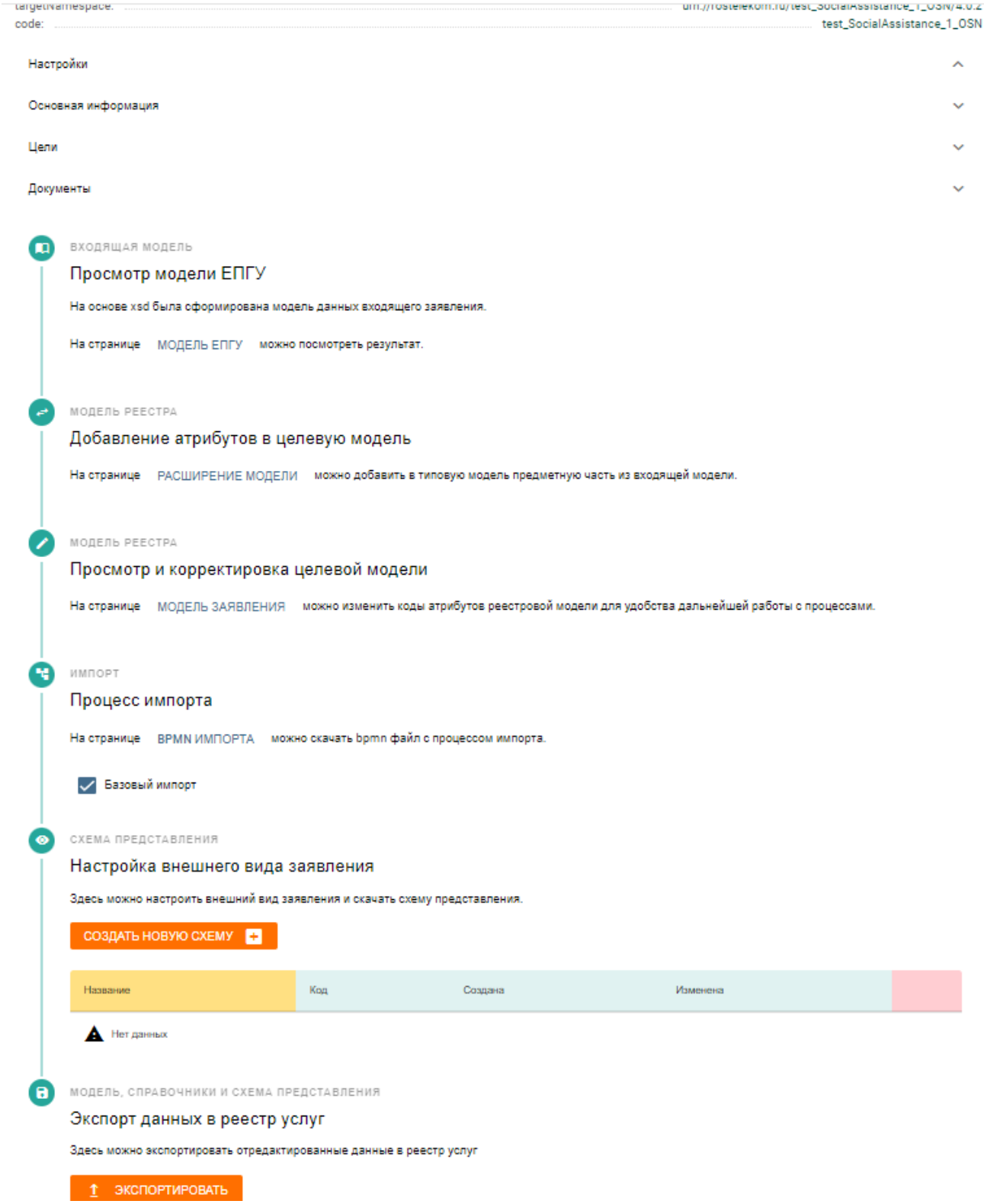


Рисунок 24 — Страница обработки данных, полученных из XSD-схемы услуги

Раздел «Настройки» содержит автоматически сформированные сущности, полученные из ЕПГУ: основную информацию по услуге, цели услуги, а также список входных документов, прилагаемых заявителем, которые можно посмотреть, раскрыв списки (рисунок 25).

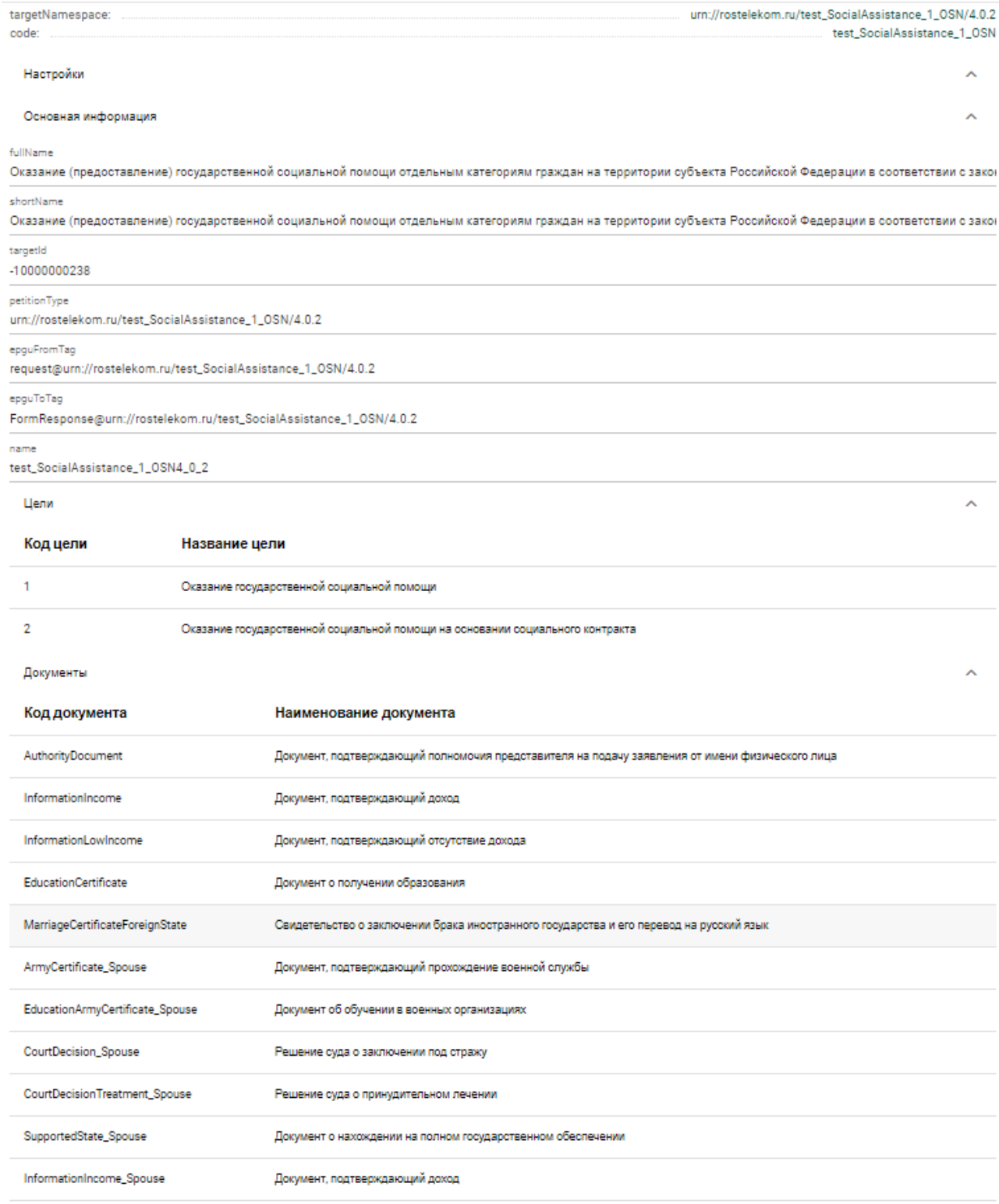


Рисунок 25 — Раздел «Настройки»

Раздел «Просмотр модели ЕПГУ» позволяет просмотреть загруженную модель данных ЕПГУ в виде наглядной структуры (рисунок 26). При этом доступны операции:

– отображения сведений целиком — кнопка («Развернуть все»);

– просмотра модели данных в форматах XML и JSON — кнопки ( ) («Показать модель») и ( ) («Показать модель JSON») соответственно;

– скачивания модели данных на компьютер — кнопка ( ) («Скачать все словари в формате *.zip»).

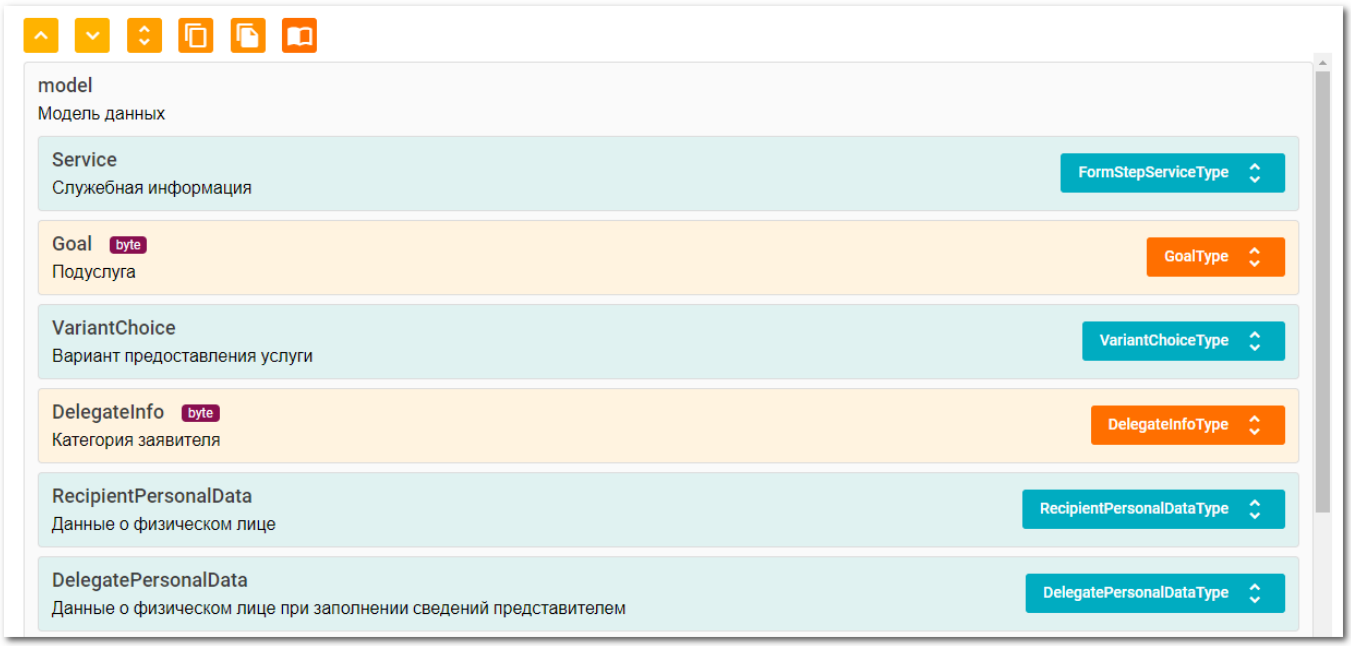


Рисунок 26 — Модель данных ЕПГУ в детальном иерархическом представлении в свернутом виде

В разделе «Добавление атрибутов в целевую модель» следует выбрать атрибуты, которые должны присутствовать в целевой модели услуги (рисунок 27). Операция выполняется следующим образом: необходимо из правой колонки удалить с помощью кнопок () («Удалить») неиспользуемые атрибуты, присутствующие в ней по умолчанию (все, кроме statement), и перенести посредством перетаскивания мышью из левой колонки в правую (в переменную petition) те атрибуты модели ЕПГУ, которые должны присутствовать в целевой модели. Тем самым устанавливается соответствие между атрибутами заявления на ЕПГУ и атрибутами заявления в ПГС. Переносить следует только уникальные для данной услуги атрибуты, т. е. такие, которые отсутствуют в предустановленной в ПГС таблице соответствия[1] атрибутов. По завершении данных действий следует нажать кнопку ( ) («Сохранить изменения целевой модели»), затем вернуться на страницу обработки данных, полученных из XSD-схемы услуги, нажав кнопку ( ) («Назад»).

[1] В интерфейсе ПГС может использоваться термином «маппинг» для обозначения таблицы соответствия.

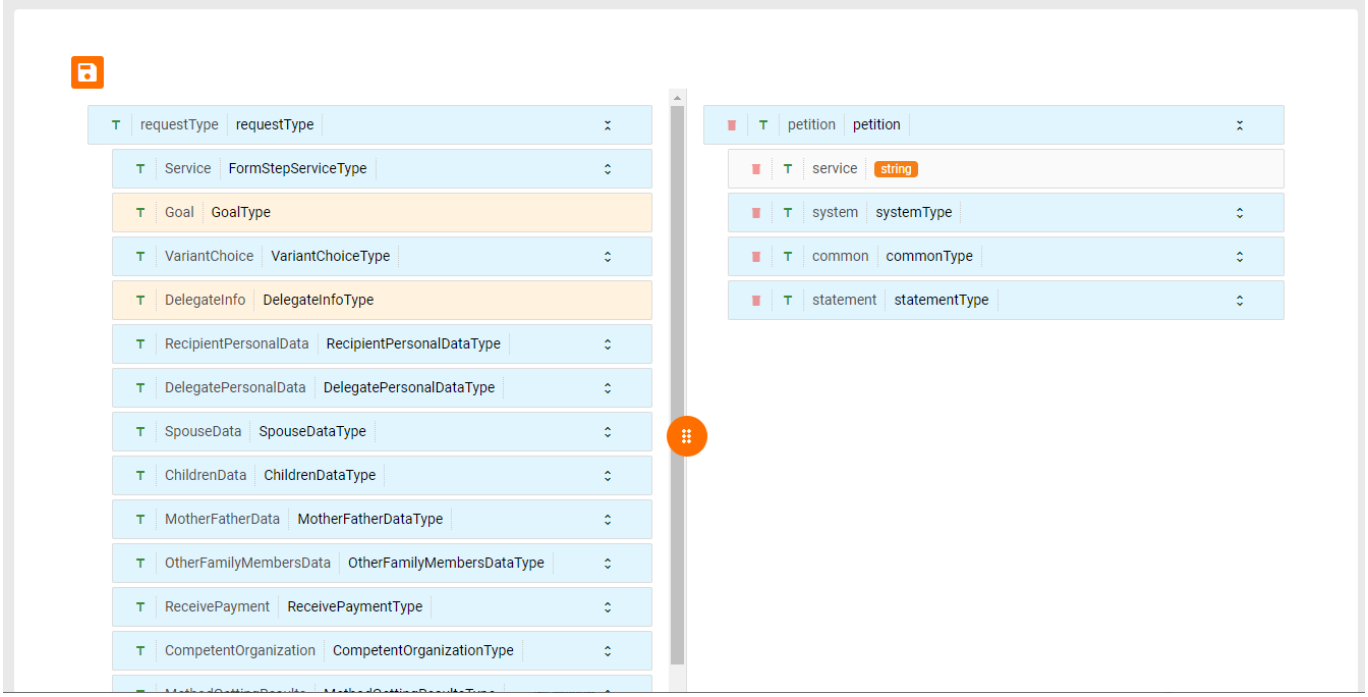
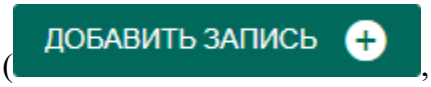
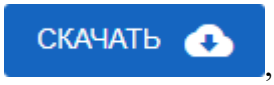


Рисунок 27 — Выбор атрибутов целевой модели

Раздел «Просмотр и корректировка целевой модели» позволяет просмотреть сформированную целевую модель в виде наглядной структуры, проверить ее целостность и при необходимости отредактировать ее атрибуты (рисунок 28). Интерфейс раздела аналогичен интерфейсу раздела «Просмотр модели ЕПГУ». При этом также доступны кнопки создания (— «Создать комплексный тип», — «Создать простой тип или enumeration») и удаления (— «Удалить атрибут») атрибутов, добавления и скачивания, редактирования и удаления записей (

, 

, — «Редактировать запись», — «Удалить запись» соответственно). Данный раздел обычно не используется.

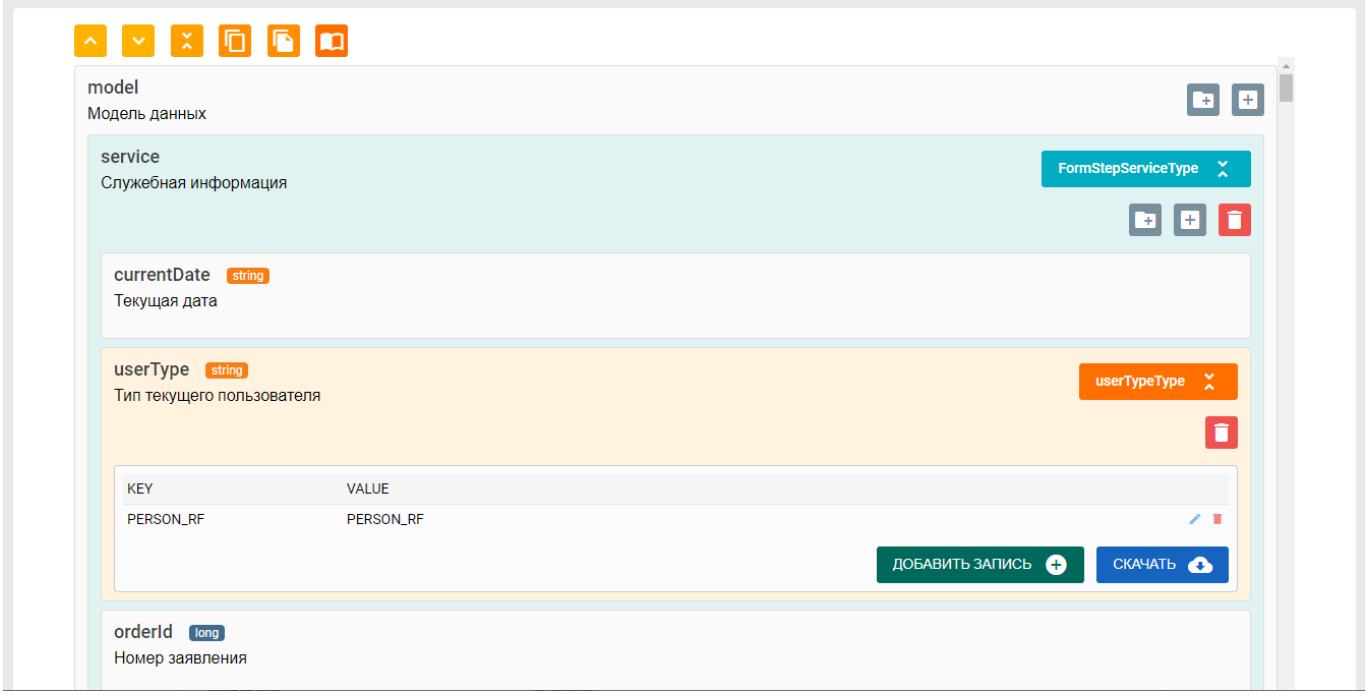


Рисунок 28 — Целевая модель услуги
в детальном иерархическом представлении в развернутом виде

Раздел «Процесс импорта» позволяет просмотреть сформированную таблицу соответствия атрибутов в формате JSON и скачать сформированный BPMN-процесс импорта на компьютер в формате XML (рисунок 29). Проставив соответсвующий чек-бокс, можно перейти к выполнению базового импорта (подраздел 3.6.2).

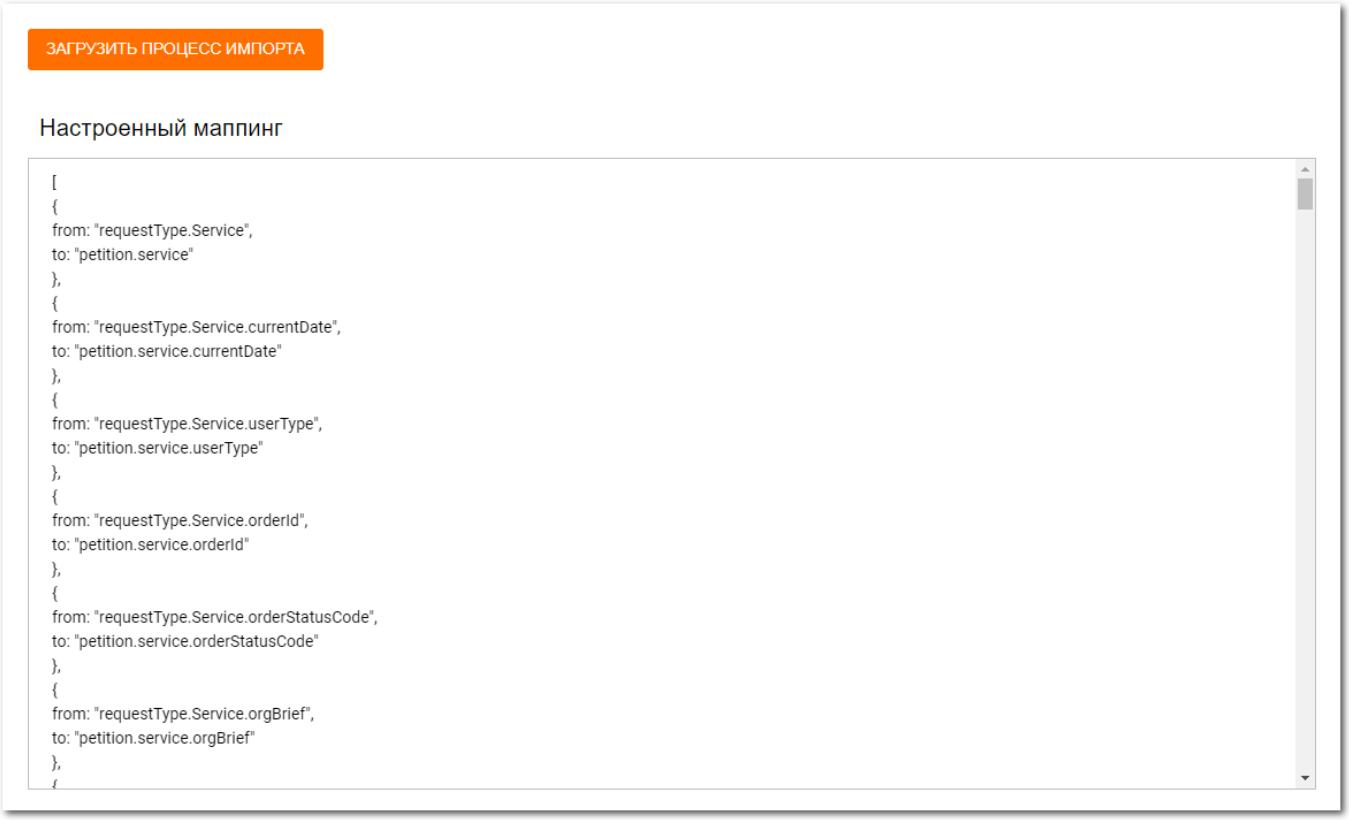


Рисунок 29 — Страница просмотра настроенной таблицы соответствия атрибутов и скачивания BPMN-процесса импорта

В разделе «Настройка внешнего вида заявления» следует выполнить настройку экранной формы заявления о предоставлении услуги и, при необходимости, создать экранную форму очного приема заявителя, которые будут использоваться для данной услуги в ПГС.

Операция выполняется следующим образом:

1. Необходимо нажать «Создать новую схему» (рисунок 30).

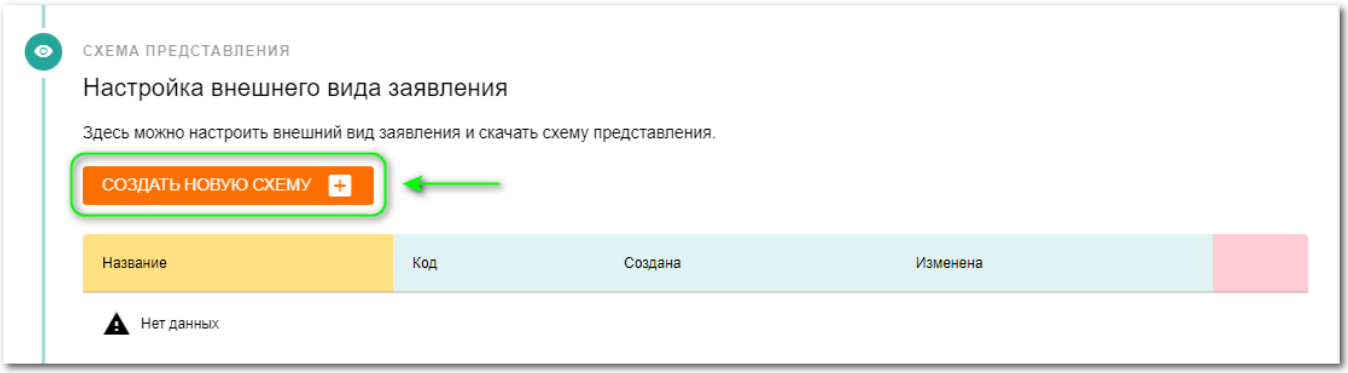


Рисунок 30 — Страница обработки данных, полученных из XSD-схемы услуги. Кнопка создания экранных форм заявления и очного приема заявителя

2. В открывшемся окне ввести название и код экранной формы (рисунок 31).

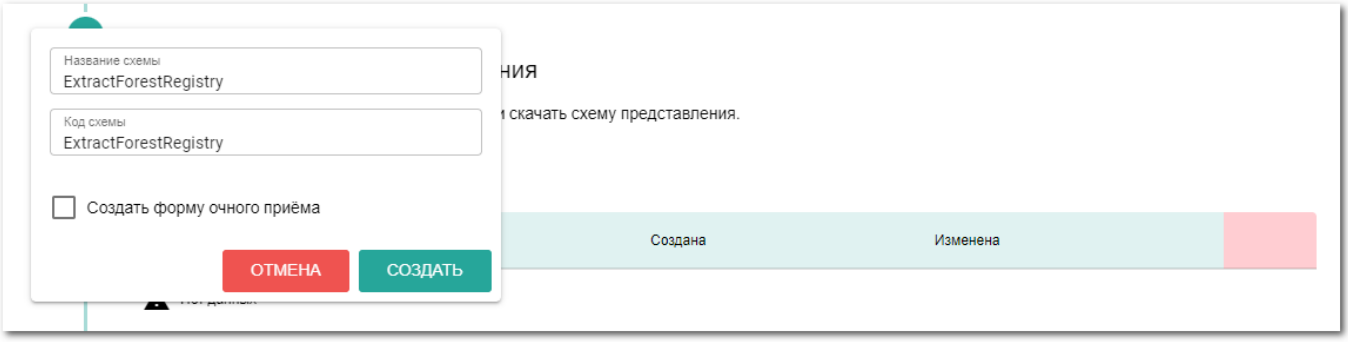


Рисунок 31 — Окно создания экранных форм заявления и очного приема заявителя

3. Если требуется создать экранную форму очного приема заявителя, то установить отметку «Создать форму очного приема» (рисунок 32).

4. Нажать кнопку «Создать» (рисунок 32).

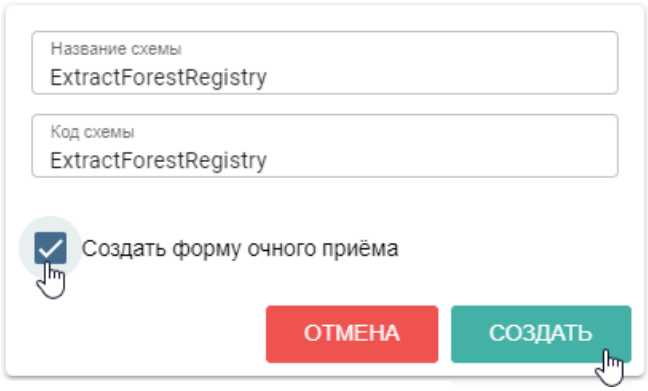


Рисунок 32 — Окно создания экранных форм заявления и очного приема заявителя. Установленная отметка «Создать форму очного приема» и кнопка «Создать»

5. На открывшейся странице в правой колонке отредактировать атрибуты, которые должны использоваться для экранной формы заявления в ПГС (рисунок 33). Для этого использовать кнопки:

– — редактировать атрибут/группу;

– — создать новый атрибут;

– — создать новую группу;

– — удалить атрибут/группу.

Кроме того, можно добавлять в правую колонку доступные элементы из левой колонки посредством перетаскивания мышью.

Переключатель «JSON» позволяет просмотреть/отредактировать код, задающий экранную форму заявления, в формате JSON.

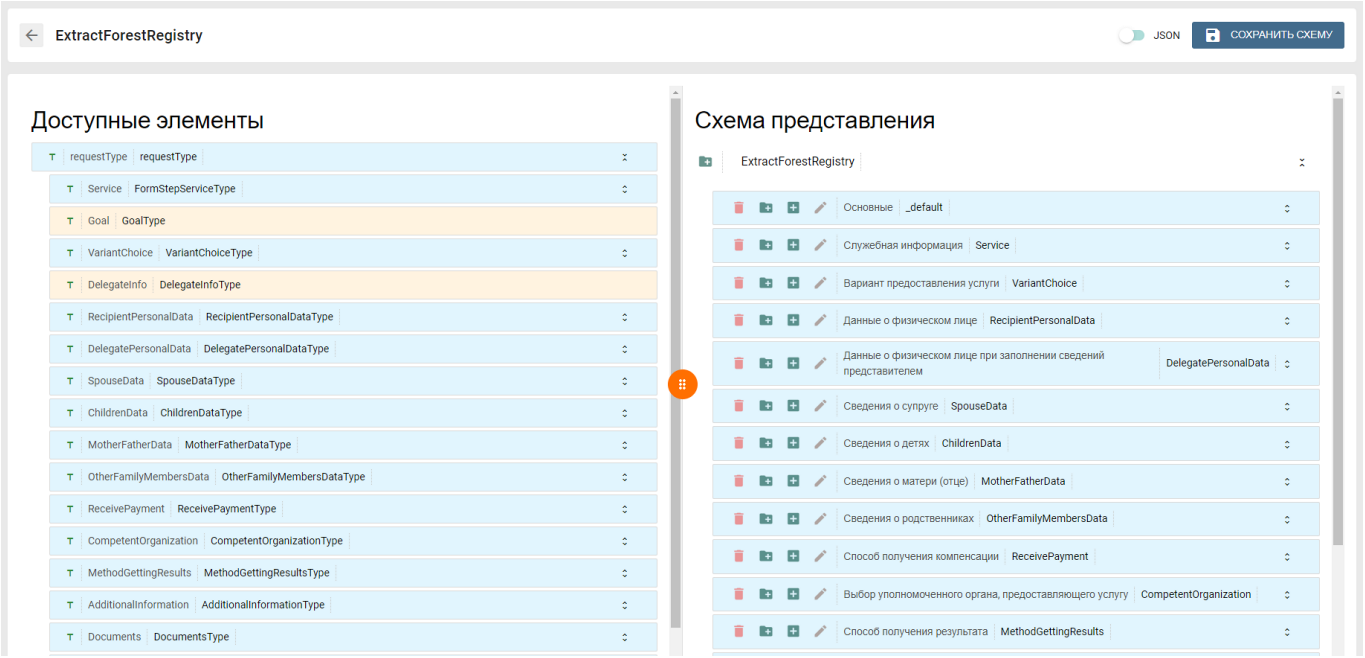


Рисунок 33 — Страница настройки схемы представления (экранной формы заявления)

6. Нажать кнопку «Сохранить схему», затем вернуться на страницу обработки данных, полученных из XSD-схемы услуги, нажав кнопку («Назад»).

После этого на странице обработки данных, полученных из XSD-схемы услуги, появится созданная схема представления (экранная форма заявления) — рисунок 34. Для нее доступны операции:

– скачать схему в формате JSON — кнопка («Скачать JSON»);

– просмотреть схему в формате JSON — кнопка («Просмотреть схему»);

– редактировать схему — нажать на строку, в которой отображаются данные о схеме;

– удалить схему — кнопка («Удалить схему»).

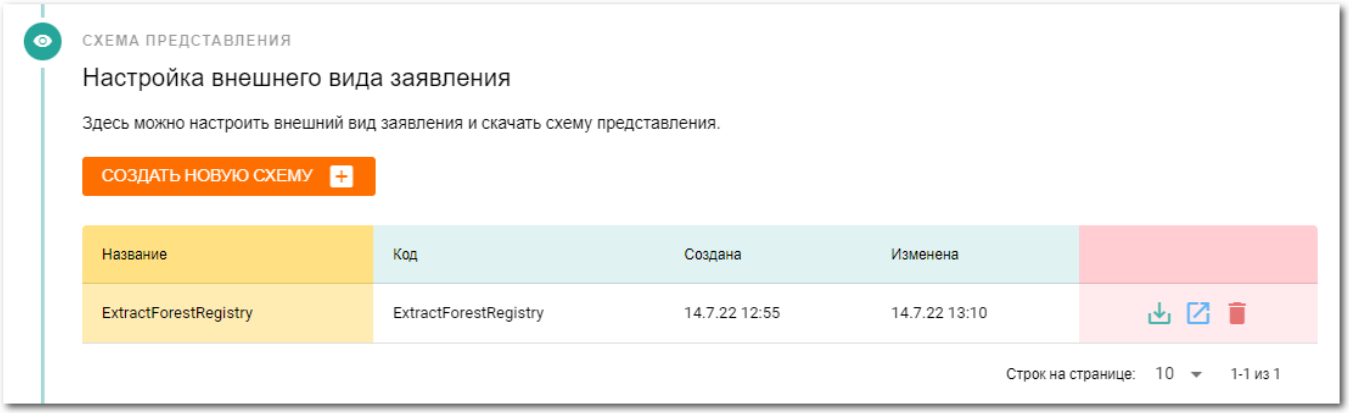


Рисунок 34 — Созданная экранная форма заявления
на странице обработки данных, полученных из XSD-схемы услуги

В разделе «Экспорт данных в реестр услуг» следует экспортировать полученные данные в ПГС.

Операция выполняется следующим образом:

1. Нажать кнопку «Экспортировать» (рисунок 35).

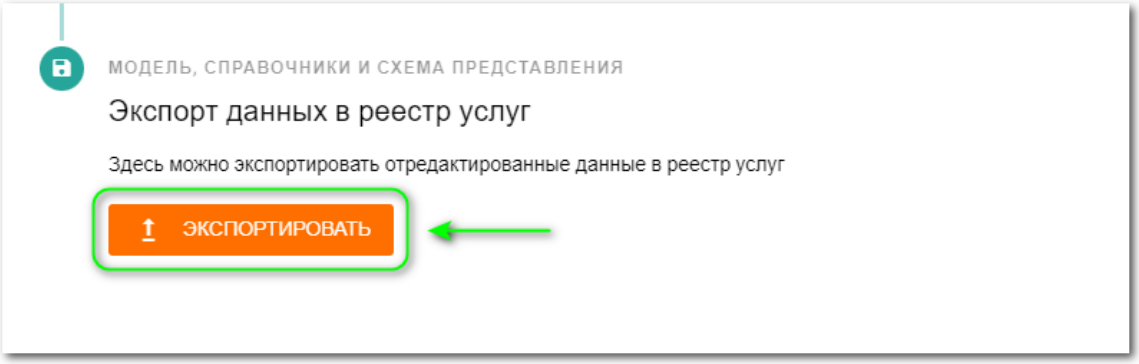


Рисунок 35 — Страница обработки данных, полученных из XSD-схемы услуги. Кнопка экспорта данных в ПГС

2. В открывшемся окне установить все отметки, выбрать в выпадающем списке созданную схему представления, затем нажать кнопку «Экспортировать» (рисунок 36).

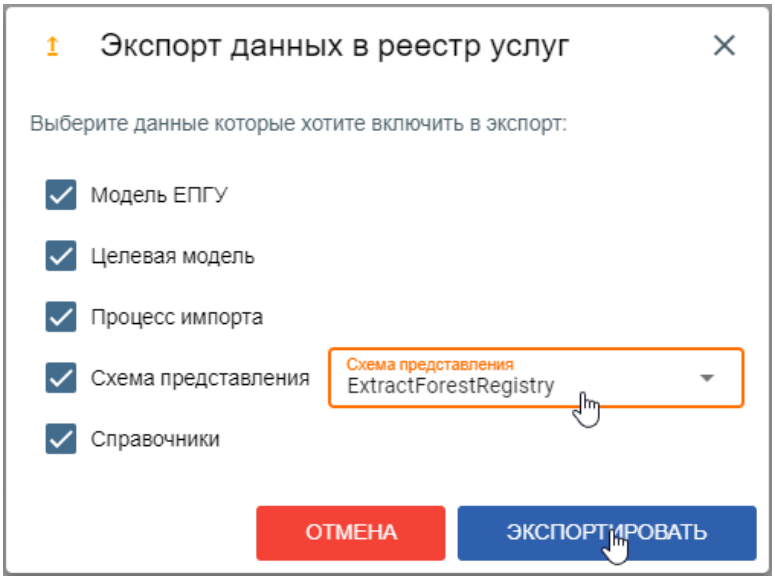


Рисунок 36 — Окно экспорта данных в ПГС

3. Дождаться завершения процесса экспорта до появления окна, изображенного на рисунке 37.

4. Нажать кнопку «Вернуться на платформу ГС».

После этого произойдет переход на вкладку «Процессы» карточки услуги (откроется в новой вкладке браузера), на которой будет присутствовать полученный процесс импорта (см. рисунок 38 и подраздел 3.6).

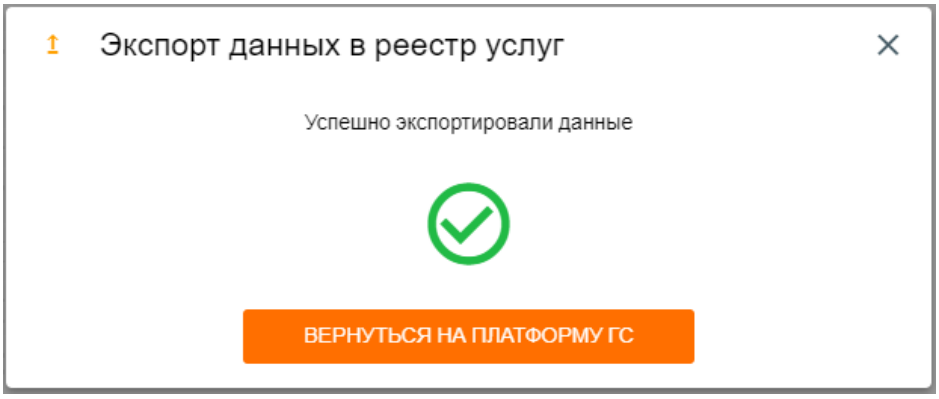


Рисунок 37 — Окно, уведомляющее об успешности экспорта обработанных данных XSD-схемы услуги в ПГС

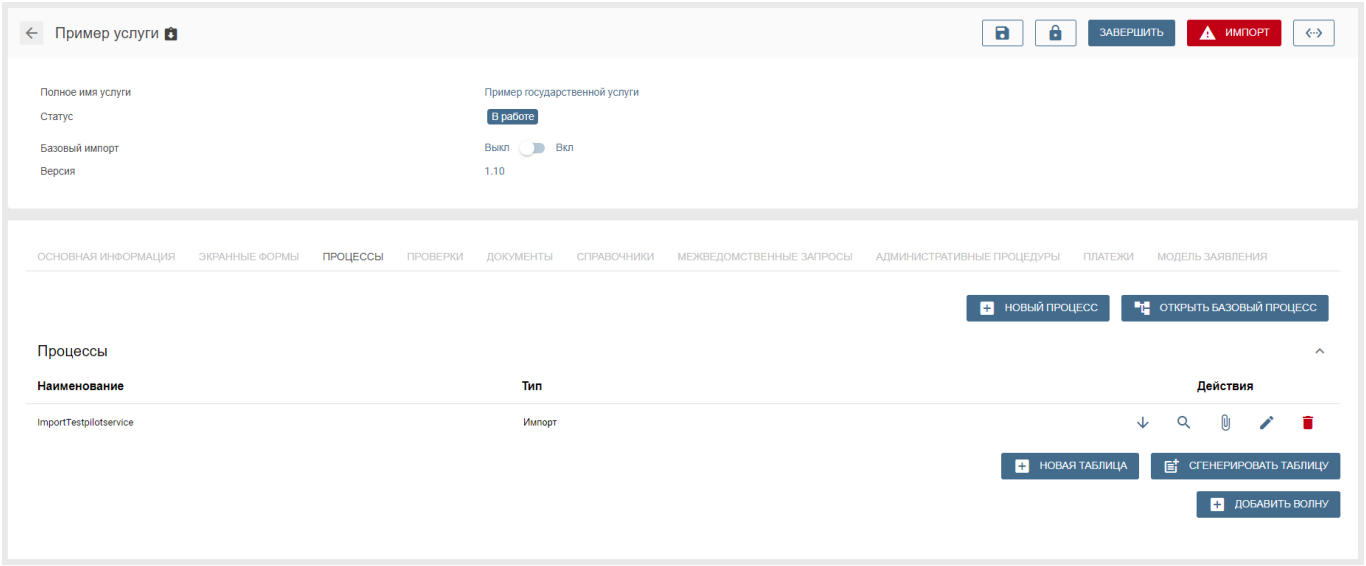


Рисунок 38 — Процесс импорта на вкладке карточки услуги «Процессы»

3.6. Просмотр и редактирование процесса импорта

3.6.1. Настраиваемый процесс импорта

После выполнения действий, описанных в подразделе 3.5, в таблице процессов на вкладке карточки услуги «Процессы» появляется стандартный процесс импорта, который может быть донастроен лицом, настраивающим процесс предоставления услуги, при необходимости.

Для того чтобы открыть для просмотра процесс импорта, необходимо перейти на данную вкладку и нажать кнопку ( ) («Открыть в редакторе BPMN»), соответствующую данному процессу. После этого отобразится схема процесса в нотации BPMN 2.0 во встроенном конструкторе (рисунок 39).

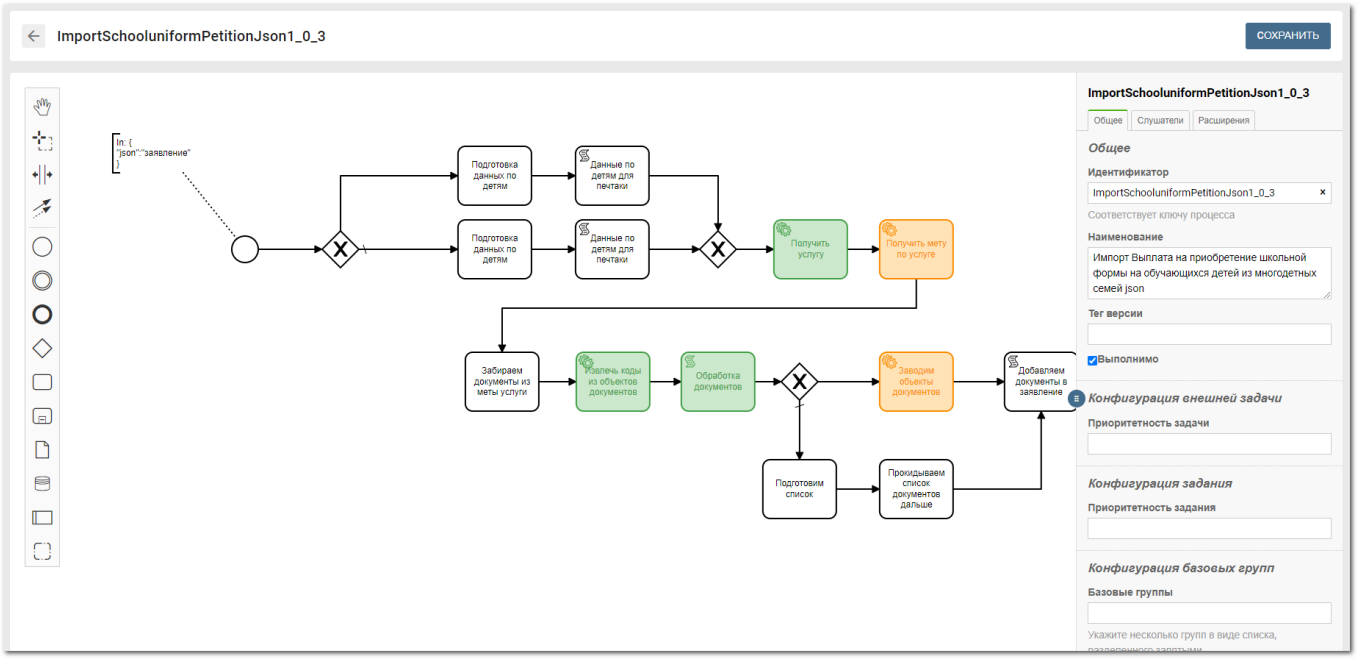


Рисунок 39 — BPMN-схема стандартного процесса импорта
во встроенном BPMN-конструкторе ПГС

С левой стороны расположена BPMN-схема процесса, которую можно изменять с помощью инструментов конструктора. Справа от BPMN-схемы процесса присутствует панель параметров процесса.

Настроенная таблица соответствия атрибутов содержится в элементе «Собираем бизнес-данные». Для просмотра таблицы соответствия необходимо:

1. Нажать мышью на элемент «Собираем бизнес-данные».

2. На панели параметров элемента справа от схемы процесса перейти на вкладку «Ввод/Вывод» (рисунок 40).

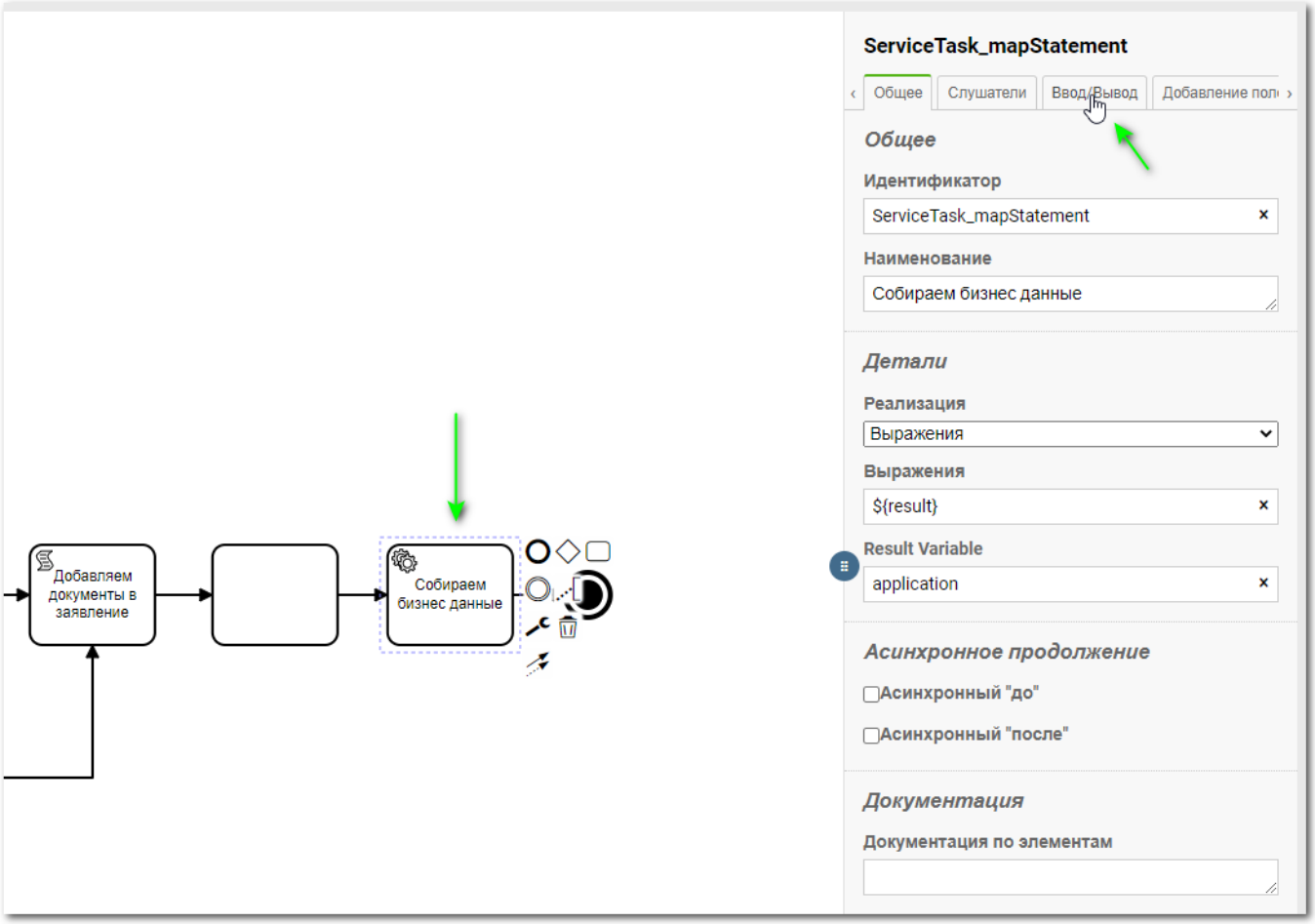


Рисунок 40 — Переход на вкладку «Ввод/Вывод»
на панели параметров элемента процесса импорта «Собираем бизнес-данные»

3. В поле «Входящие параметры» выделить переменную fields.

После этого внизу панели параметров отобразится таблица соответствия атрибутов (рисунок 41).

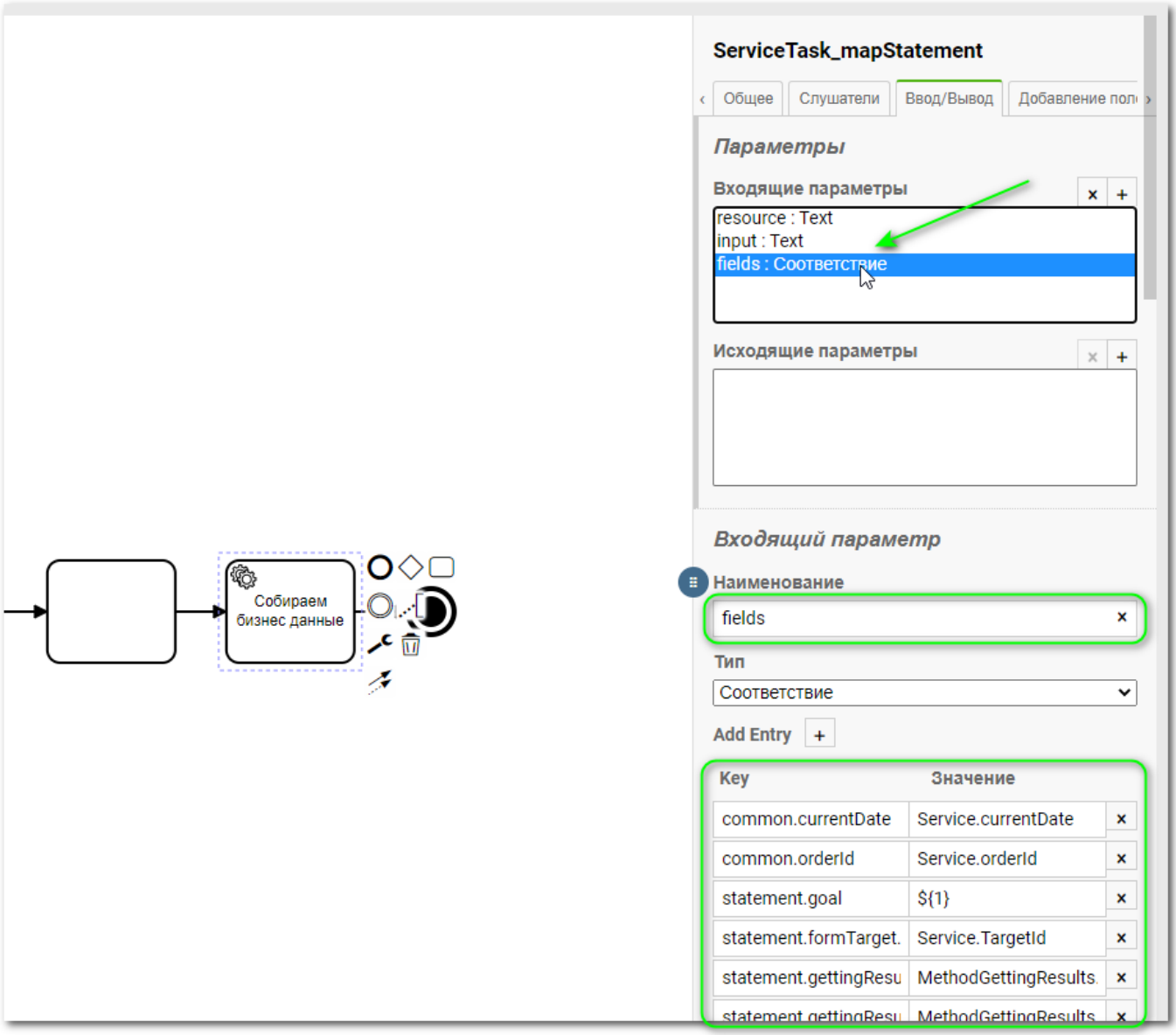


Рисунок 41 — Таблица соответствия атрибутов в переменной fields на вкладке «Ввод/Вывод» панели параметров элемента процесса импорта «Собираем бизнес-данные»

Атрибуты XSD-схемы содержатся в переменной petition и на вкладке «Ввод/Вывод» панели параметров элемента «Собираем бизнес-данные» отображаются в правой колонке. В левой колонке находятся соответствующие атрибуты, используемые в описании (коде) заявления в ПГС. Таким образом, с помощью таблицы соответствия значения атрибутов из XML-файлов заявлений переносятся в соответствующие описания заявлений в ПГС для дальнейшей работы с ними.

Пример таблицы соответствия атрибутов приведен на рисунках 42–44.

```
        <tns:SubstanceNameManual>растворитель</tns:SubstanceNameManual
        <tns:SubstanceInfo>тест</tns:SubstanceInfo>
        <tns:SubstanceForm>мазь</tns:SubstanceForm>
        <tns:SubstanceRecal>2 литра</tns:SubstanceRecal>
    </tns:MixtureData>
  - <tns:SubstanceData>
      - <tns:SubstanceDataBlock>
            <tns:SubstanceTradeName>тестовое название</tns:SubstanceTrad
            <tns:SubstanceForm>тестовая форма</tns:SubstanceForm>
            <tns:SubstanceUnit>нг</tns:SubstanceUnit>
            <tns:Amount>45</tns:Amount>
        </tns:SubstanceDataBlock>
    </tns:SubstanceData>
    <tns:ImportData>
        <tns:CountryImport>АВСТРИЯ</tns:CountryImport>
        <tns:Goal>продажа</tns:Goal>
        <tns:StartDate>12.11.2021</tns:StartDate>
        <tns:EndDate>26.11.2021</tns:EndDate>
    </tns:ImportData>
  - <tns:CustomsData>
        <tns:Transport>авиаперевозка</tns:Transport>
        <tns:Checkpoint>домодедово</tns:Checkpoint>
    </tns:CustomsData>
  - <tns:ManufacturerData>
      - <tns:ManufacturerDataBlock>
            <tns:ManufacturerName>медквадрат</tns:ManufacturerName>
            <tns:ManufacturerCountry>АЗЕРБАЙДЖАН</tns:ManufacturerCount
            <tns:ManufacturerCountryCode>31</tns:ManufacturerCountryCode>
            <tns:ManufacturerAddress>тестовый адрес</tns:ManufacturerAddre
        </tns:ManufacturerDataBlock>
```

Рисунок 42 — Пример набора атрибутов в XSD-схеме услуги

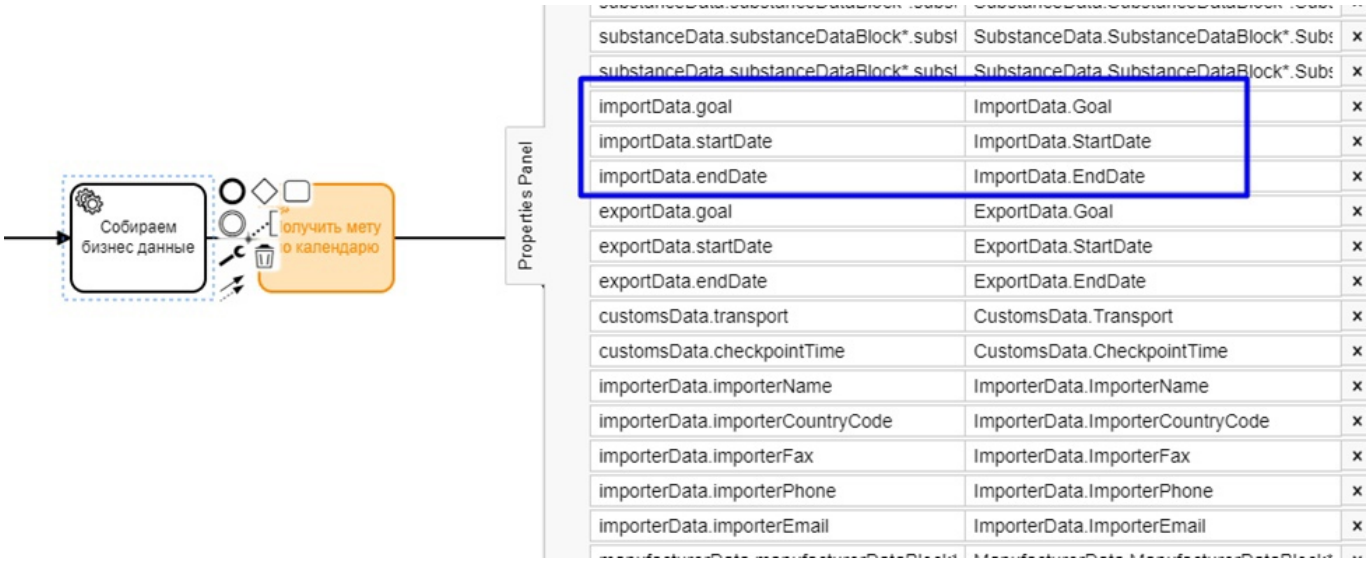


Рисунок 43 — Набор атрибутов в BPMN-схеме процесса импорта, соответствующий примеру на рисунке 42

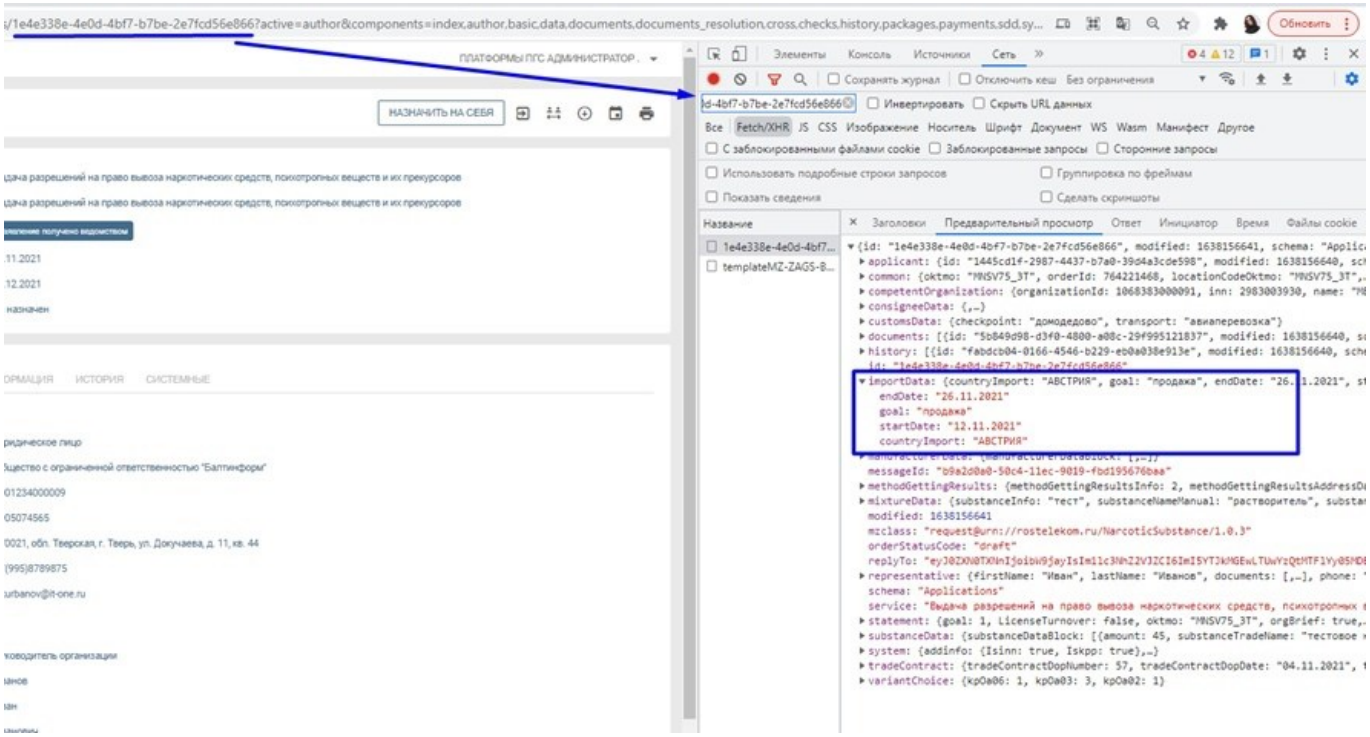


Рисунок 44 — Набор атрибутов в коде заявления в ПГС, соответствующий примеру на рисунке 42

Если услуга имеет только одну цель (в качестве единственной подуслуги выступает сама услуга, и в XSD-схеме услуги отсутствуют теги, в которых фигурирует значение *Goal*), то для переменной statement.goal необходимо вручную указать соответствие *${1}* (рисунок 41).

Переменная resource должна иметь значение *${petition}* (рисунок 45).

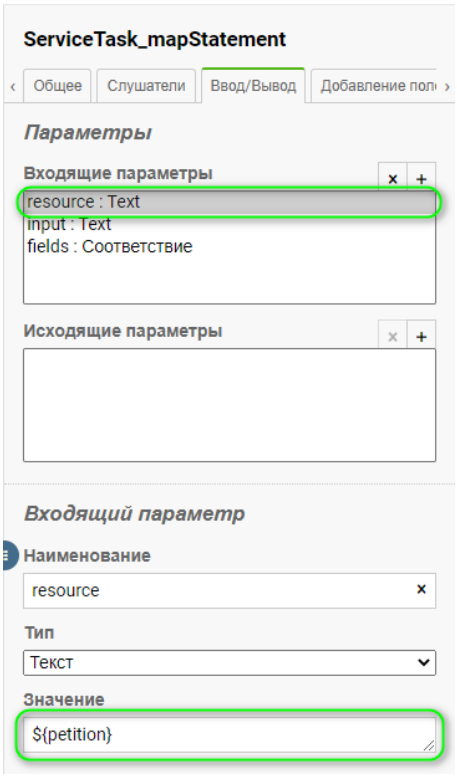


Рисунок 45 — Переменная resource и ее значение на вкладке «Ввод/Вывод» панели параметров элемента процесса импорта «Собираем бизнес-данные»

Прочие настройки процесса импорта выполняются на усмотрение лица, настраивающего процесс предоставления услуги.

В некоторых случаях процесс может иметь более одного элемента «Собираем бизнес-данные». При этом таблица соотвествия содержится в последнем элементе «Собираем бизнес-данные», а предыдущие содержат предварительную обработку данных и (или) данные о доходах заявителя. Возможны также иные конфигурации.

Для сохранения любых изменений, внесенных в процесс во встроенном конструкторе, необходимо нажимать кнопку «Сохранить» (рисунок 39).

Любые изменения процесса можно также выполнять в конструкторе процессов Camunda Modeler. Для этого необходимо сохранить процесс импорта на АРМ, используя кнопку ↓ («Скачать») на вкладке карточки услуги «Процессы» (рисунок 16), и по завершении внесения изменений заново загрузить процесс (в виде файла) в услугу на той же вкладке, используя кнопку 📎 («Загрузить файл процесса»).

Кроме того, для успешного получения XML-файлов заявлений в ПГС необходимо:

1. Сохранить процесс импорта на АРМ.
2. Внести изменения в идентификатор процесса (в коде процесса и в названии файла).
3. Для изменения идентификатора в коде процесса необходимо в Camunda Modeler нажать мышью на любое пустое место в области, в которой расположена BPMN-схема процесса, на панели справа перейти на вкладку General («Общее») и в поле ID изменить значение на *Import<код услуги>PetitionJson<номер версии услуги>* (рисунок 46). Часть <номер версии услуги> может отсутствовать, однако идентификатор процесса должен быть уникальным в ПГС.

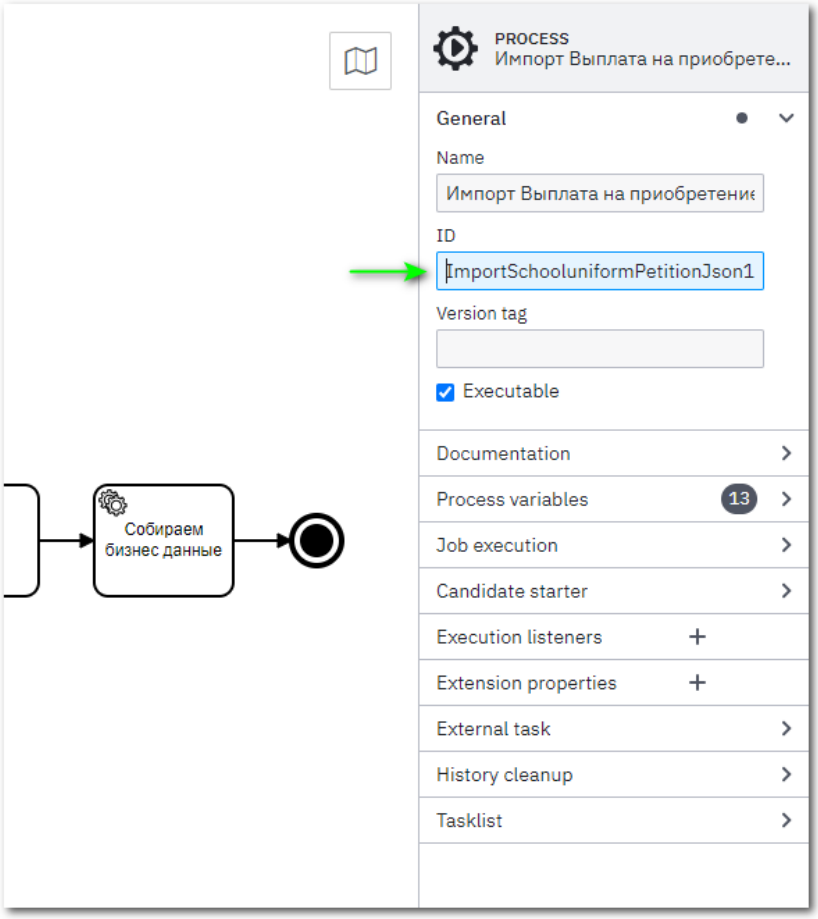


Рисунок 46 — Изменение идентификатора процесса в Camunda Modeler

1. Заново загрузить процесс в ПГС.

2. Загрузить процесс в реестр процессов. Для этого открыть его во встроенном BPMN-конструкторе и нажать кнопку «Сохранить».

3.6.2. Базовый импорт

Для большинства услуг предпочтительно использование процесса базового импорта вместо стандартного, за исключением услуг, для которых требуется сложная маршрутизация (например, когда заявления о предоставлении услуги должны поступать в одно конкретное ведомство, но данное ведомство не указано в XML-файле заявления).

Процесс базового импорта представляет собой процесс импорта, в котором присутствуют элементы начала и конца процесса и элемент «Собираем бизнес-данные» (один или более), отвечающий за соответствие атрибутов, и заданы определенные настройки для этого элемента.

Для перехода к базовому импорту необходимо открыть процесс для редактирования в Camunda Modeler или встроенном конструкторе и удалить все элементы, кроме начала, конца процесса и «Собираем бизнес-данные» (рисунок 47).

Рисунок 47 — BPMN-схема процесса базового импорта

Затем следует у элемента «Собираем бизнес данные» удалить атрибуты (в переменной fields на панели параметров элемента), которые являются общими для всех заявлений и не несут никакой информации для настраиваемой услуги (если данная операция не была осуществлена при получении процесса импорта из XSD-схемы услуги — см. подраздел 3.5).

Далее следует все атрибуты в левой колонке поместить в переменную statement, дописав значение *statement.* перед каждым атрибутом (рисунок 48).

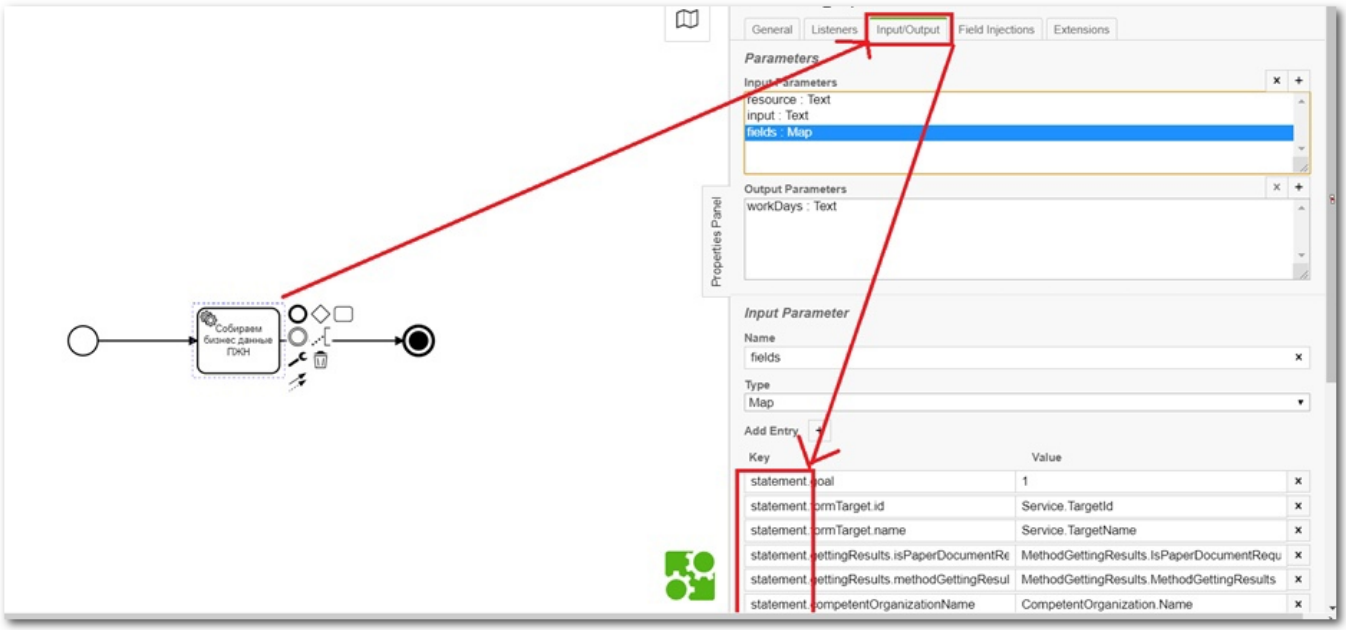


Рисунок 48 — Атрибуты заявления в ПГС в переменной statement

Переменная resource должна иметь значение *${petition}* (рисунок 45).

На вкладке панели параметров элемента «Расширения» должна присутствовать (при отсутствии — следует добавить) характеристика *command=starMapping* (рисунок 49).

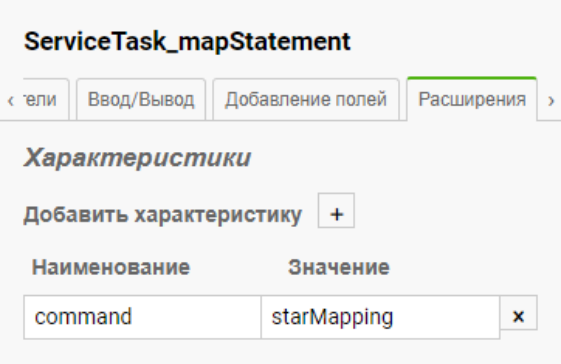


Рисунок 49 — Характеристика *command=starMapping* на вкладке «Расширения» панели параметров элемента процесса импорта «Собираем бизнес-данные»

В таблицу соответствия также необходимо добавить максимальный срок предоставления услуги (рисунок 50):

– в календарных днях — атрибут statement.dueDays;

– в рабочих днях — атрибут statement.workDays.

Рисунок 50 — Задние максимального срока предоставления услуги

Если цели (подуслуги) услуги имеют разные максимальные сроки предоставления, то для их задания требуется использовать скрипт. Пример скрипта:

```
${petition.Goal == 4 || petition.Goal == 5 ? 9 : 19}
${petition.Goal == 1 ? 24 : 9}
```

Для сохранения изменений, внесенных в процесс во встроенном конструкторе, необходимо нажимать кнопку «Сохранить» (рисунок 39).

Если редактирование процесса осуществлялось в Camunda Modeler, то необходимо:

1. Конечному процессу присвоить идентификатор такой же, как у исходного процесса импорта (см. п. 3.6.1, рисунок 46).

2. Сохранить процесс в виде XML-файла с именем таким же, как у исходного процесса (с расширением .bpmn).

3. Загрузить файл процесса в услугу взамен исходного процесса, используя кнопку ( ) («Загрузить файл процесса») на вкладке карточки услуги «Процессы».

4. Загрузить файл процесса в ПГС в разделе «Черновики процессов», используя кнопку «Загрузить схему» (рисунок 51).

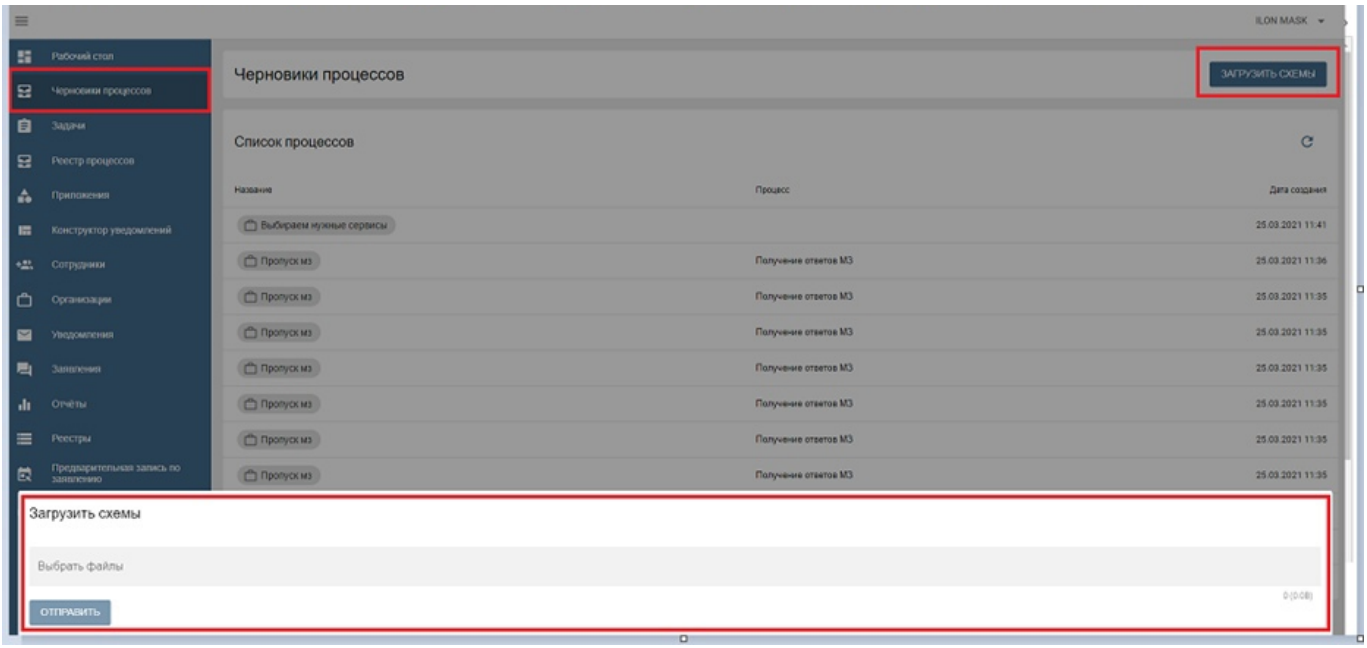


Рисунок 51 — Загрузка процесса в ПГС в разделе «Черновики процессов»

После загрузки процесса необходимо перевести переключатель «Базовый импорт» вверху карточки услуги в положение «Вкл».

Примечание. Переключатель «Базовый импорт» в положении «Вкл» определяет, что должен использоваться базовый импорт вместо настраиваемого, но не заменяет настраиваемый процесс на процесс базового импорта. Таким образом, для использования базового импорта настраиваемый процесс импорта должен быть заменен на процесс базового импорта согласно вышеописанному порядку. Аналогично при переключении с базового импорта на настраиваемый в качестве процесса импорта сохраняется исходный процесс базового импорта. Для того чтобы был возможен переход от процесса базового импорта обратно к использованию настраиваемого процесса импорта, необходимо *перед его изменением настраиваемого процесса импорта сохранить его в виде файла* и повторно загрузить в услугу, используя кнопку ( ) («Загрузить файл процесса») на вкладке карточки услуги «Процессы».

3.7. Добавление и редактирование справочников. Добавление оснований для отказа в приеме документов и оснований для отказа в предоставлении услуги

На вкладке карточки услуги «Справочники» присутствуют справочники, которые были добавлены при выполнении операции получения процесса импорта из XSD-схемы услуги (рисунок 52).

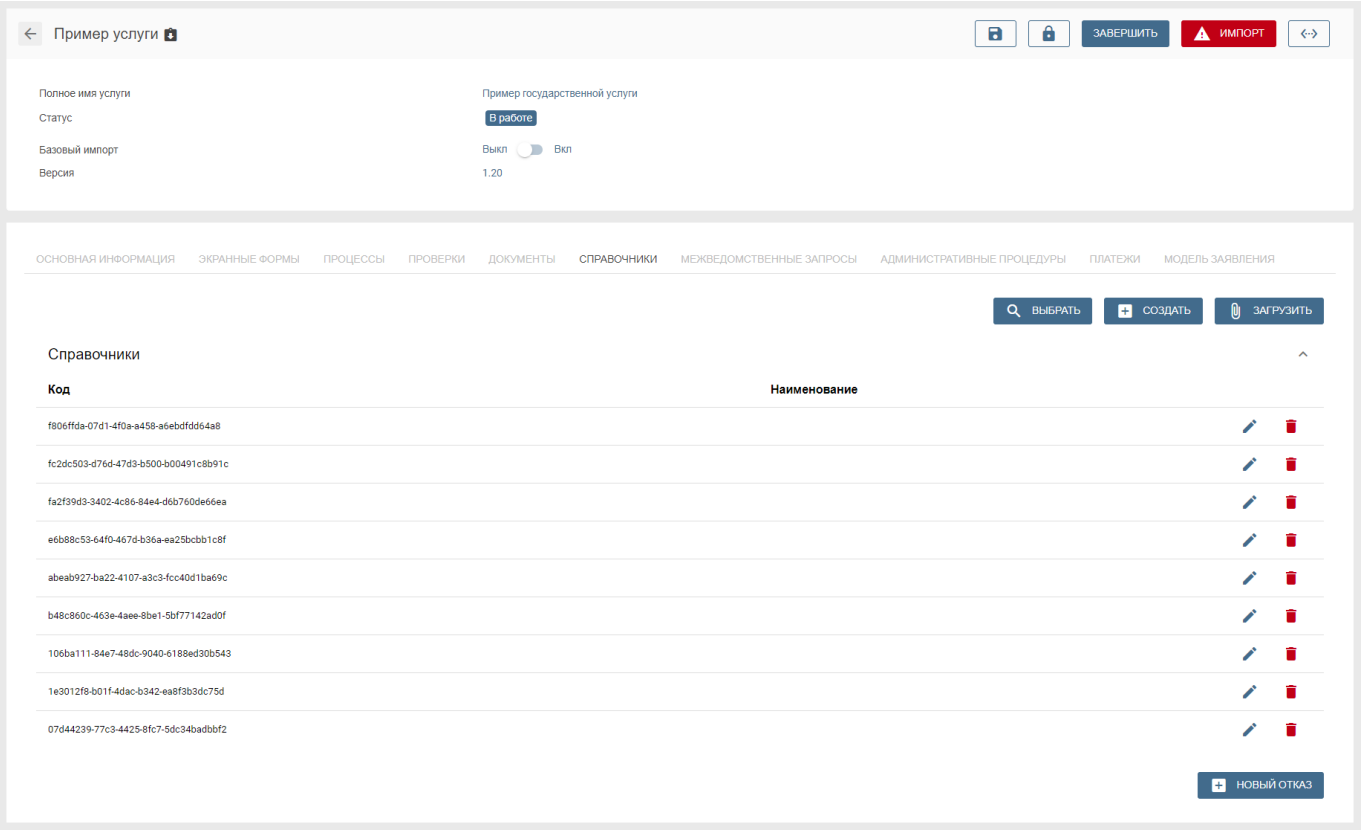


Рисунок 52 — Справочники на вкладке карточки услуги «Справочники»

Необходимо проверить, анализируя справочники на вкладке и XSD-схему услуги, все ли справочники из XSD-схемы были корректно импортированы в услугу. При необходимости следует внести изменения в существующие справочники или (и) добавить новые.

В XSD-схеме услуги признаками справочников являются теги, в которых фигурирует значение *xsd:byte* (рисунок 53).

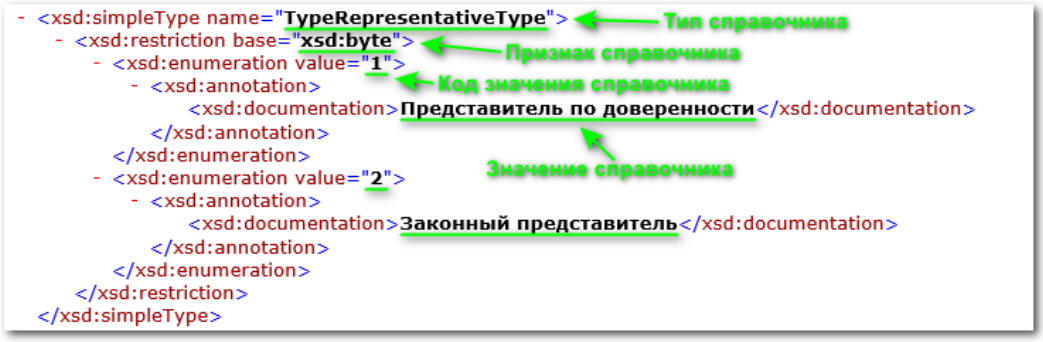


Рисунок 53 — Данные справочника в XSD-схеме услуги:
тег <xsd:enumeration> содержит код значения справочника;
тег <xsd:documentation> содержит значение справочника

В качестве новых справочников в услугу могут быть добавлены как справочники, хранящиеся в каталоге НСИ ПГС, так и справочники, создаваемые вручную лицом, настраивающим процесс предоставления услуги. Для добавления справочника в первом случае следует нажать кнопку «Выбрать» и в открывшемся окне найти и выбрать требуемые справочники (рисунок 54). Во втором случае следует нажать кнопку «Создать» и в открывшемся окне внести код и имя справочника, а также необходимые записи справочника в формате «код-значение», затем нажать кнопку «Создать» (рисунок 55). Дополнительные поля не заполняются.

Значения кода и имени справочника должны совпадать и присваиваться в виде *<код услуги>_<тип справочника>* или *<код услуги>_<номер версии услуги>_<тип справочника>*, где <тип справочника> — тип справочника, указанный в XSD-схеме услуги в качестве значения атрибута name тега <xsd:simpleType> (рисунок 53). Значение не должно содержать пробелов и точек.

Для добавления записи в справочник следует нажать кнопку «Добавить запись», ввести код и значение записи в появившиеся поля и нажать кнопку «Добавить» (рисунок 56). Код записи должен являться целым числом, уникальным в пределах данного справочника; значение может быть любым. При необходимости повторить данные действия для добавления других записей.

Для просмотра и редактирования данных справочника следует использовать соответствующую ему кнопку («Редактировать»), для удаления справочника — кнопку («Удалить»).

Редактирование данных справочника возможно в двух режимах:

– в режиме формы;

– в режиме редактора JSON.

Редактирование данных справочника в режиме формы осуществляется аналогично внесению данных при создании нового справочника, и окно редактирования данных в этом режиме (рисунок 57) имеет вид, аналогичный окну создания справочника (рисунок 55).

Окно редактирования данных в режиме редактора JSON имеет вид, представленный на рисунке 58.

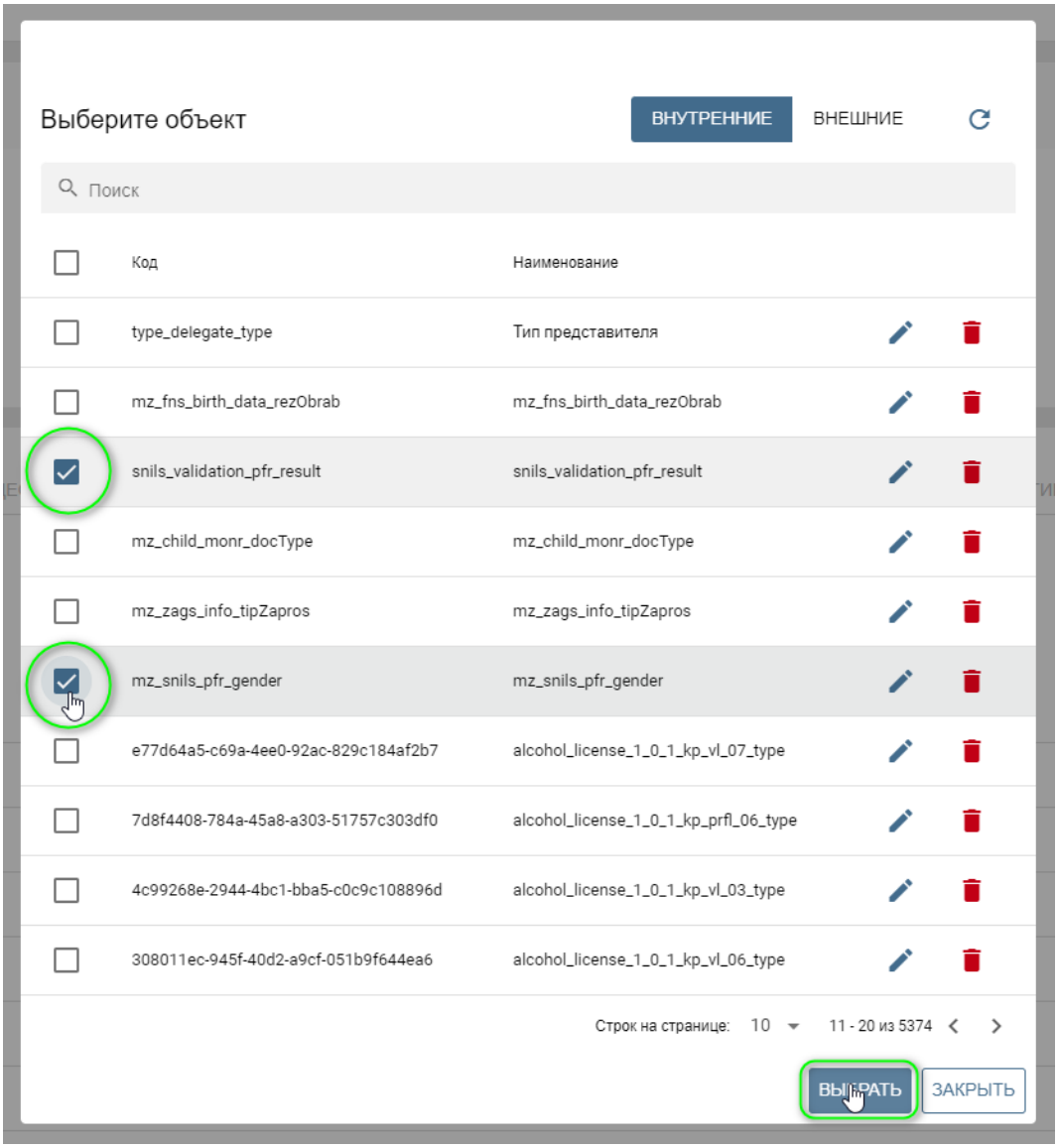


Рисунок 54 — Окно выбора справочников в каталоге НСИ ПГС

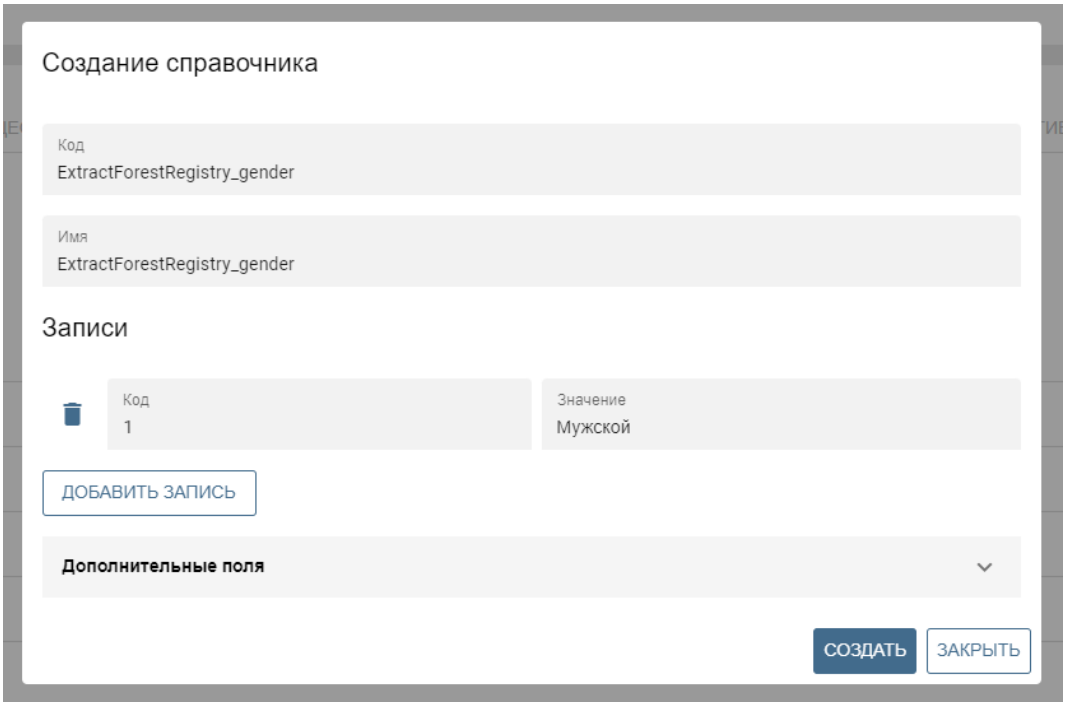


Рисунок 55 — Окно создания справочника

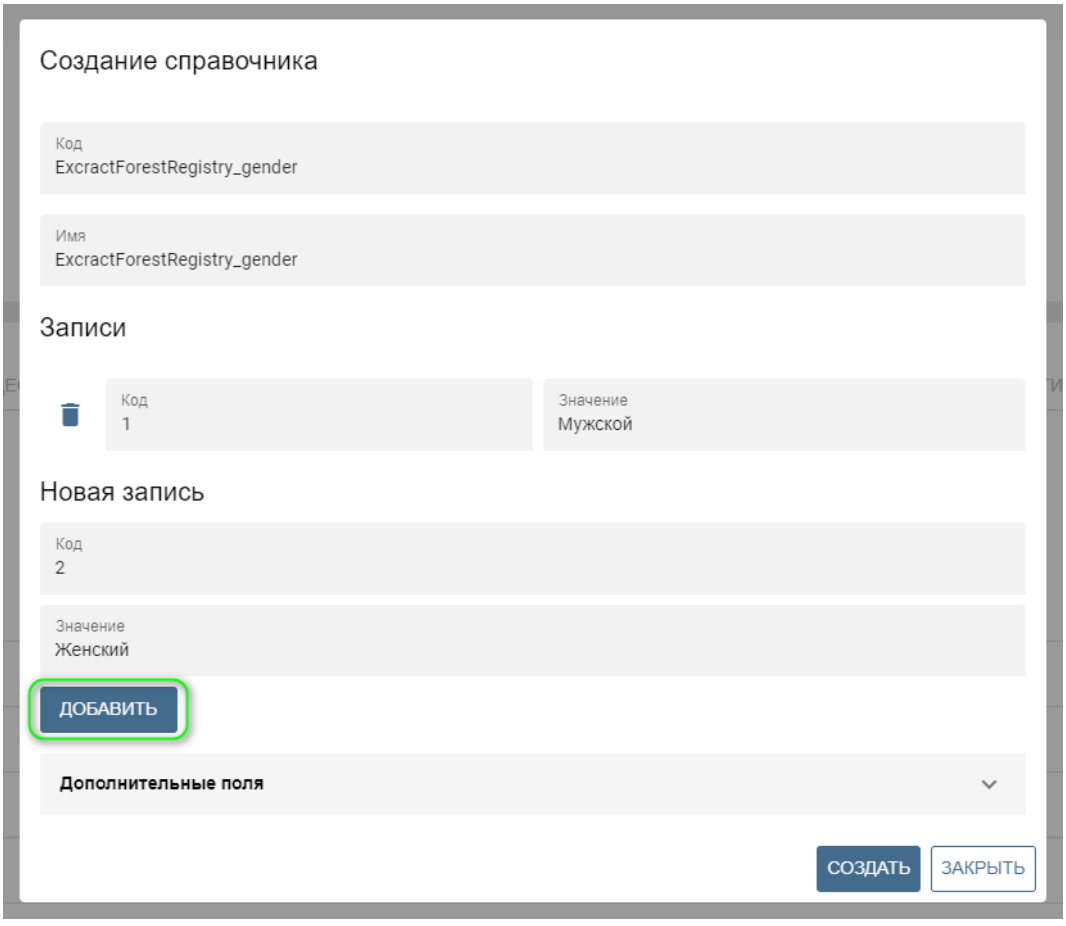


Рисунок 56 — Ввод данных и кнопка добавления записи справочника

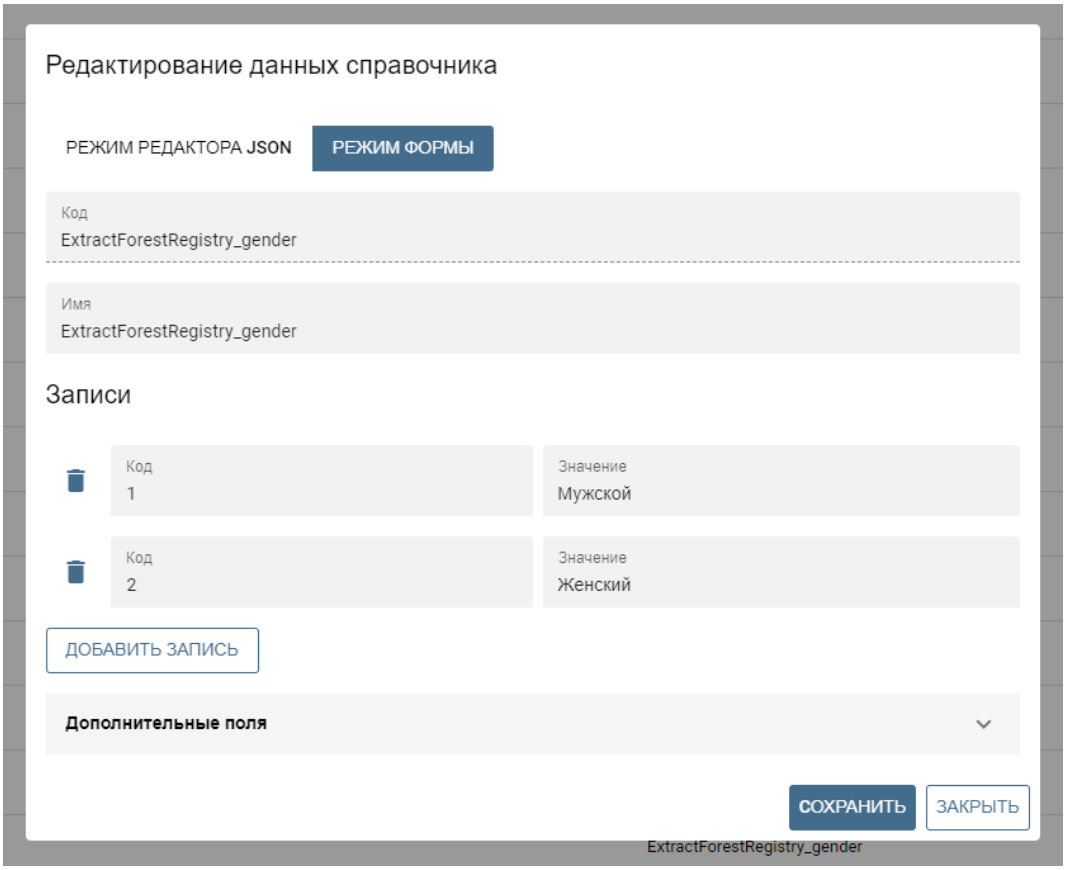


Рисунок 57 — Окно редактирования данных справочника в режиме формы

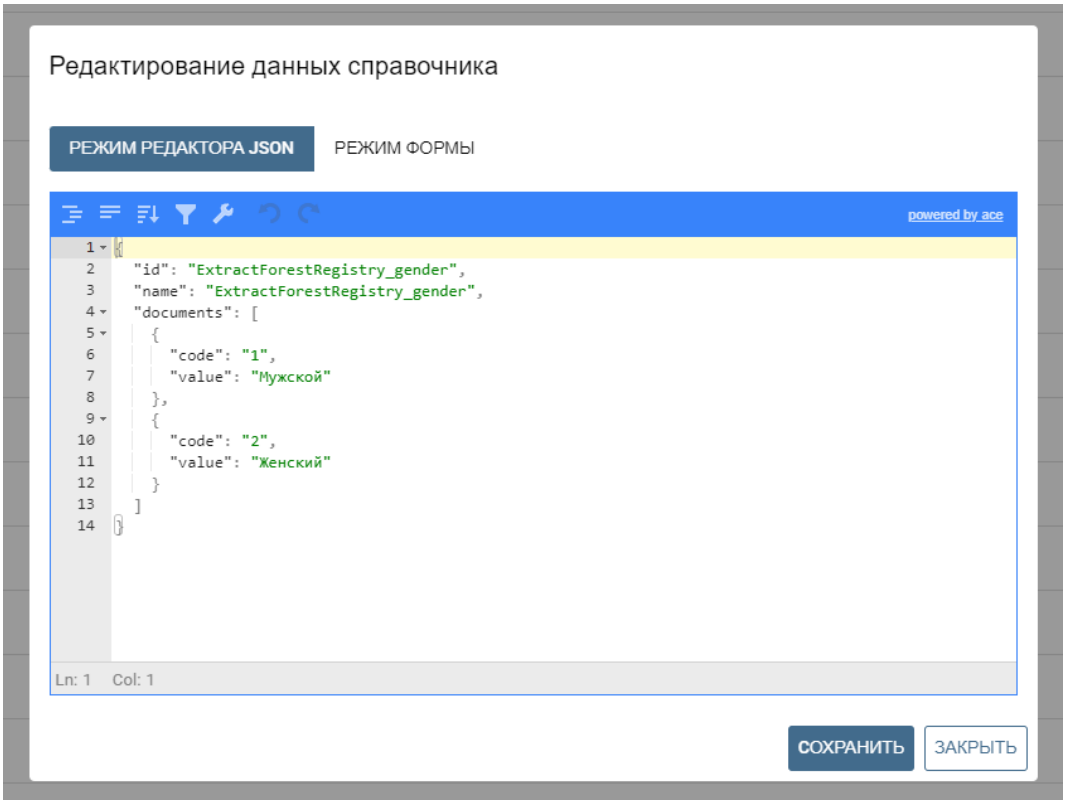


Рисунок 58 — Окно редактирования данных справочника в режиме редактора JSON

Помимо справочников, на вкладке «Справочники» должен быть сформирован исчерпывающий перечень оснований для отказа в приеме документов заявителя и в предоставлении

услуги (см. пример на рисунке 59). Как правило, такой перечень содержится в документации на услугу.

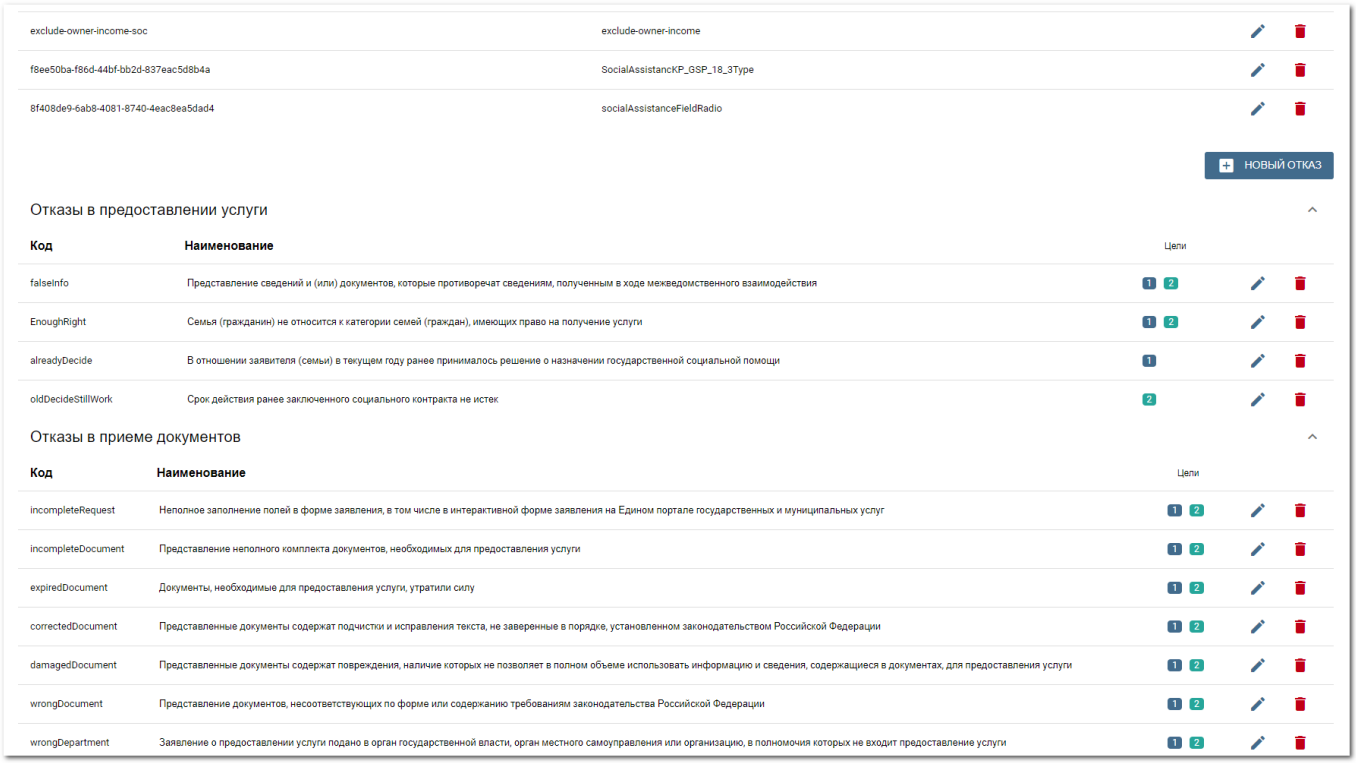


Рисунок 59 — Основания для отказа на вкладке карточки услуги «Справочники» для услуги «Оказание (предоставление) государственной социальной помощи отдельным категориям граждан на территории субъекта Российской Федерации в соответствии с законодательством субъекта Российской Федерации»

Для добавления основания для отказа, необходимо:

1. Нажать кнопку «Новый отказ» (рисунок 60).

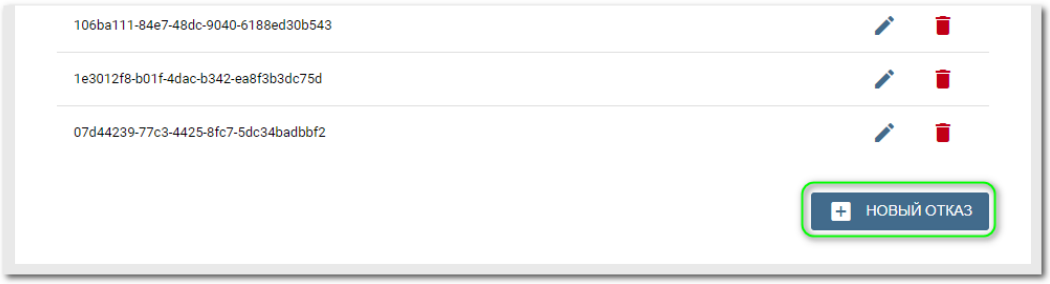


Рисунок 60 — Вкладка карточки услуги «Справочники». Кнопка добавления основания для отказа

2. В открывшемся окне (рисунок 61):

– указать:

а) код отказа — любое значение, уникальное для данной услуги;

б) имя отказа — формулировка основания для отказа согласно документации на услугу;

в) тип отказа — отказ в приеме документов или отказ в предоставлении услуги (выбирается в выпадающем списке — рисунок 62);

г) цели — подуслуги, которым соответствует данное основание для отказа (выбираются в выпадающем списке — рисунок 63; могут быть выбраны одна или более целей — рисунок 64);

– нажать кнопку «Создать».

После этого добавленное основание для отказа появится под списком справочников (рисунок 65).

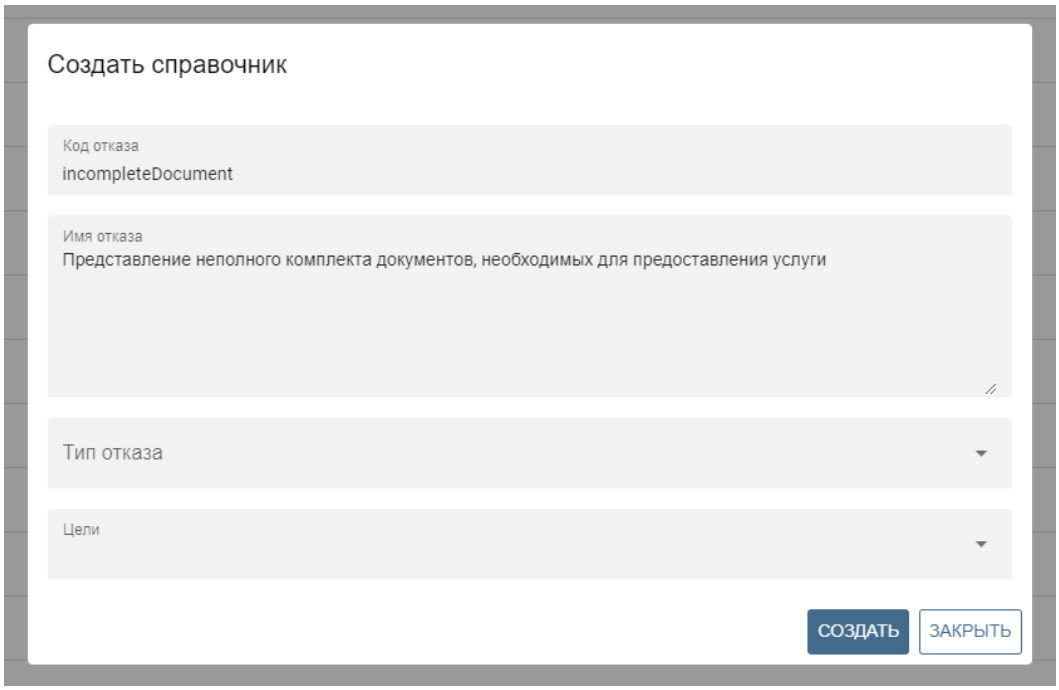


Рисунок 61 — Окно добавления основания для отказа

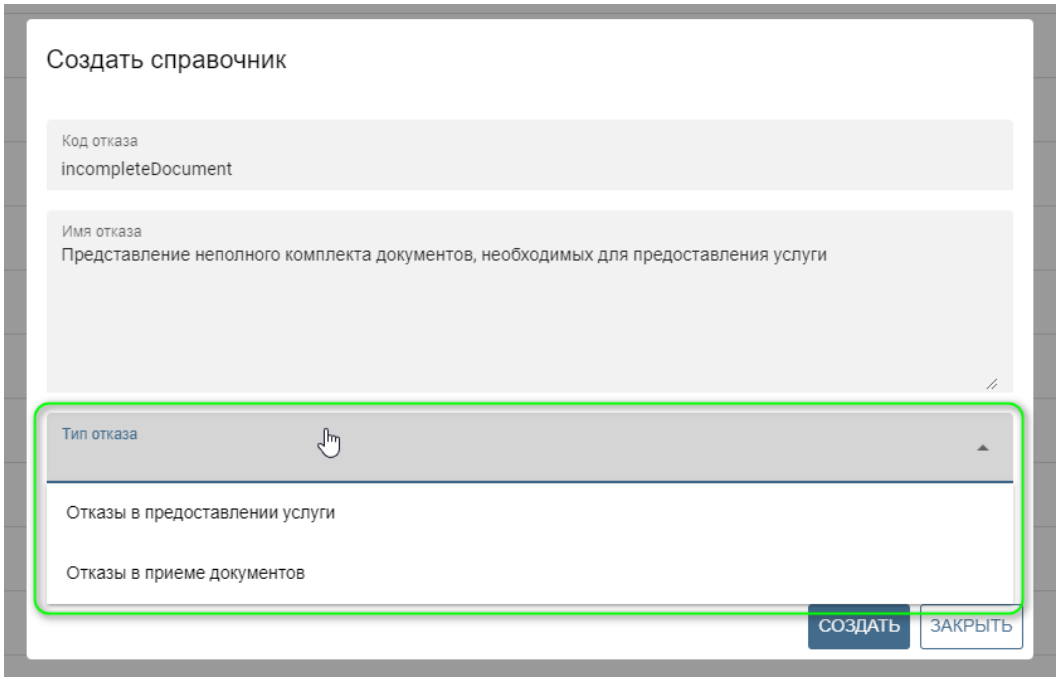


Рисунок 62 — Окно добавления основания для отказа. Выпадающий список типов отказа

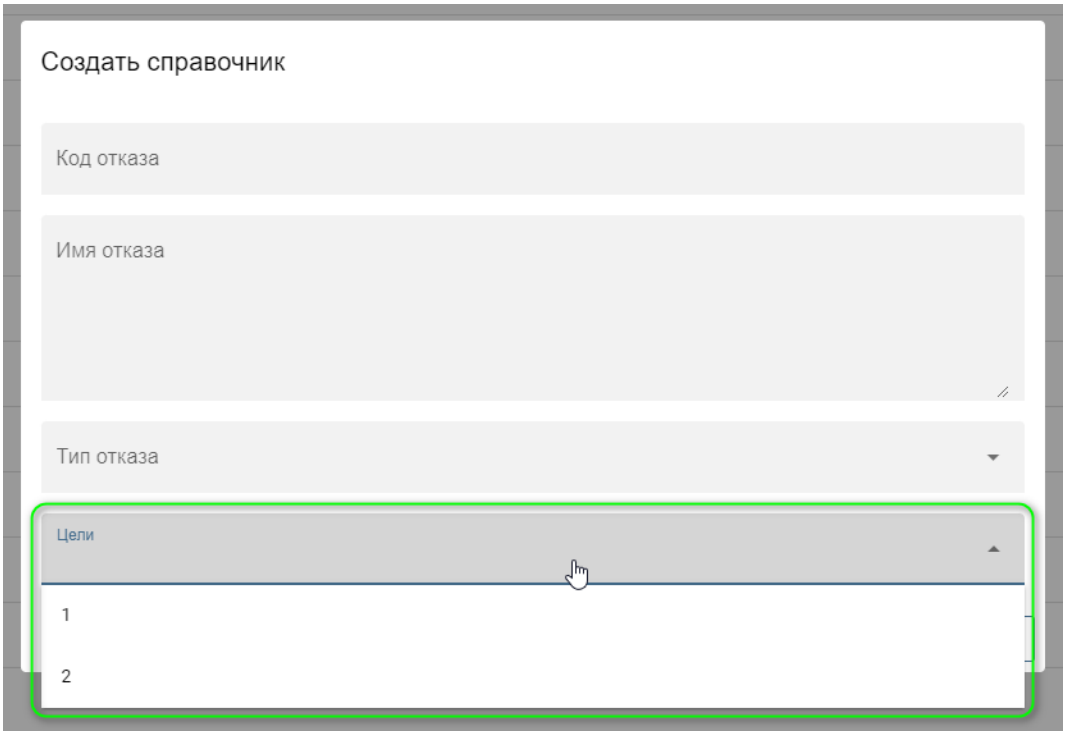


Рисунок 63 — Окно добавления основания для отказа. Выпадающий список целей (подуслуг)

Рисунок 64 — Окно добавления основания для отказа. Выбранные цели (более одной)

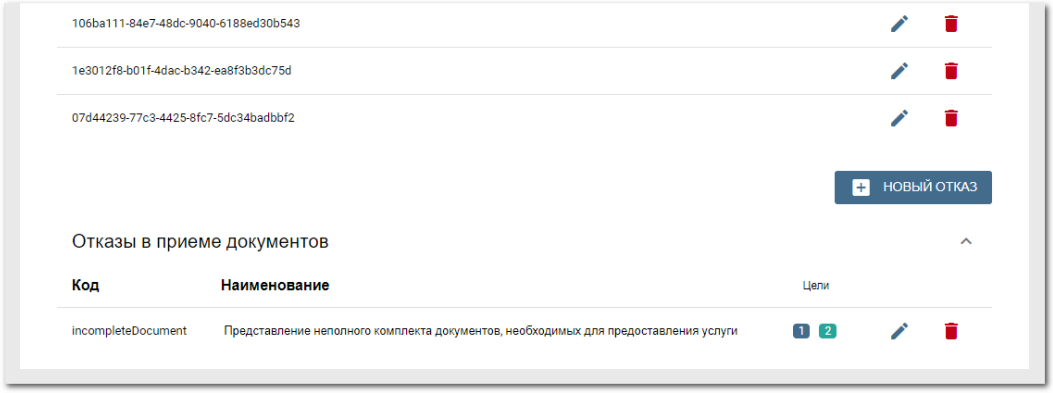


Рисунок 65 — Добавленное основание для отказа на вкладке карточки услуги «Справочники»

3.8. Входные и выходные документы. Добавление видов входных документов для предоставления услуги

На вкладке карточки услуги «Документы» необходимо задать виды документов, которые будут поступать в ведомство, предоставляющее услугу, и создаваться ведомством в процессе предоставления услуги.

Виды документов подразделяются на пять категорий по источнику происхождения документа:

– входные документы:

а) входящие документы — документы, предоставляемые заявителем вместе с заявлением о предоставлении услуги;

б) документы, добавляемые специалистом, — документы, которые в ходе процесса предоставления услуги заявителю создает и (или) получает и прилагает к заявлению в ПГС специалист, исполняющий данный процесс (например, документы, получаемые при «ручном» межведомственном взаимодействии не через СМЭВ, промежуточные результаты предоставления услуги и т. п.);

в) документы, отправляемые заявителю, — документы, добавляемые специалистом, предоставляемые заявителю в ходе процесса предоставления услуги как промежуточные;

– выходные документы:

а) шаблон печати заявления — настраиваемые шаблоны документов для печати;

б) результирующие документы — результирующие документы по заявлению о предоставлении услуги, предоставляемые заявителю.

Пример добавленных видов документов для услуги приведен на рисунке 66.

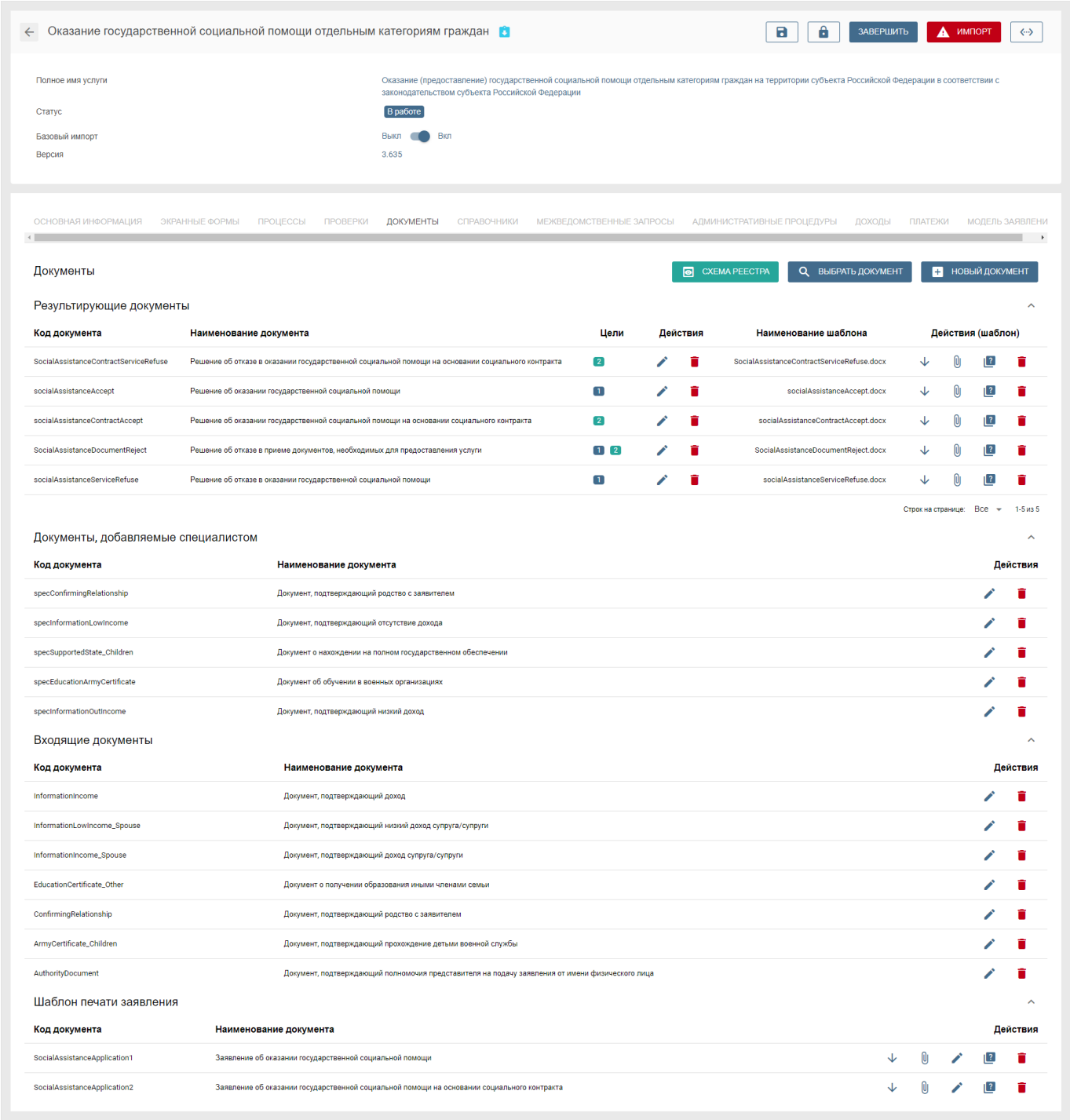


Рисунок 66 — Виды входных и выходных документов на вкладке карточки услуги «Документы» для услуги «Оказание (предоставление) государственной социальной помощи отдельным категориям граждан на территории субъекта Российской Федерации в соответствии с законодательством субъекта Российской Федерации»

Для видов документов доступны следующие операции:

– для видов входных документов:

а) редактирования атрибутов вида документа — кнопка ( ) («Редактировать»);

б) удаления вида документа — кнопка ( ) («Удалить»);

– для результирующих документов:

а) редактирования атрибутов вида документа — кнопка ( ) («Редактировать»);

б) удаления вида документа — кнопка ( ) («Удалить»);

в) загрузки настроенного шаблона документа в формате .docx или .doc в ПГС — кнопка ( ) («Загрузить шаблон»);

г) скачивания шаблона документа на компьютер — кнопка ( ) («Скачать шаблон»);

д) редактирования шаблона документа — кнопка ( ) («Редактировать шаблон»);

е) проверки (тестирования) шаблона документа — кнопка ( ) («Тестировать шаблон»);

ж) удаления шаблона документа — кнопка ( ) («Удалить шаблон»);

– для шаблона печати заявления:

а) загрузки настроенного шаблона документа в формате .docx или .doc в ПГС — кнопка ( ) («Загрузить шаблон»);

б) скачивания шаблона документа на компьютер — кнопка ( ) («Скачать шаблон»);

в) редактирования шаблона документа — кнопка ( ) («Редактировать шаблон»);

г) проверки (тестирования) шаблона документа — кнопка ( ) («Тестировать шаблон»);

д) удаления шаблона документа — кнопка ( ) («Удалить шаблон»).

Виды документов в услуге задаются в несколько этапов. Первыми добавляются виды входных документов, в том числе и автоматически при импорте из ЕПГУ (см. подраздел 3.5). Шаблоны выходных документов добавляются позднее (см. подраздел 3.15).

В услугу могут быть добавлены виды входных документов как хранящиеся в каталоге НСИ ПГС, так и пользовательские. Для добавления вида документа в первом случае следует нажать кнопку «Выбрать документ» и в открывшемся окне найти и выбрать требуемые виды документов (рисунок 67). Во втором случае следует нажать кнопку «Новый документ» и в открывшемся окне (рисунок 68) ввести код и наименование вида документа, выбрать источник происхождения документа в выпадающем списке, затем нажать кнопку «Создать». Прочие поля не заполняются.

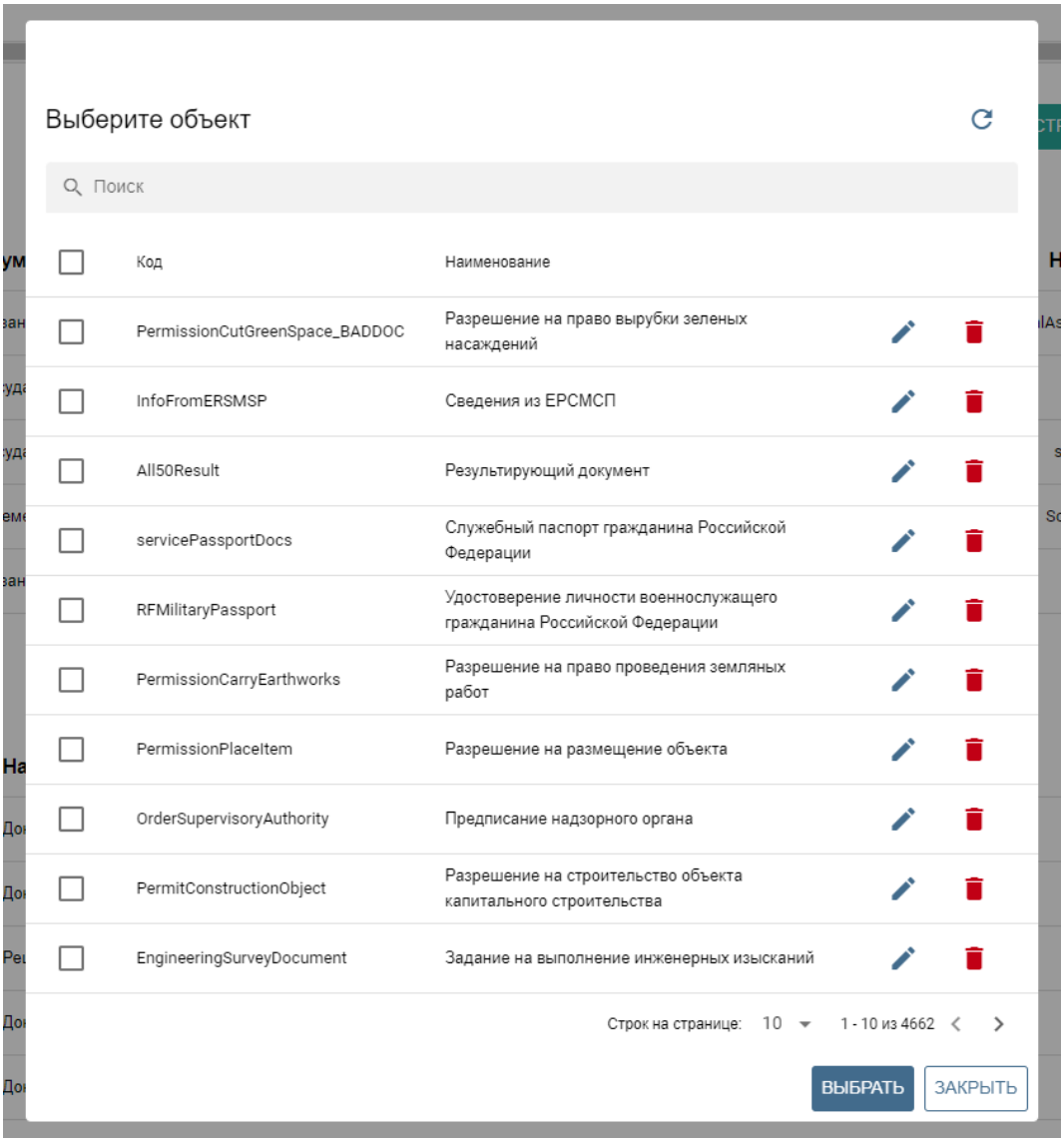


Рисунок 67 — Окно выбора видов документов в каталоге НСИ ПГС

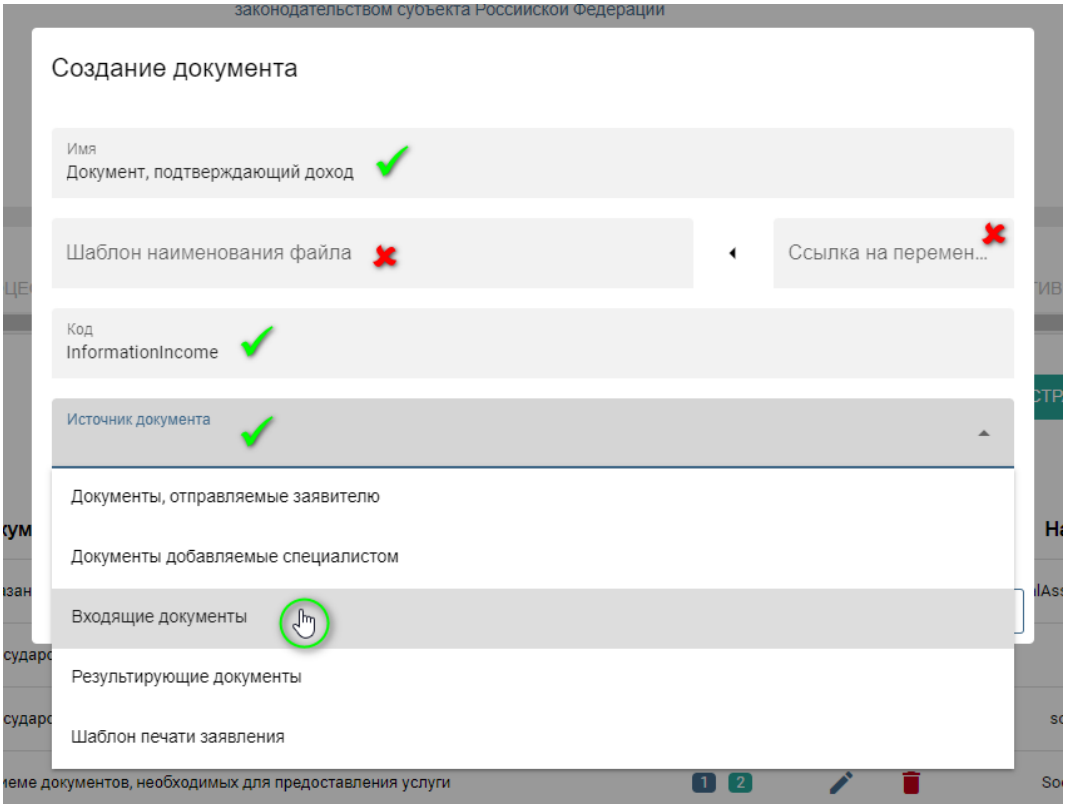


Рисунок 68 — Окно добавления вида документа в услугу

Коды и наименования видов входящих документов можно найти в XSD-схеме услуги. Коды и наименования прочих видов документов либо содержатся в документации на услугу, либо присваиваются лицом, настраивающим процесс предоставления услуги. Код представляет собой значение, состоящее из латинских букв, цифр и символов. Наименование может быть любым.

Как и редактирование данных справочников, редактирование атрибутов документов возможно в режиме формы и режиме редактора JSON:

– в режиме формы;

– в режиме редактора JSON.

3.9. Добавление административных процедур услуги

На вкладке карточки услуги «Административные процедуры» необходимо задать административные процедуры, предусмотренные для услуги. Как правило, перечень административных процедур содержится в документации на услугу.

Для добавления административной процедуры следует нажать кнопку «Новый этап» (рисунок 69) и в открывшемся окне (рисунок 70) указать:

– номер административной процедуры по порядку;

– наименование административной процедуры (выбирается в выпадающем списке или указывается вручную — рисунки 71 и 72 соответственно);

– статусы, соответствующие административной процедуре (выбираются в выпадающем списке — рисунок 73; могут быть выбраны один или более статусов — рисунок 74);

– триггер — условие перехода к данной административной процедуре от предыдущей (выбирается в выпадающем списке — рисунок 75).

Затем следует нажать кнопку «Создать». Поля «Период предоставления (рабочие дни)», «Время уведомления», «Единицы измерения» не заполняются. Результирующие документы задаются позднее (см. подраздел 3.17).

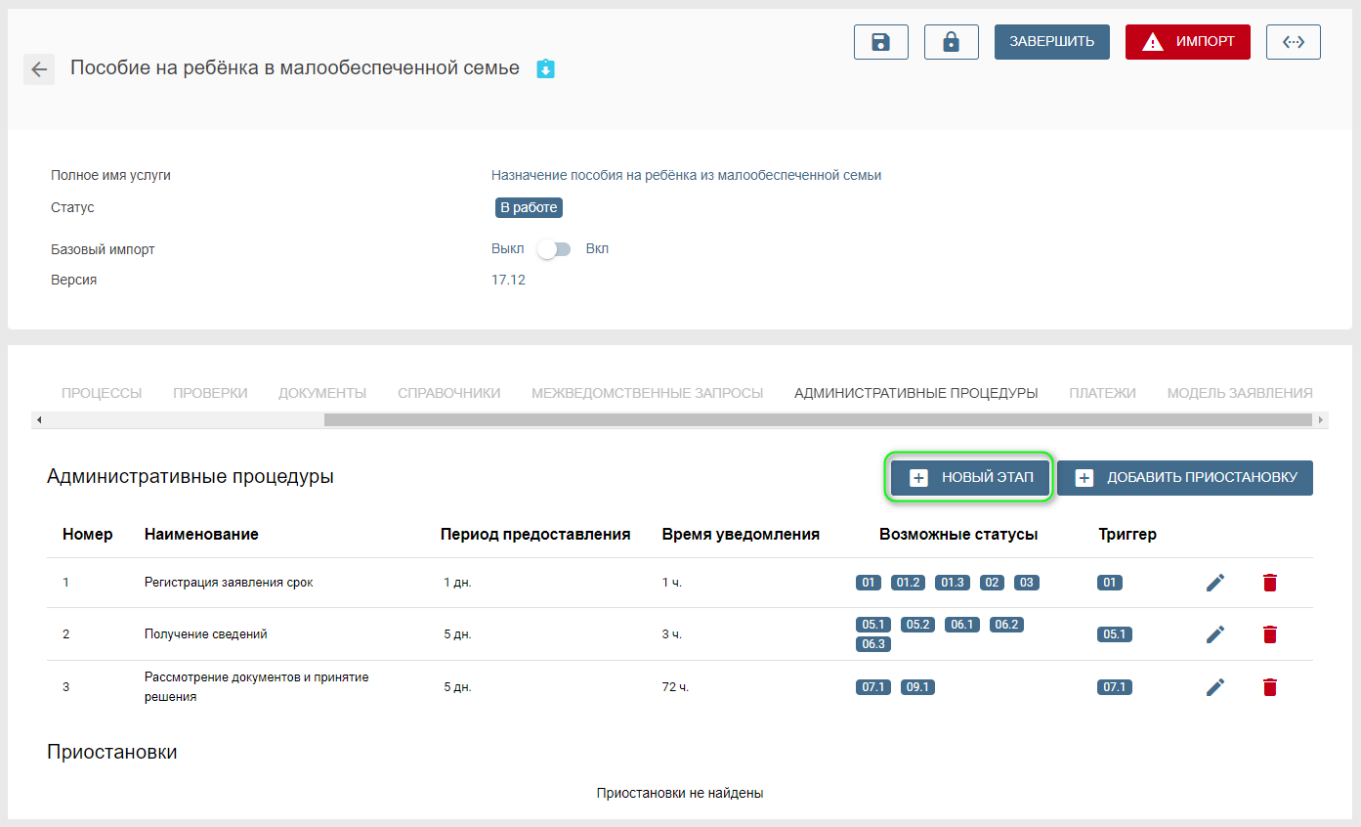


Рисунок 69 — Административные процедуры услуги
на вкладке карточки услуги «Административные процедуры». Кнопка «Новый этап»

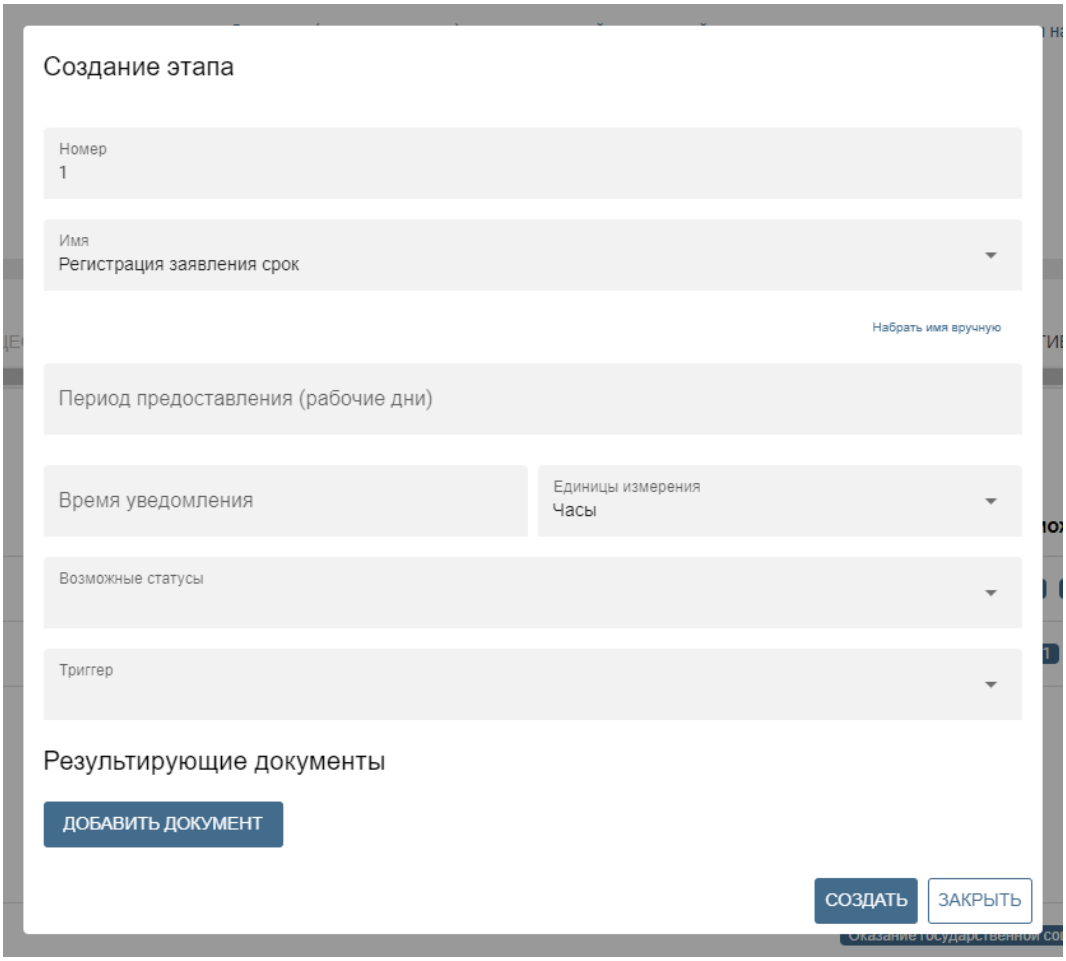


Рисунок 70 — Окно добавления административной процедуры в услугу

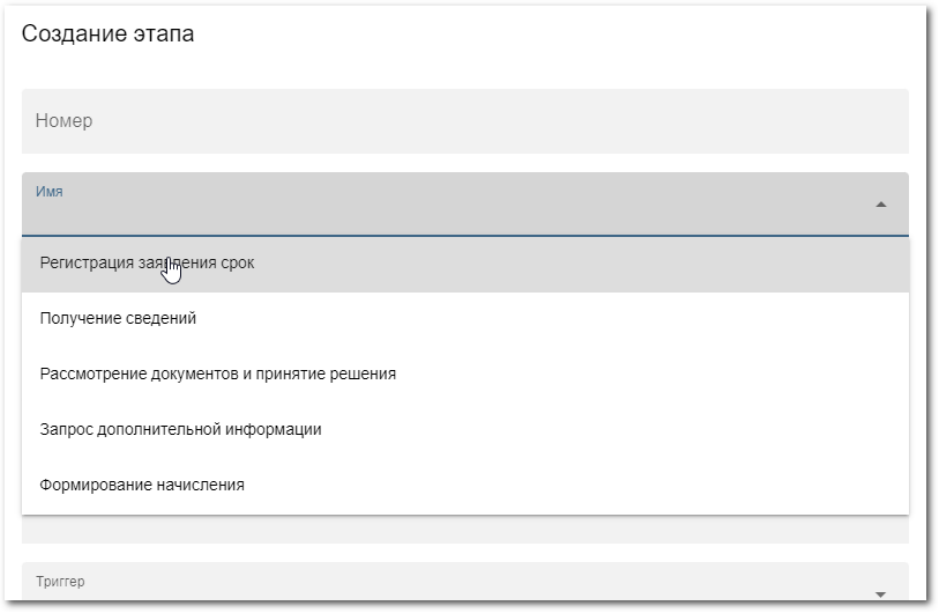


Рисунок 71 — Окно добавления административной процедуры в услугу. Выбор административной процедуры в выпадающем списке

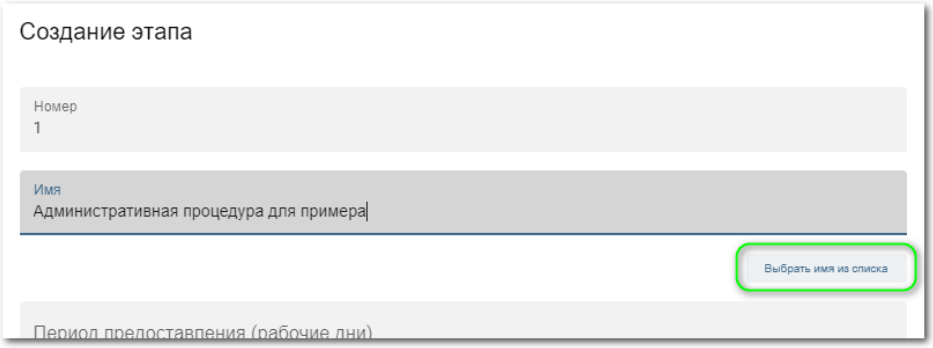


Рисунок 72 — Окно добавления административной процедуры в услугу. Ввод наименования административной процедуры вручную

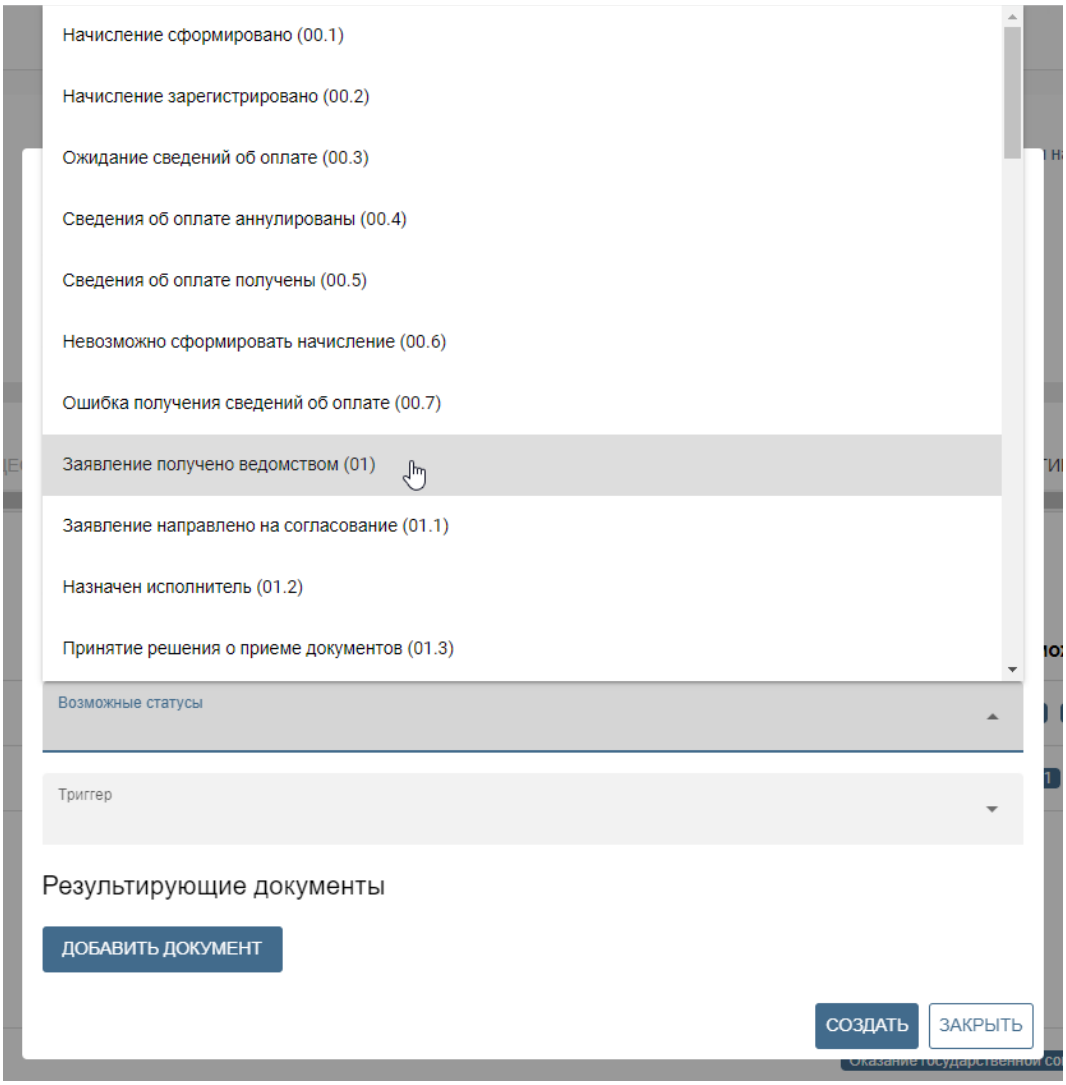


Рисунок 73 — Окно добавления административной процедуры в услугу. Выбор статусов для административной процедуры

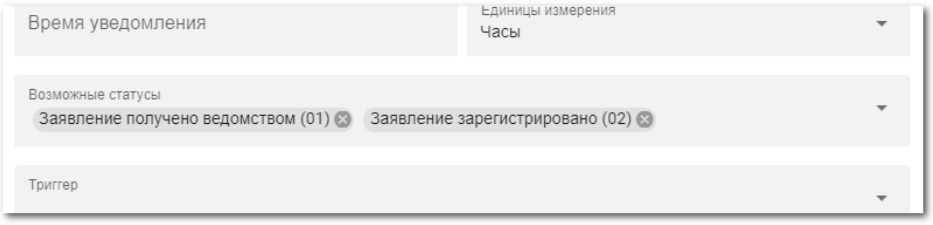


Рисунок 74 — Окно добавления административной процедуры в услугу. Выбранные статусы для административной процедуры

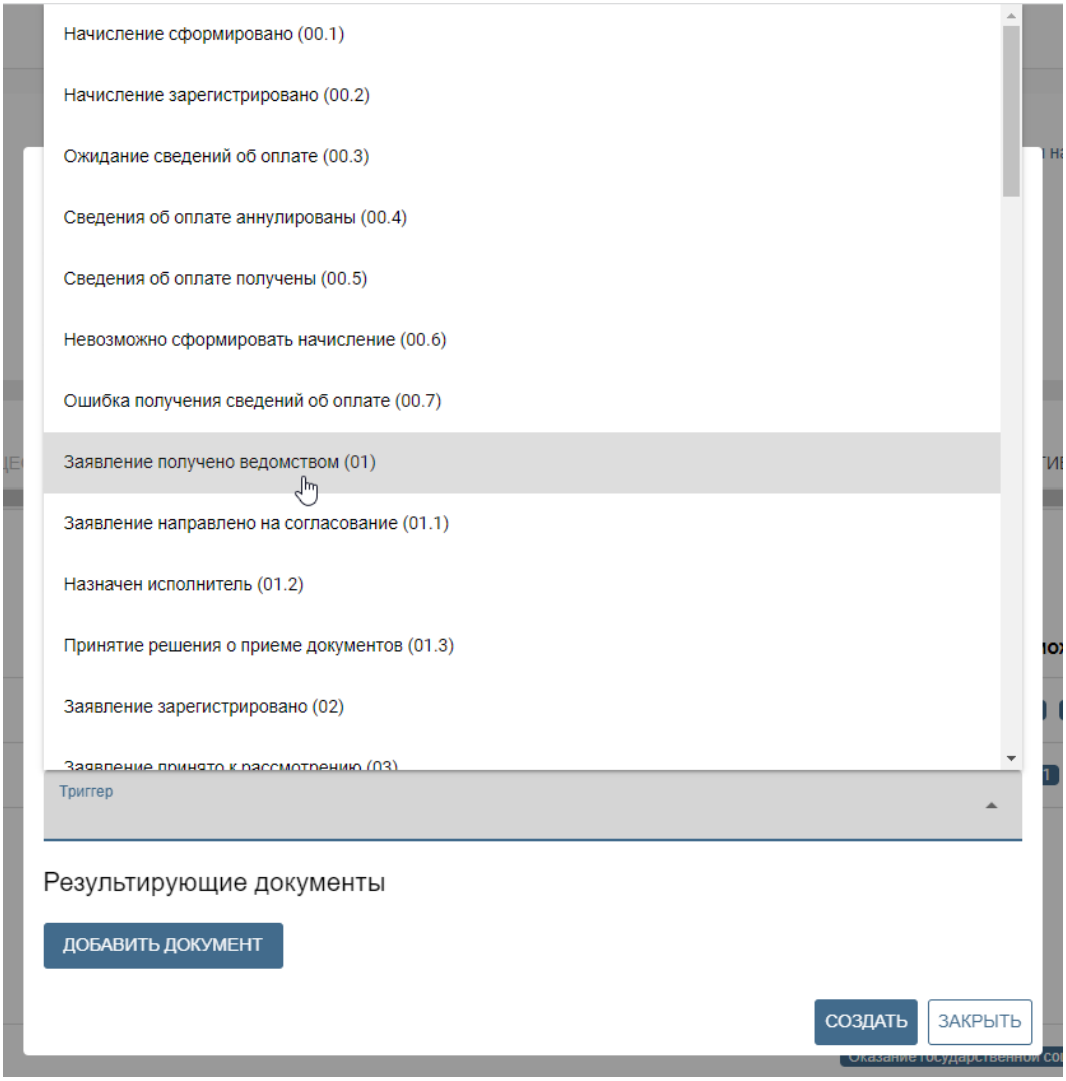


Рисунок 75 — Окно добавления административной процедуры в услугу. Выбор триггера перехода для административной процедуры

Для административных процедур доступны операции редактирования атрибутов (кнопка — «Редактировать») и удаления административной процедуры (кнопка — «Удалить»).

3.10. Работа с межведомственными запросами для предоставления услуги

3.10.1. Добавление межведомственных запросов для предоставления услуги

На вкладке карточки услуги «Межведомственные запросы» необходимо задать межведомственные запросы (МЗ), предусмотренные для услуги. Как правило, перечень необходимых МЗ содержится в документации на услугу.

На данной вкладке добавляются только МЗ, осуществляемые с использованием СМЭВ. Если в документации на услугу для какого-либо МЗ отсутствует ссылка на СМЭВ и указано, что ВС в СМЭВ отсутствует, то это означает, что данный МЗ выполняется «вручную» без использования СМЭВ. Данный «ручной» МЗ добавляется в услугу как документ категории «Документы, добавляемые специалистом» на вкладке карточки услуги «Документы» (см. подраздел 3.8).

Для добавления в услугу МЗ, осуществляемого с использованием СМЭВ, необходимо:

1. Перейти на страницу вида сведений, соответствующего данному МЗ, на технологическом портале СМЭВ 2 или СМЭВ 3. Как правило, ссылки на страницы видов сведений для МЗ, содержатся в документации на услугу.

2. Скопировать в буфер обмена значение поля с наименованием:

– «Namespace URI» — в случае если ВС размещен на портале СМЭВ 3 https://smev3.gosuslugi.ru (рисунок 76);

– «URI» — в случае если ВС размещен на портале СМЭВ 3 https://lkuv.gosuslugi.ru (рисунок 77);

– «Адрес описания» — в случае если ВС размещен на портале СМЭВ 2 https://smev.gosuslugi.ru (рисунок 78).

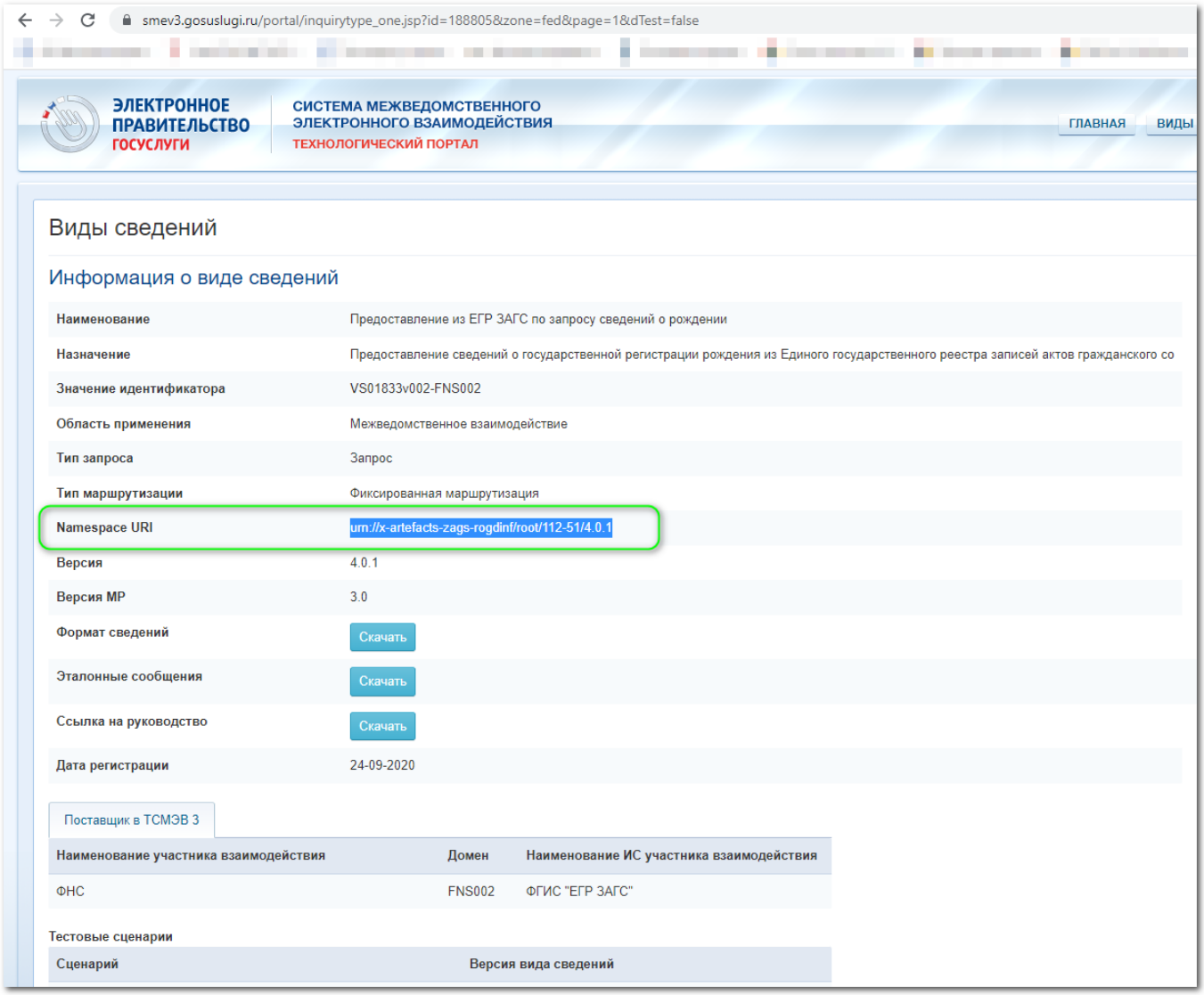


Рисунок 76 — Страница ВС на портале https://smev3.gosuslugi.ru . Поле «Namespace URI»

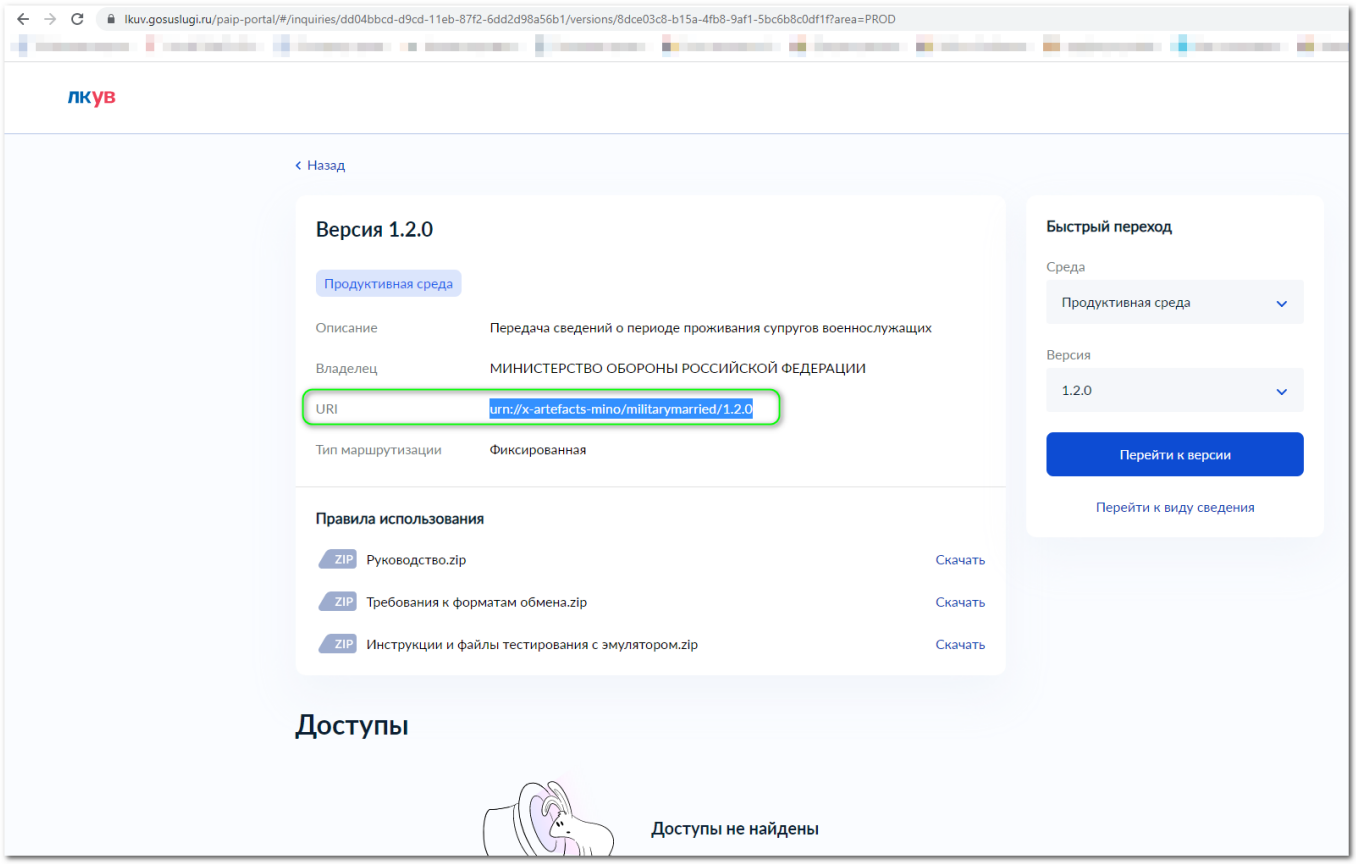


Рисунок 77 — Страница ВС на портале https://lkuv.gosuslugi.ru . Поле «URI»

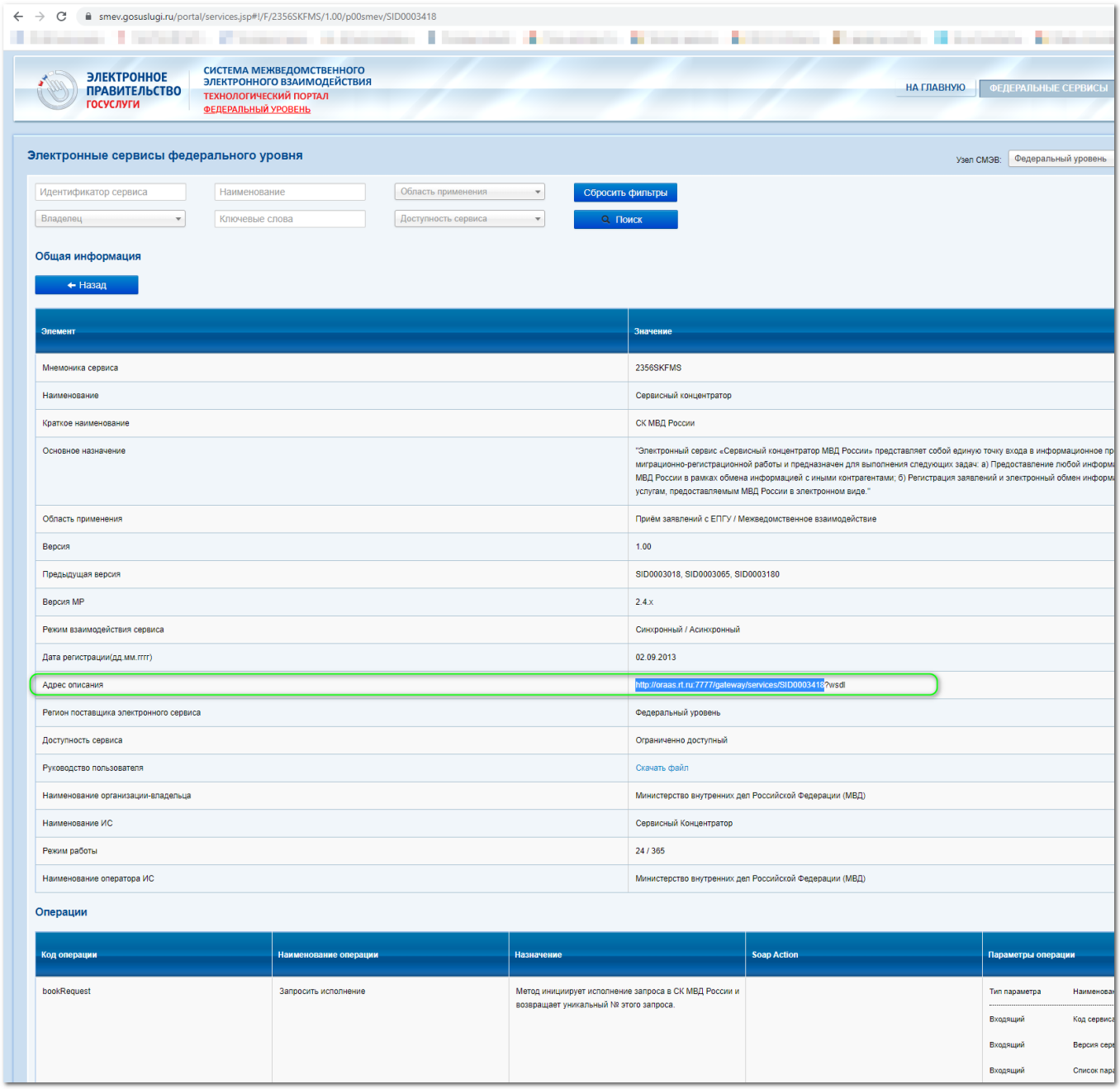


Рисунок 78 — Страница ВС на портале https://smev.gosuslugi.ru . Поле «Адрес описания»

3. В разделе ПГС «Конструктор услуг» найти услугу «Метаданные для МЗ» и перейти в ее карточку, на вкладку «Межведомственные запросы».

4. Используя функцию поиска (*Ctrl + F*), найти МЗ, содержащие в столбце «Класс» в качестве фрагмента значение из шага 2. Если таких МЗ более одного, определить требуемый МЗ.

5. Скопировать в буфер обмена значение в столбце «Идентификатор» для требуемого МЗ (рисунок 79).

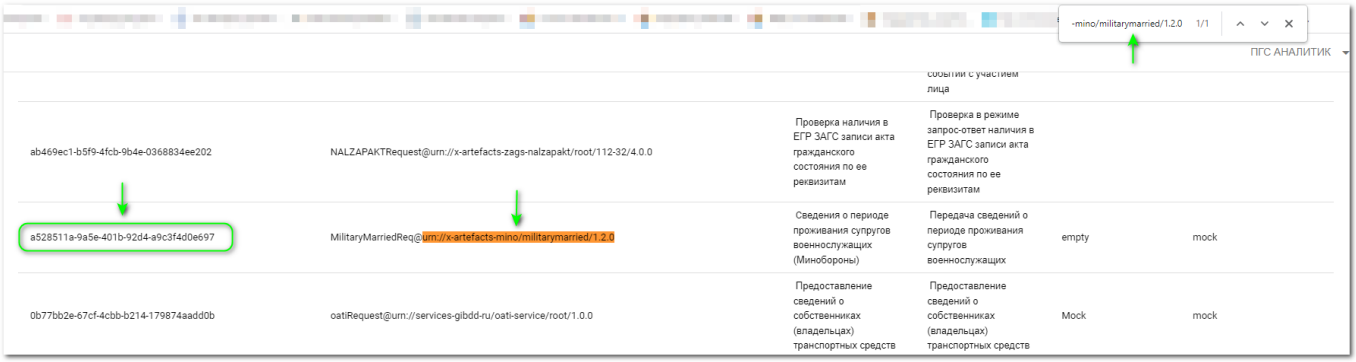


Рисунок 79 — Идентификатор требуемого МЗ
на вкладке «Межведомственные запросы» карточки услуги «Метаданные для МЗ»

6. Перейти в карточку настраиваемой услуги, на вкладку «Межведомственные запросы».

7. Нажать кнопку «Выбрать межвед» (рисунок 80).

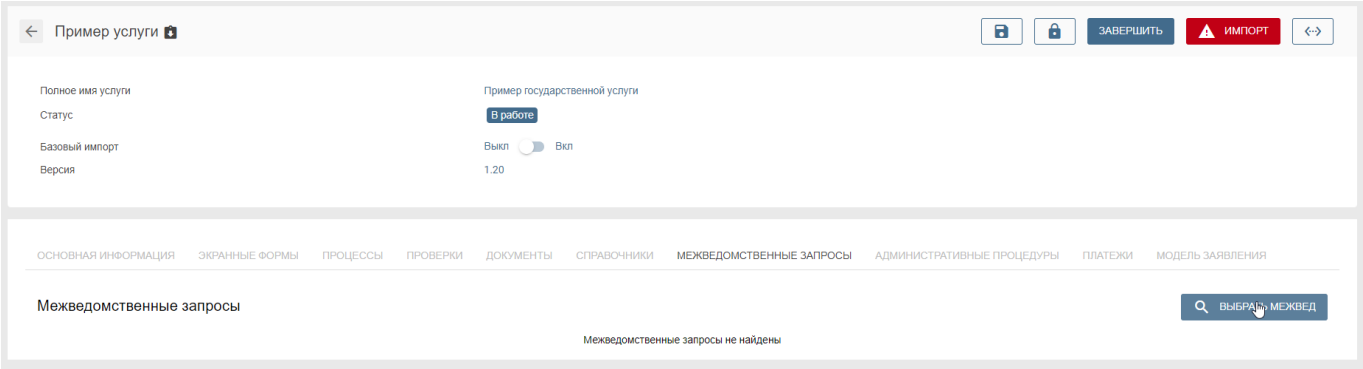


Рисунок 80 — Вкладка карточки услуги «Межведомственные запросы».
Кнопка «Выбрать межвед»

8. В открывшемся окне вставить значение идентификатора МЗ из буфера обмена в поисковое поле. Убедиться, что после значения отсутствуют пробелы, удалить их при необходимости. Выполнить поиск МЗ по указанному значению.

Примечание. Поиск в окне выбора МЗ может также осуществляться по наименованию МЗ. Например, могут быть найдены все МЗ, содержащие в наименовании фрагмент «ЗАГС», среди которых могут быть выбраны требуемые для услуги.

9. Установить отметку у найденного МЗ и нажать кнопку «Выбрать» (рисунок 81).

После этого выбранный МЗ появится на вкладке карточки настраиваемой услуги «Межведомственные запросы».

Действия 1–9 необходимо выполнить для каждого МЗ, предусмотренного для услуги.

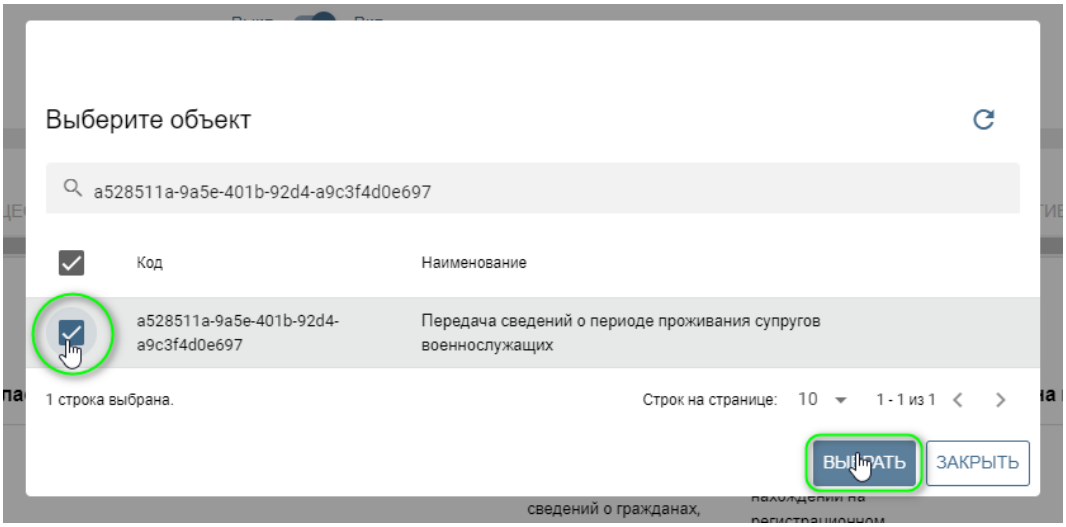


Рисунок 81 — Окно выбора МЗ на вкладке карточки услуги «Межведомственные запросы». Выбор МЗ с использованием функции поиска

3.10.2. Создание процесса межведомственного взаимодействия

Один из вариантов процесса МВ представлен на рисунке 82.

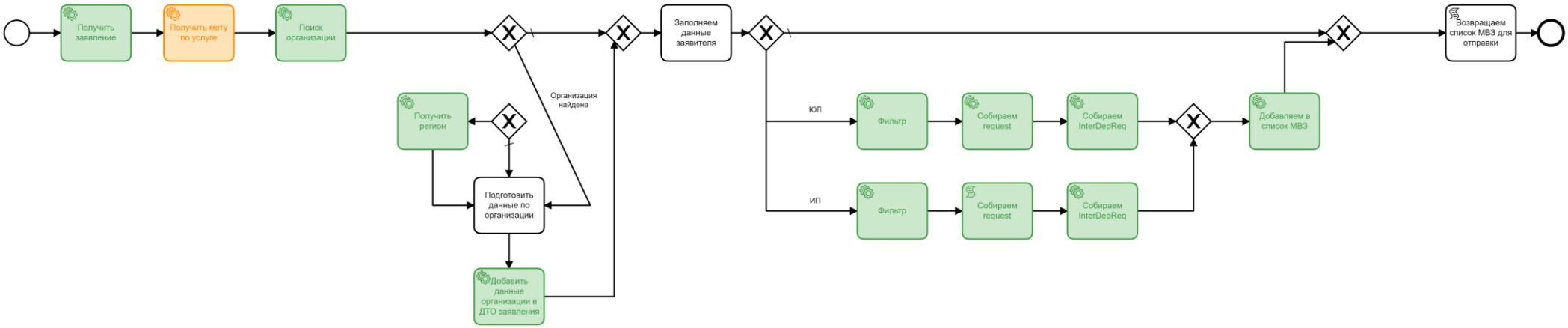


Рисунок 82 — BPMN-схема процесса МВ

В процессе МЗ имеются стандартные задачи в начале и самом конце процесса, которые используются в любом процессе МЗ (рисунок 83).

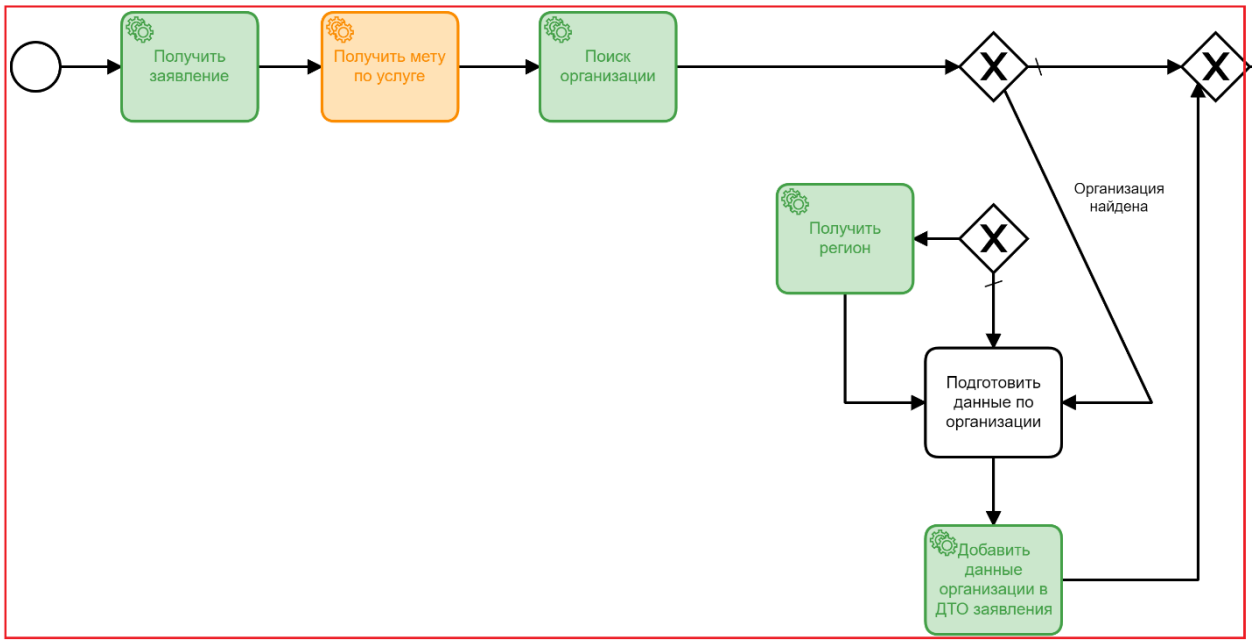


Рисунок 83 — Стандартные задачи в начале процесса МВ

Блок на рисунке 83 отвечает за получение информации о заявлении, услуге и дополнительной информации относительно ведомства (организации) и субъекта Российской Федерации.

После стандартных задач, добавляется первая уникальная для услуги задача (рисунок 84).

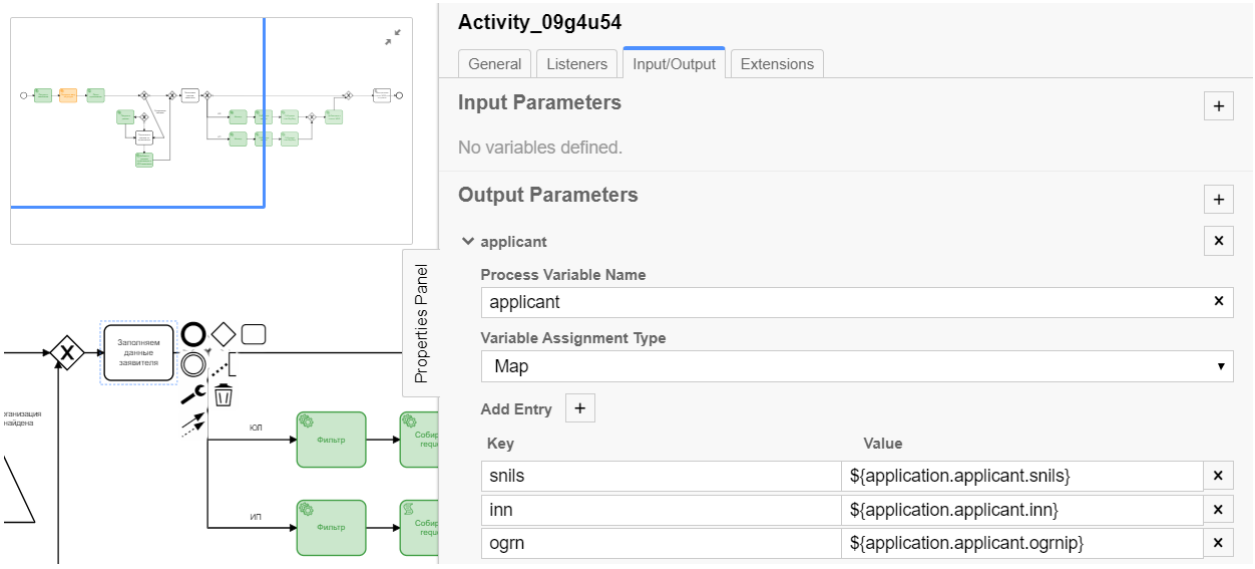


Рисунок 84 — Задача «Заполняем данные заявителя» в процессе МВ

В данной задаче создается переменная для заявителя, в которой указываются атрибуты, используемые в дальнейшем.

Установление соответствия атрибутов осуществляется следующим образом:

1. Создается исходящая переменная applicant.

2. Присваивается тип (в данном случае — «Соответствие» или Map).

3. Создаются переменные в формате «ключ-значение».

Значения для переменных получают из стандартных задач. В частности, обобщенная переменная application объявляется в задаче «Получить заявление» посредством отправки GET-запроса.

После данной задачи заканчивается этап подготовки данных для МЗ.

Любой МЗ собирается из четырех задач (рисунок 85):

1. «Фильтр» — поиск объекта МЗ в услуге.

2. «Собираем request» — создание запроса.

3. «Собираем InterDepReq» — добавление запроса и дополнительной информации в типовую переменную.

4. «Добавляем в список МВЗ» — добавление созданной переменной в общий список МЗ для отправки.

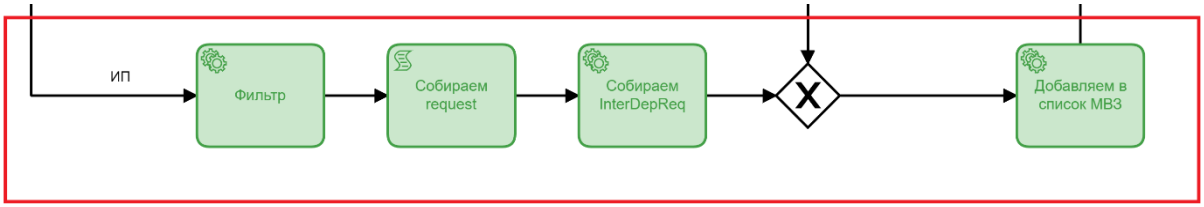


Рисунок 85 — Задачи, задающие МЗ, в процессе МВ

На вкладке General («Общее»)[1] задачи «Фильтр» (рисунок 86) указывается, какое имя будет у новой созданной и заполненной переменной. Переменная создается на основании данных, полученных в результате обработки данных со вкладки Input/Output («Ввод/Вывод») — рисунок 87.

[1] Во встроенном BPMN-конструкторе ПГС вкладки на панелях параметров элементов процессов имеют названия на русском языке. В Camunda Modeler соответствующие вкладки имеют названия на английском языке. Здесь и далее названия таких вкладок приводятся на обоих языках.

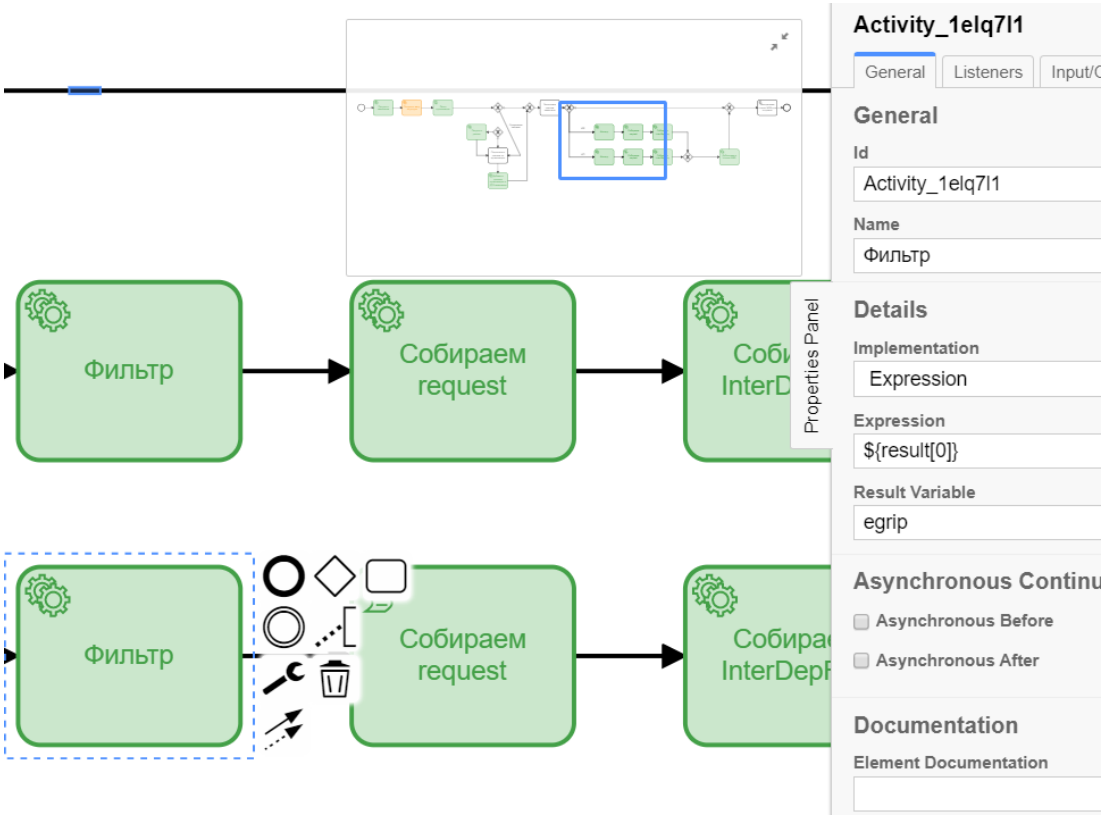


Рисунок 86 — Параметры задачи «Фильтр». Вкладка General («Общее»)

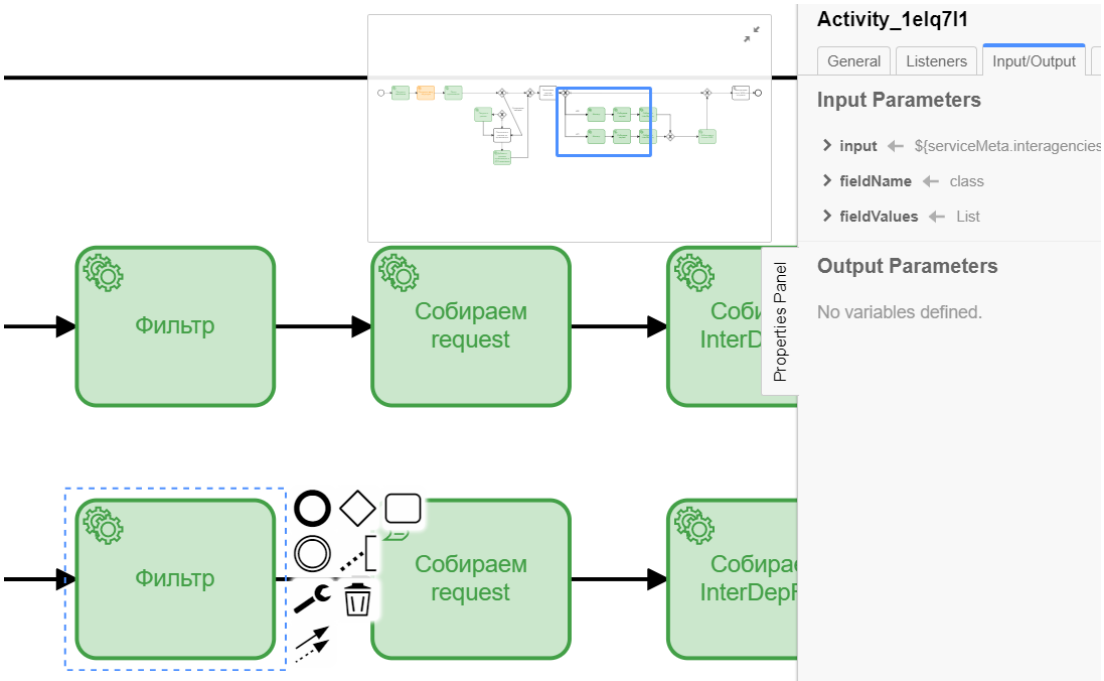


Рисунок 87 — Параметры задачи «Фильтр». Вкладка Input/Output («Ввод/Вывод»)

Переменные на данной вкладке:

– input — содержит обращение к переменной, которая берется из услуги, — в данном случае *${serviceMeta.interagencies}*, где serviceMeta — полное описание услуги, полученное в задаче «Получить мету по услуге», interagencies — все сущности МЗ, которые были добавлены в услугу;

– fieldName — содержит строку class, которая указывает, по какому атрибуту будет осуществляться поиск необходимого МЗ в общем списке из переменной Input;

– fieldValues — содержит строку, которая указывает, какому значению должна соответствовать переменная class в коллекции interagencies.

Для фильтрации используется команда со вкладки Extensions («Расширения») — рисунок 88.

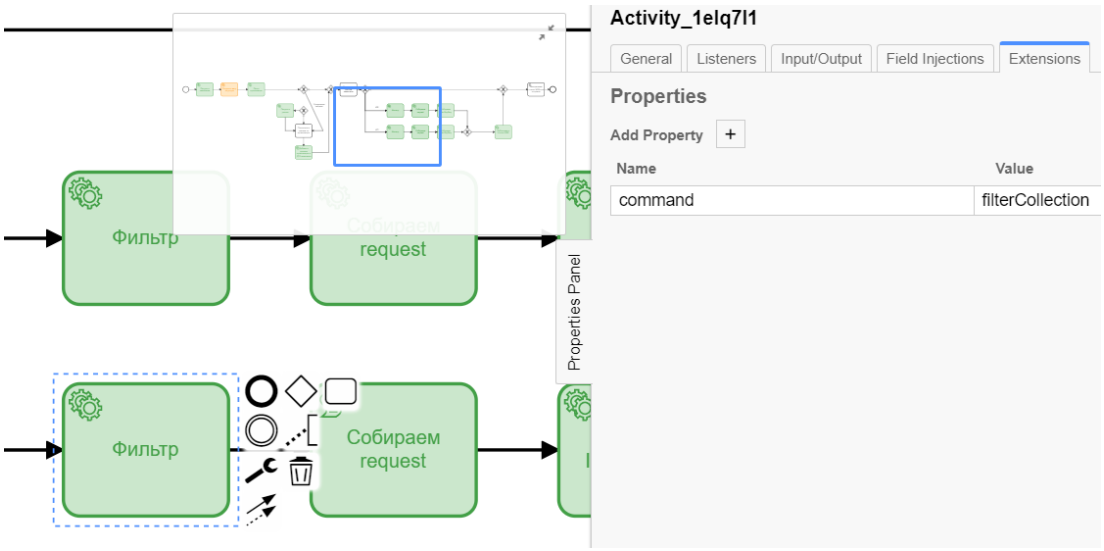


Рисунок 88 — Параметры задачи «Фильтр». Вкладка Extensions («Расширения»)

Данная команда по введенным параметрам вернет результат фильтрации коллекции МЗ в новую переменную со вкладки General.

После задачи «Фильтр» следует специальная задача-скрипт (рисунок 89).

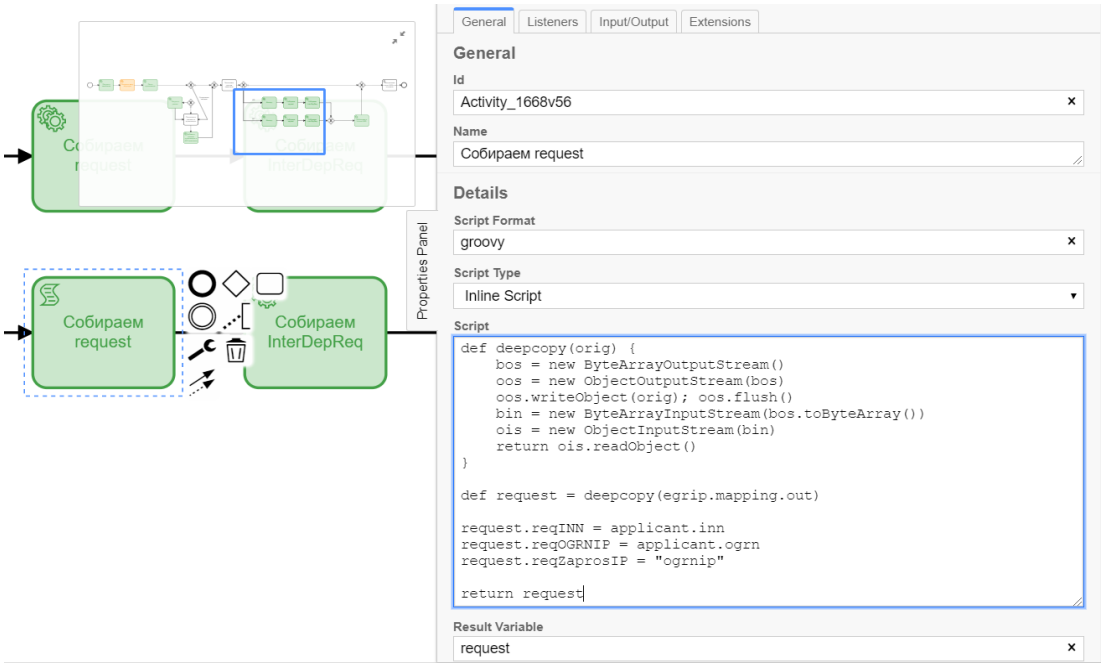


Рисунок 89 — Параметры задачи «Собираем request». Вкладка General

Для задачи-скрипта «Собираем request» необходимо указать:

– язык скрипта (в поле «Script Format») — используется groovy;

– тип скрипта (в поле «Script Type») — выбирается из списка значение «Inline Script» («Встроенный сценарий»);

– текст скрипта (в поле для ввода скрипта Script);

– название результирующей переменной (в поле «Result Variable»).

Верхний блок скрипта является стандартным для всех МЗ:

```
def deepcopy(orig) {
    bos = new ByteArrayOutputStream()
    oos = new ObjectOutputStream(bos)
    oos.writeObject(orig); oos.flush()
    bin = new ByteArrayInputStream(bos.toByteArray())
    ois = new ObjectInputStream(bin)
    return ois.readObject()
}
```

Нижняя часть скрипта используется для задания соответствий атрибутов будущего запроса (для данного МЗ и данной услуги). Пример:

```
def request = deepcopy(egrip.mapping.out)

request.reqINN = applicant.inn
request.reqOGRNIP = applicant.ogrn
request.reqZaprosIP = "ogrnip"

return request
```

Здесь:

– reqINN — ИНН из заявления;

– reqOGRNIP — ОГРН из заявления;

– reqZaprosIP — тип запроса в соответствии с руководством пользователя вида сведений.

После выполнения задачи-скрипта выполняется задача по формированию оставшихся атрибутов для МЗ запроса «Собираем InterDepReq» (рисунок 90).

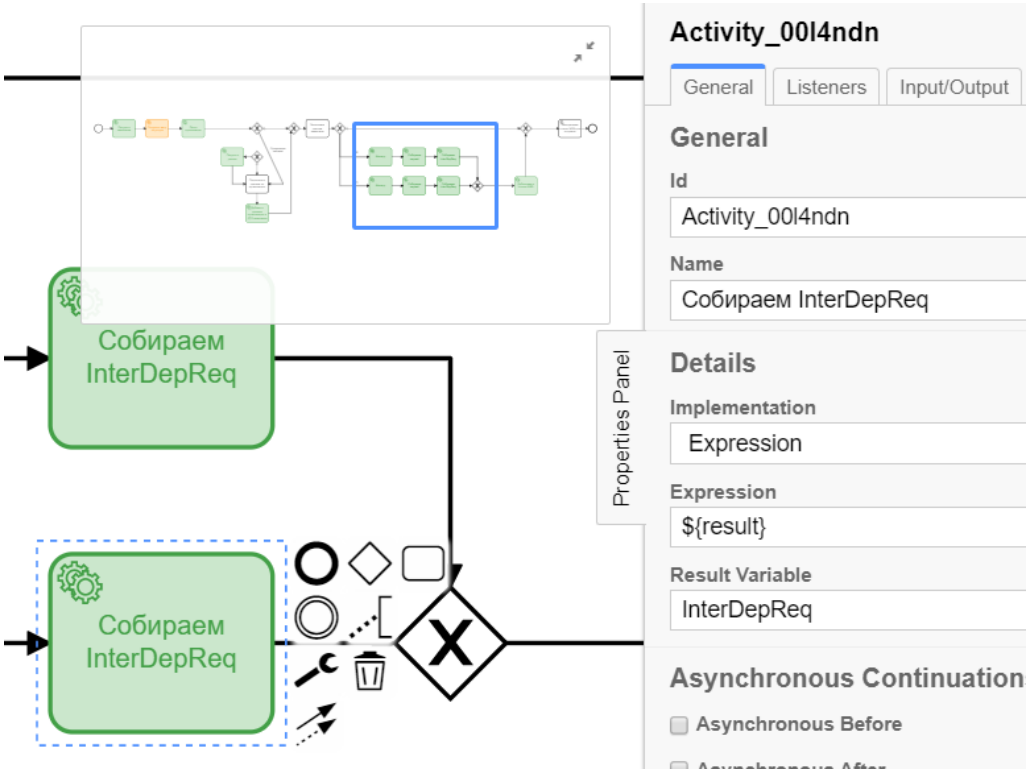


Рисунок 90 — Параметры задачи «Собираем InterDepReq». Вкладка General

Как и для задачи «Фильтр», на вкладке General (рисунок 90) указывается, какое имя будет у новой созданной и заполненной переменной, а переменная создается на основании данных, полученных в результате обработки данных со вкладки Input/Output (рисунок 91). Разница между задачами заключается в команде на вкладке Extensions (рисунок 92), которая используется для генерации и\или наполнения переменной, а также в параметрах, которые используются командой. Данная команда осуществляет обогащение переменной, указанной на вкладке General, используя данные со вкладки Input/Output.

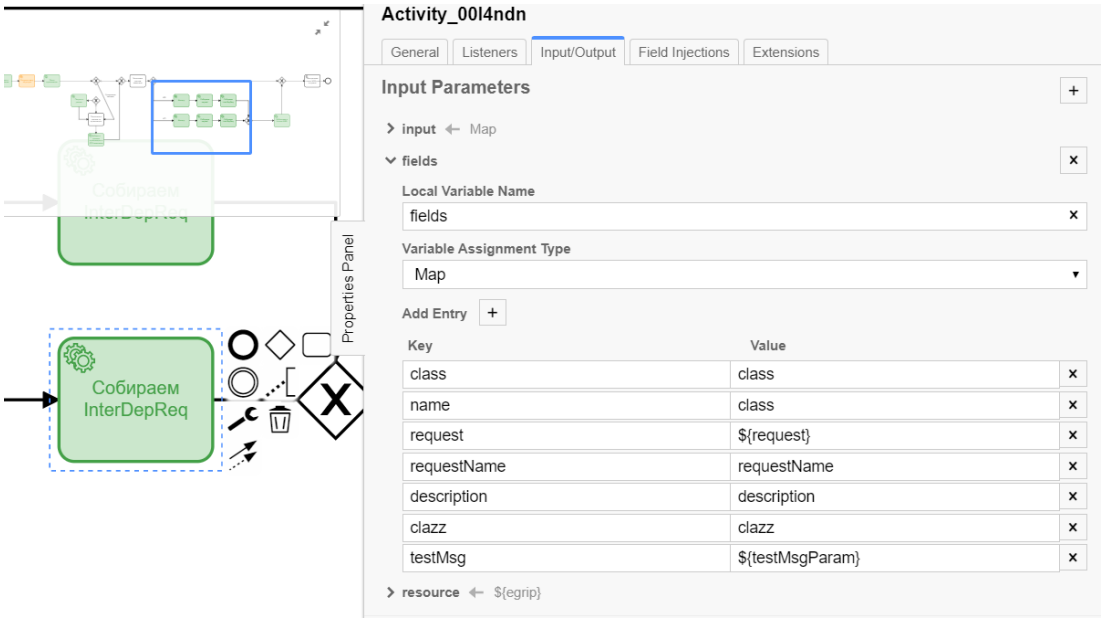


Рисунок 91 — Параметры задачи «Собираем InterDepReq». Вкладка Input/Output

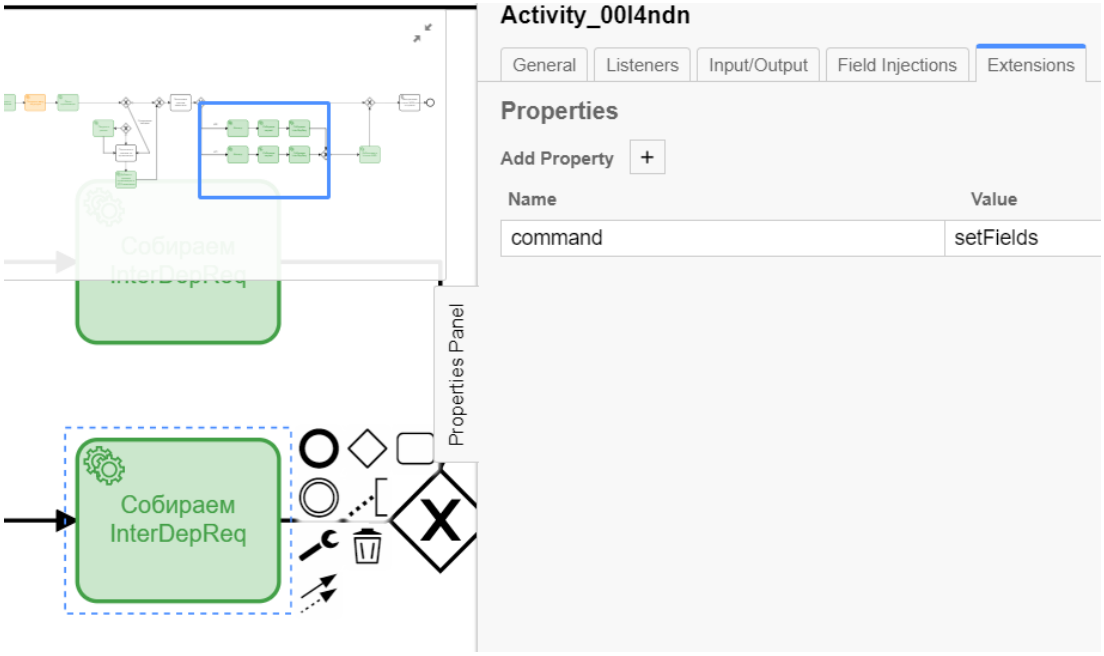


Рисунок 92 — Параметры задачи «Собираем InterDepReq». Вкладка Extensions

Для данной команды используются следующие параметры:

– input — пустой, поскольку переменной для обогащения ранее не создавалось;

– fields — таблица соответствия атрибутов объекта МЗ и атрибутов, полученных в ходе процесса создания запроса;

– resource — переменная, которая берется за основу для наполнения.

После выполнения команды новая сформированная переменная переходит в следующую задачу: «Добавляем в список МВЗ» (рисунок 93).

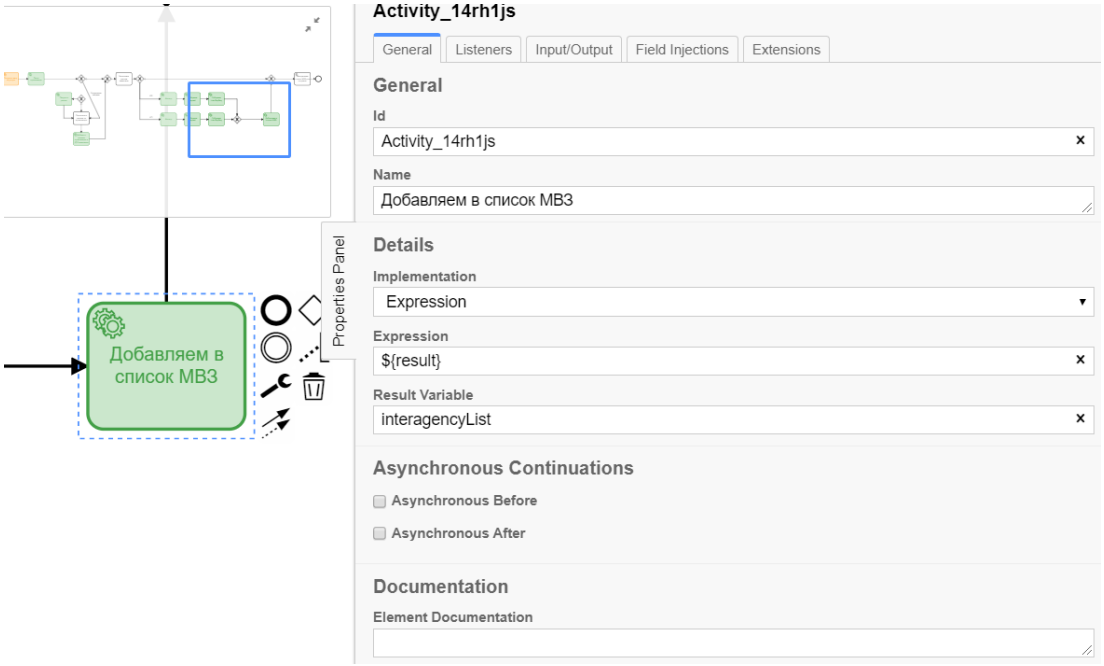


Рисунок 93 — Параметры задачи «Добавляем в список МВЗ». Вкладка General

Для данной задачи используется собственная команда на вкладке Extensions (рисунок 94).

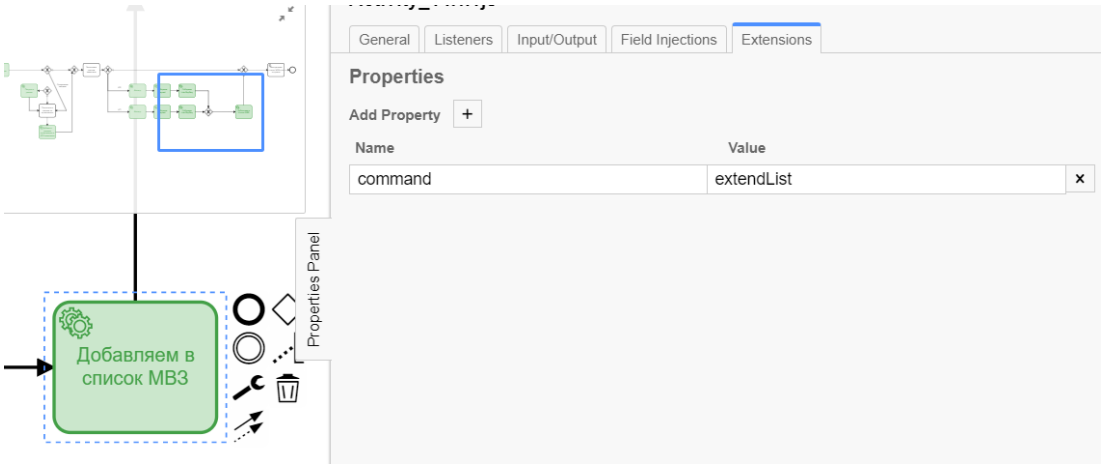


Рисунок 94 — Параметры задачи «Добавляем в список МВЗ». Вкладка Extensions

Данная команда принимает на вход две коллекции и объединяет их в одну общую переменную (в данном случае interagencyList) — рисунок 95.

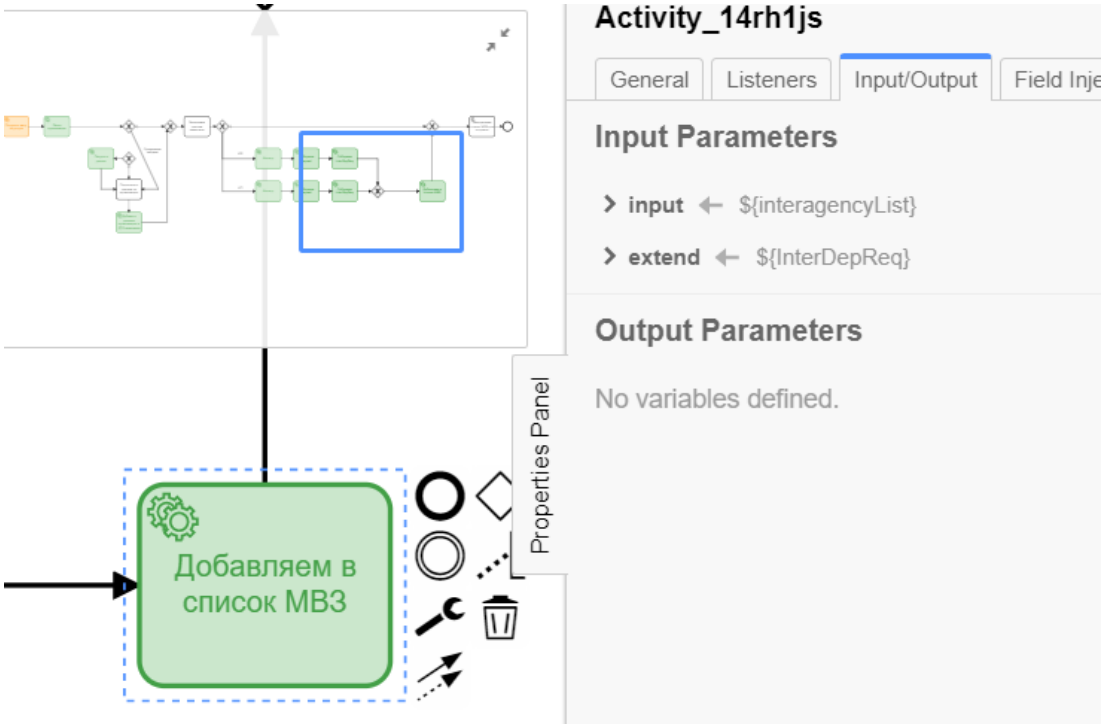


Рисунок 95 — Параметры задачи «Добавляем в список МВЗ». Вкладка Input/Output

Для данной команды используются следующие параметры:

– input — коллекция, которая будет дополняться (принимается как главная);

– extend — коллекция, которая дополняет (принимается как второстепенная).

Задача «Добавляем в список МВЗ» является последней для одного МЗ. После сбора всех запросов выполняется задача, предназначенная для создания общей переменной всего процесса processData.Interagencies.

В данную переменную будут помещены все запросы, собранные в ходе процесса МЗ.

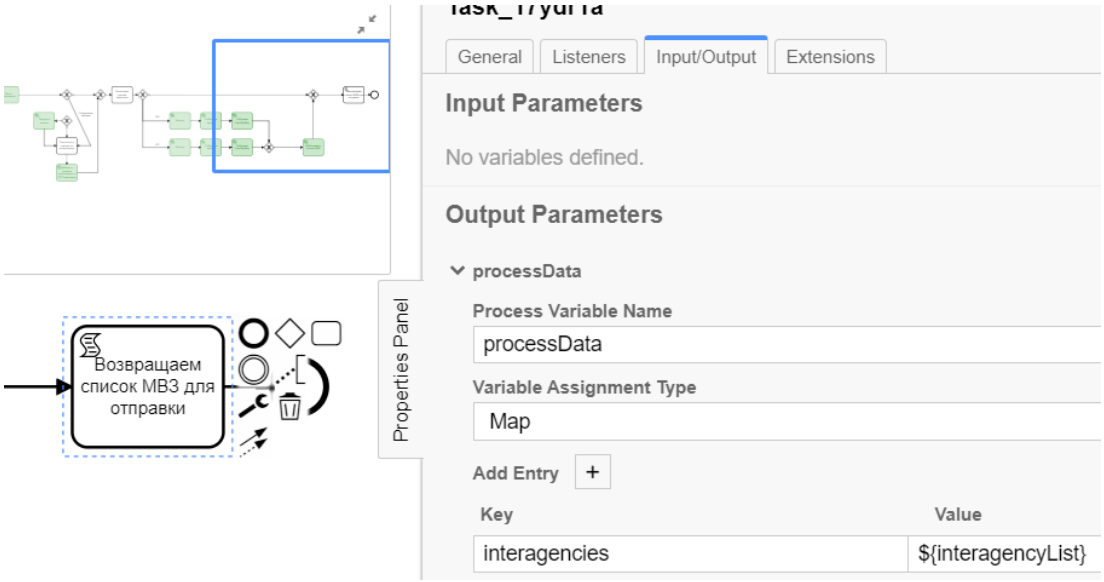


Рисунок 96 — Параметры задачи «Возвращаем список МВЗ для отправки». Вкладка Input/Output

3.10.3. Создание волн межведомственного взаимодействия (если необходимо)

Для некоторых услуг может возникать ситуация, когда все требуемые для услуги МЗ не могут быть выполнены одновременно в рамках одного процесса, а какие-либо МЗ (один или более) могут быть выполнены только после получения данных в результате других МЗ, поскольку требуется передача этих данных в запросах.

В этом случае требуется создать два или более процессов МВ, которые выполняются последовательно: каждый последующий процесс выполняется только после получения данных в результате выполнения предыдущего процесса. Такие процессы называются волнами МВ. При этом первый выполняемый процесс (волна) является основным, а последующие — дополнительными.

Основной процесс создается как процесс стандартного типа «Межведомственное взаимодействие». Дополнительные процессы должны иметь пользовательский тип additionalInteragency и такое же наименование, как у основного процесса, с добавлением постфиксов «_wave_2», «_wave_3» и т. д.

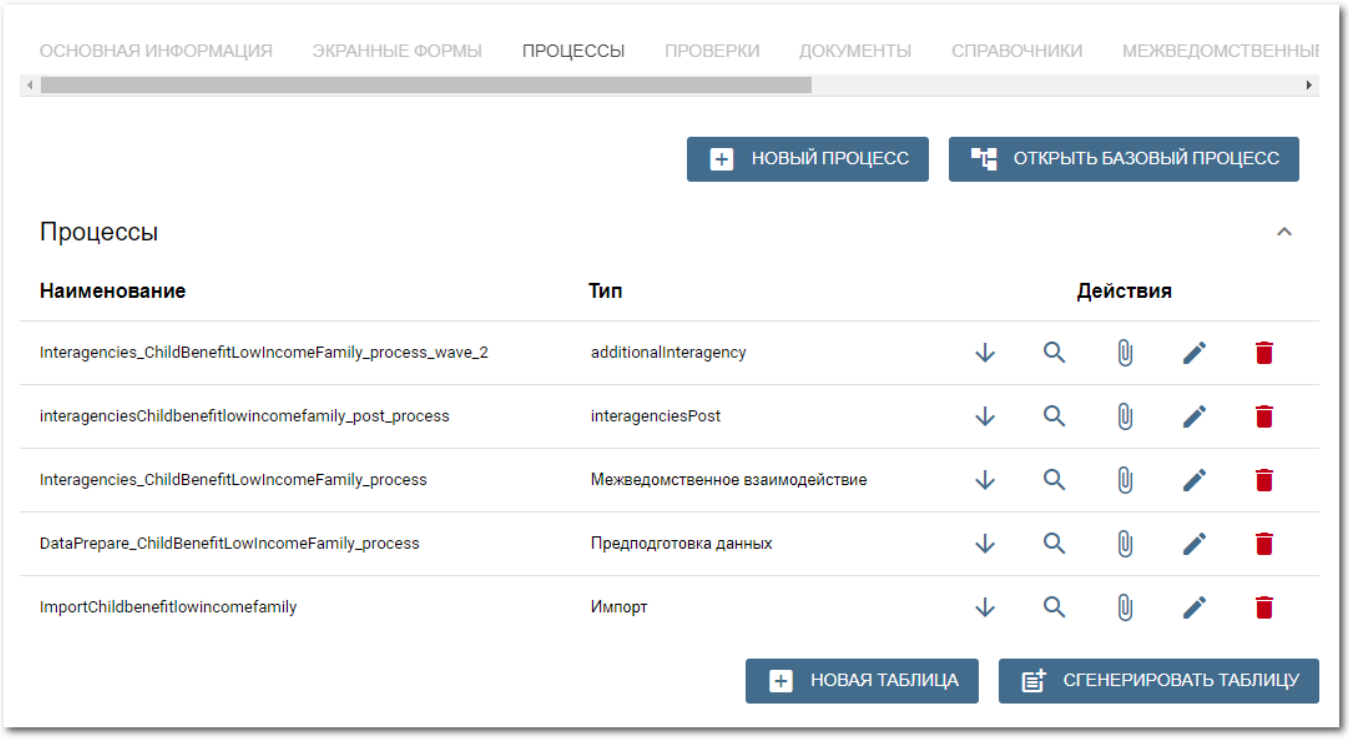


Рисунок 97 — Процессы (волны) МВ, завершающий процесс МВ
на вкладке карточки услуги «Процессы»

Помимо основного и дополнительных процессов МВ на вкладке карточки услуги «Процессы» необходимо создать для каждого дополнительного процесса по объекту «Волна», который определяет условия запуска данного дополнительного процесса.

Объект «Волна» (рисунок 98) добавляется при помощи кнопки «Добавить волну». В окне создания волны (рисунок 99) необходимо выбрать дополнительный процесс МВ и выбрать один или несколько классов МЗ в соответствующих выпадающих списках (рисунки 100 и 101 соответственно). Классы МЗ соответствуют МЗ, предусмотренным для услуги (см. подраздел 3.10).

Примечание. Класс МЗ определяет МЗ аналогично идентификатору, однако, в отличие от идентификатора, не зависит от версии МЗ.

Тем самым дополнительный процесс МВ будет запускаться автоматически только после получения сведений в результате выполнения всех МЗ, которым соответствуют выбранные классы МЗ.

Примечание. Результаты выполнения МЗ следует смотреть в карточках заявлений о предоставлении услуги в разделе «Заявления», по которым выполняется или был выполнен процесс предоставления услуги, в том числе были выполнены МЗ.

Примечание. Помимо процессов типа additionalInteragency, в зависимости от особенностей процесса предоставления услуги, также может требоваться завершающий процесс, который выполняется после завершения выполнения всех волн МВ. Например, такой процесс может включать в себя некоторые преобразования данных, полученных в результате межведомственного

взаимодействия. Завершающий процесс должен иметь пользовательский тип interagenciesPost. Для него объект «Волна» не создается. Такой процесс запускается автоматически после выполнения всех процессов типа additionalInteragency.

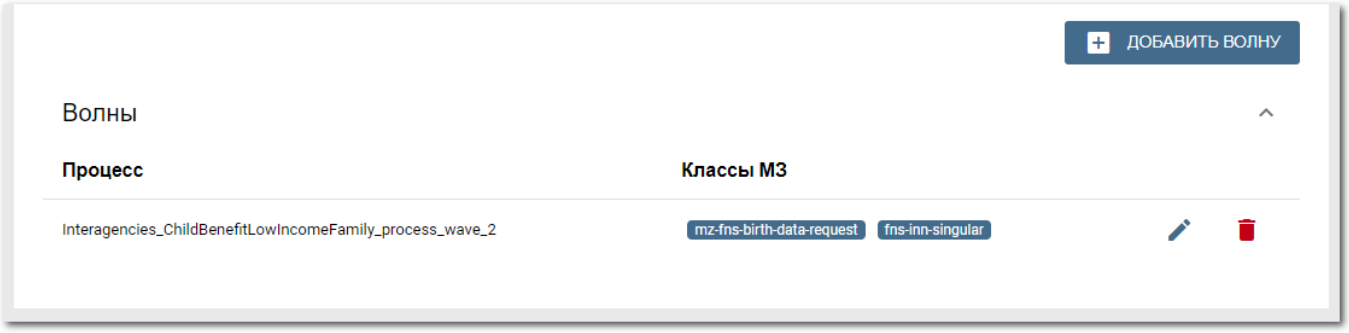


Рисунок 98 — Объект «Волна» на вкладке карточки услуги «Процессы»

Рисунок 99 — Окно добавления объекта «Волна» в услугу

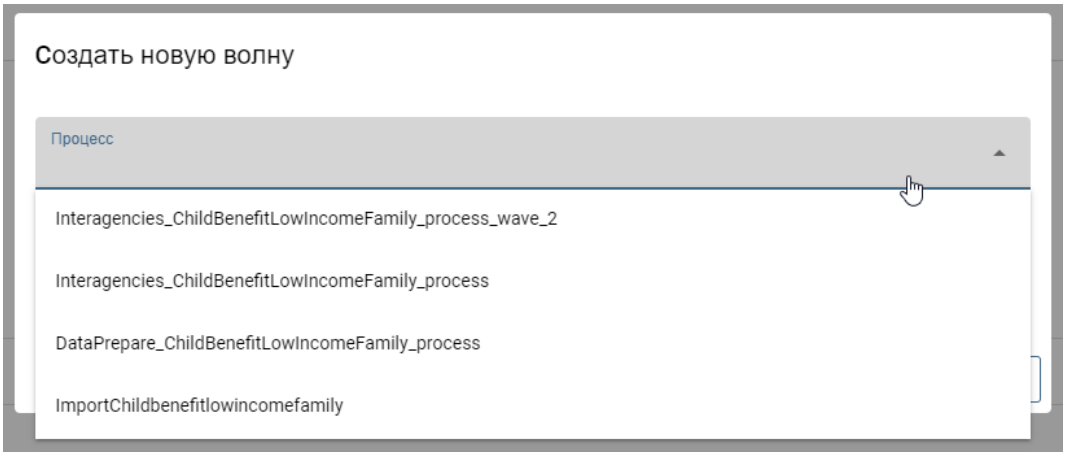


Рисунок 100 — Окно добавления объекта «Волна» в услугу.
Выпадающий список процессов, предусмотренных для услуги

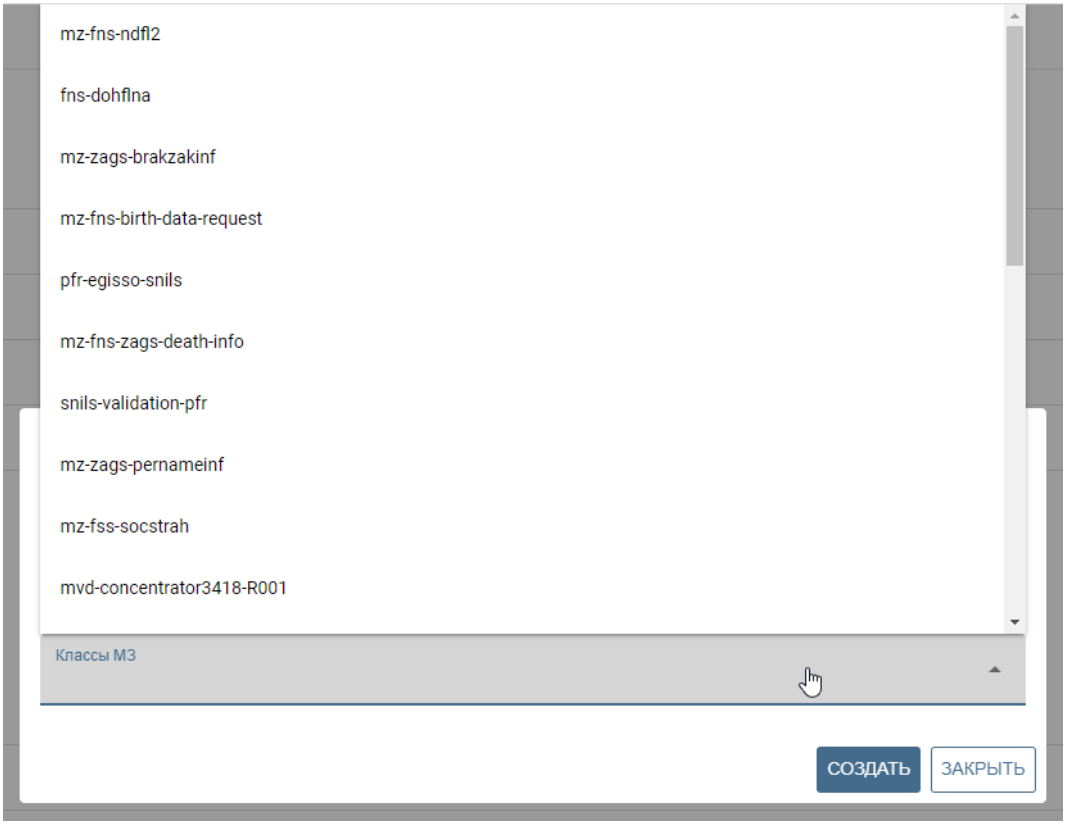


Рисунок 101 — Окно добавления объекта «Волна» в услугу.
Выпадающий список классов МЗ, предусмотренных для услуги

В процесс второй волны МВ необходимо добавить элемент (задачу) «Фильтр <наименование МЗ>»[1], где <наименование МЗ> — наименование МЗ первой волны, данные из ответа на который должны использоваться в МЗ второй волны. Данный элемент (аналогично элементу «Фильтр» в процессе МВ первой волны — см. подраздел 3.10.2) содержит команду FilterCollection (рисунок 102), которая выделяет из полного набора данных, полученных в результате всех МЗ первой волны, данные того МЗ, которые требуются для второй волны. Одним из параметров команды FilterCollection является переменная fieldValues, в качестве значения которой необходимо указать класс МЗ, ответы на который используются во второй волне (рисунок 103).

Если вторая волна должна использовать данные ответов на более чем один МЗ первой волны, то в процесс второй волны необходимо добавить по задаче-фильтру, соответствующей каждому из этих МЗ. Если процессов МВ типа additionalInteragency в услуге более одного, то задачи-фильтры необходимо добавить в каждый из них, указывая в качестве значений переменной fieldValues классы требуемых МЗ из процесса, предшествующего данному.

Примечание. Рисунки и описание далее приводятся на примере, когда вторая волна МВ зависит от МЗ в ЕГР ЗАГС.

[1] Может также называться «Получение данных из МВЗ <наименование МЗ>» или аналогично. Наименование и сам элемент можно скопировать из первой волны.

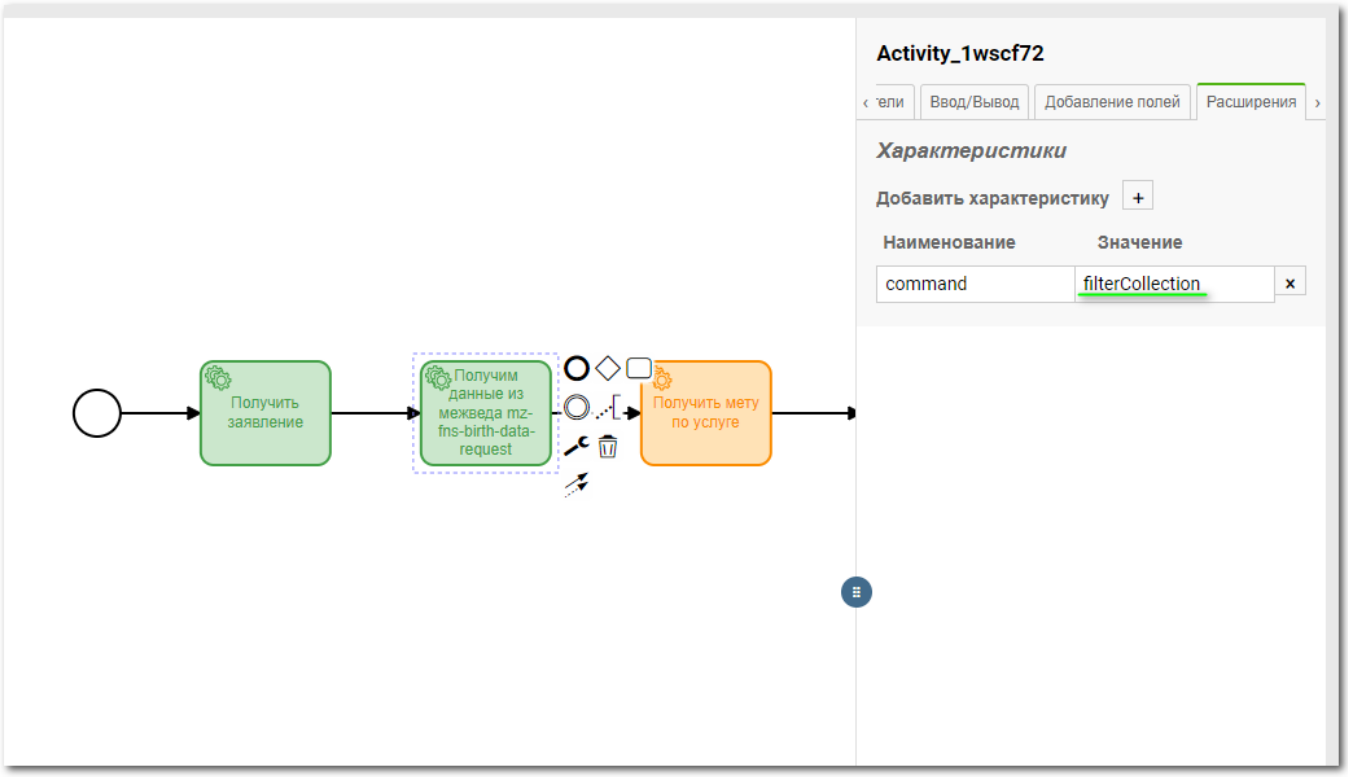


Рисунок 102 — Параметры задачи-фильтра процесса типа additionalInteragency. Команда filterCollection на вкладке «Расширения» (Extensions)

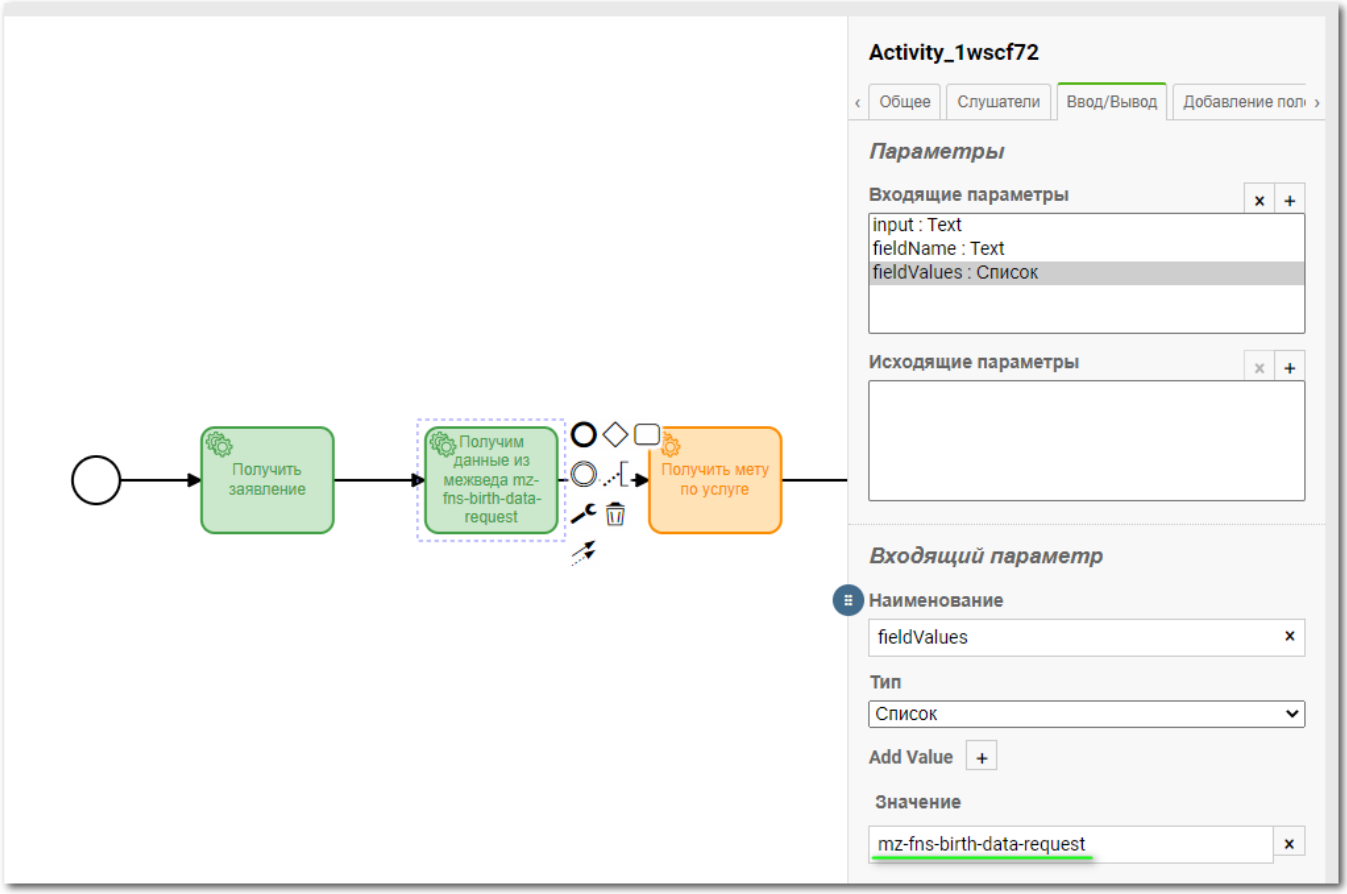


Рисунок 103 — Параметры задачи-фильтра процесса типа additionalInteragency. Класс МЗ, указанный в качестве значения переменной fieldValues на вкладке «Ввод/Вывод» (Input/Output)

Перед задачами «Собираем request», «Собираем InterDepReq», «Добавляем в список МВЗ», соответствующими МЗ второй волны, должны находиться задачи получения данных для МЗ второй волны из заявления и из МЗ первой волны. В рассматриваемом примере такими задачами являются «Собираем ФИО, СНИЛС» и «Собираем series, number, date» соответственно (рисунки 104 и 105, 106 соответственно).

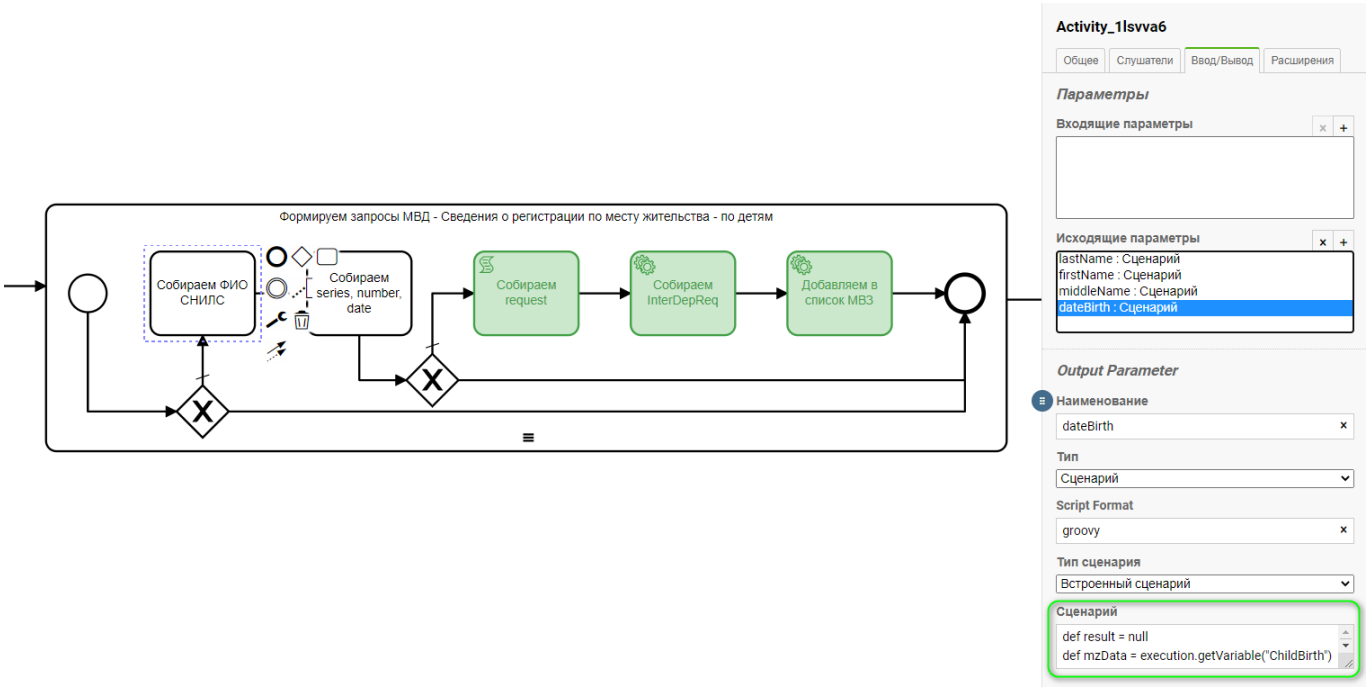


Рисунок 104 — Параметры задачи «Собираем ФИО, СНИЛС» процесса типа additionalInteragency. Вкладка «Ввод/Вывод» (Input/Output)

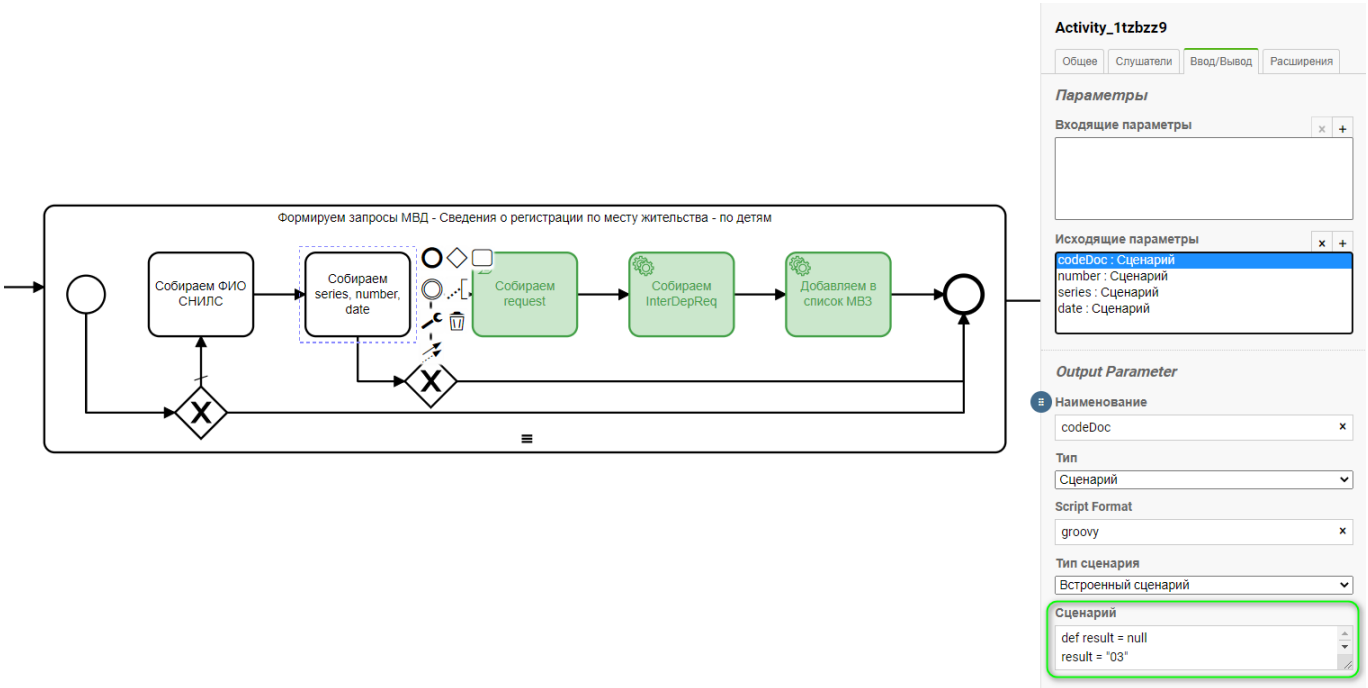


Рисунок 105 — Параметры задачи «Собираем series, number, date» процесса типа additionalInteragency. Вкладка «Ввод/Вывод» (Input/Output). (Здесь «03» в сценарии, являющемся значением переменной codeDoc, — код свидетельства о рождении ребенка)

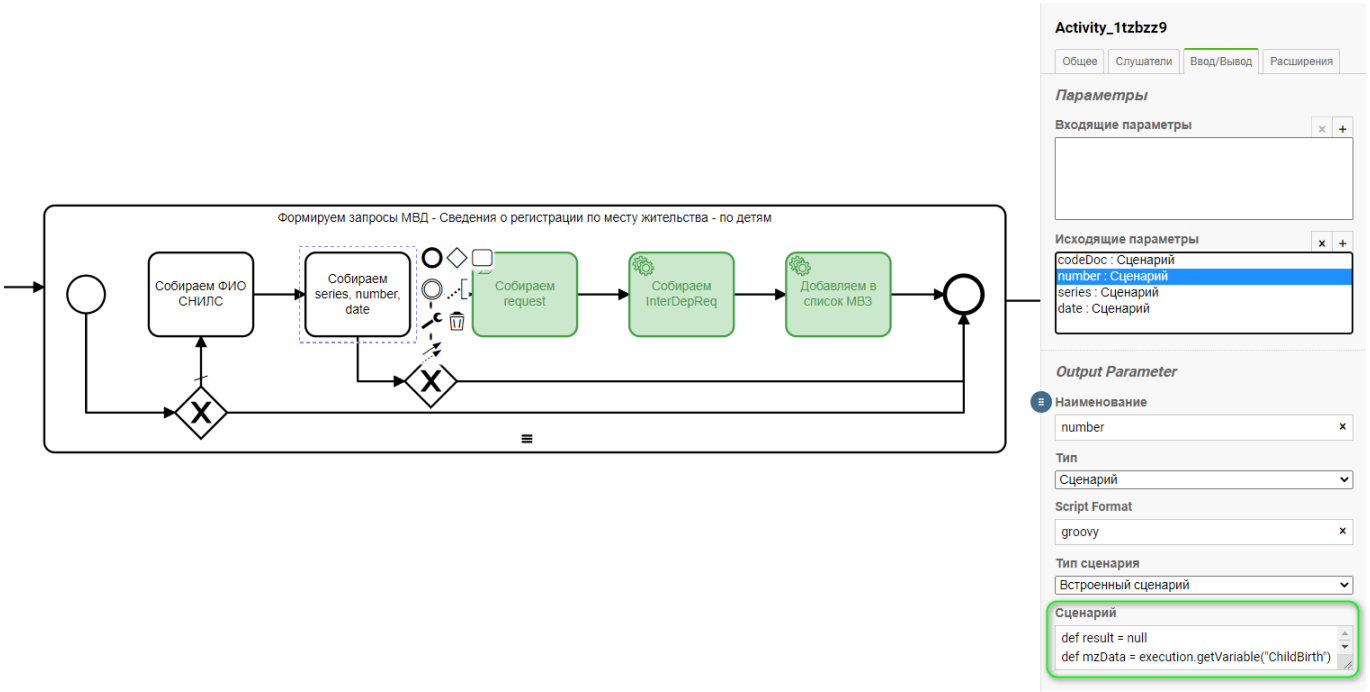


Рисунок 106 — Параметры задачи «Собираем series, number, date» процесса типа additionalInteragency. Вкладка «Ввод/Вывод» (Input/Output). Значение переменной number

Пример скрипта сбора номера, указываемого в качестве значения переменной number в задаче «Собираем series, number, date» (рисунок 106):

```
def result = null
def mzData = execution.getVariable("ChildBirth")

            for(b in ChildBirth) {
                if (lastName == b?.request?.reqSvedZapros?.reqSvedFlFamilia &&
firstName == b?.request?.reqSvedZapros?.reqSvedFlImya) {

                    def                          kodDoc                          =
b?.response?.svedOtvetAgs?.getAt(0)?.svedRegRozjd?.getAt(0)?.svedRojd?.getAt(
0)?.nomerSvidet?.toString()
                    if (kodDoc != null) {
                        result = kodDoc
                    }
                }
            }

        return result
```

В задаче «Собираем request» указывается скрипт, использующий ранее собранные данные (рисунок 107):

```
def deepcopy(orig) {
     bos = new ByteArrayOutputStream()
     oos = new ObjectOutputStream(bos)
     oos.writeObject(orig); oos.flush()
     bin = new ByteArrayInputStream(bos.toByteArray())
     ois = new ObjectInputStream(bin)
     return ois.readObject()
}
```

```
def utils = execution.getVariable("utils")

def request = deepcopy(mvdR001Child.mapping.out)

request.params.param[0]                                                    =
[name:"CITIZEN_LASTNAME",value:data.childrenData.lastname]
request.params.param[1]                                                    =
[name:"CITIZEN_FIRSTNAME",value:data.childrenData.firstname]
request.params.param[2]                                                    =
[name:"CITIZEN_GIVENNAME",value:data.childrenData.middlename]
request.params.param[3]                                                    =
[name:"CITIZEN_BIRTHDAY",value:data.childrenData.dateBirth]
request.params.param[4]                                                    =
[name:"CITIZEN_SNILS",value:data.childrenData.snils.replaceAll("[^0-9]", "")]
request.params.param[5] = [name:"DOC_TYPE",value:"03"]
request.params.param[6] = [name:"DOC_SERIE",value:series]
request.params.param[7] = [name:"DOC_NUMBER",value:number]
request.params.param[8] = [name:"DOC_ISSUEDATE",value:date]
request.params.param[9] = [name:"REGION_CODE",value:regionCode]

return request
```

Собранные данные содержатся в переменных вида value:data.childrenData.lastname, value:data.childrenData.firstname и т. п., которые указываются в правых частях выражений вида «request.params.param[0] = [name:"CITIZEN_LASTNAME",value:data.childrenData.lastname]», «request.params.param[1] = [name:"CITIZEN_FIRSTNAME",value:data.childrenData.firstname]» и т. д.

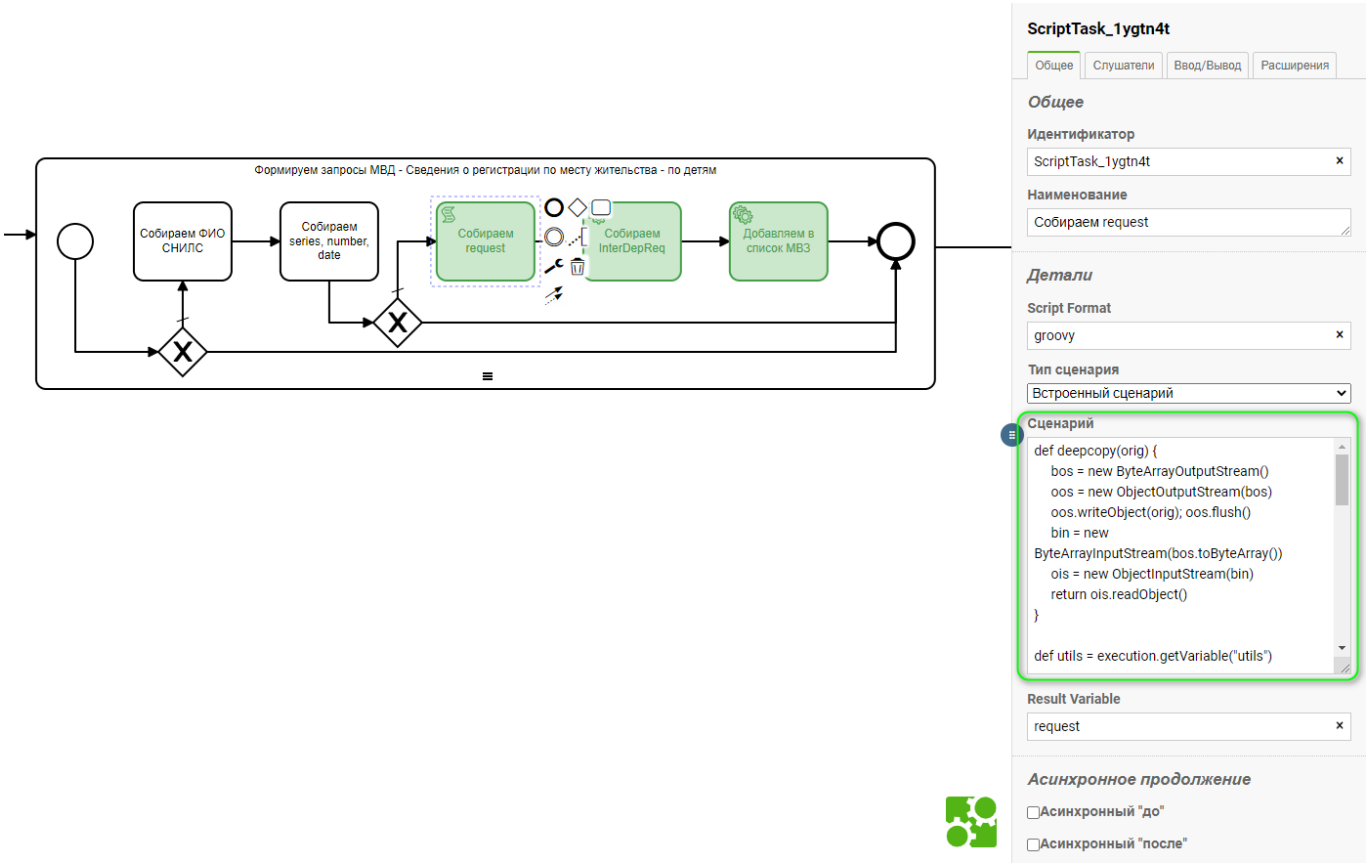


Рисунок 107 — Параметры задачи «Собираем request» процесса типа additionalInteragency. Скрипт на вкладке «Общее» (General)

3.11. Настройка проверок

Настройка проверок, выполняемых во время предоставления услуги, осуществляется на вкладке карточки услуги «Проверки» (рисунок 108).

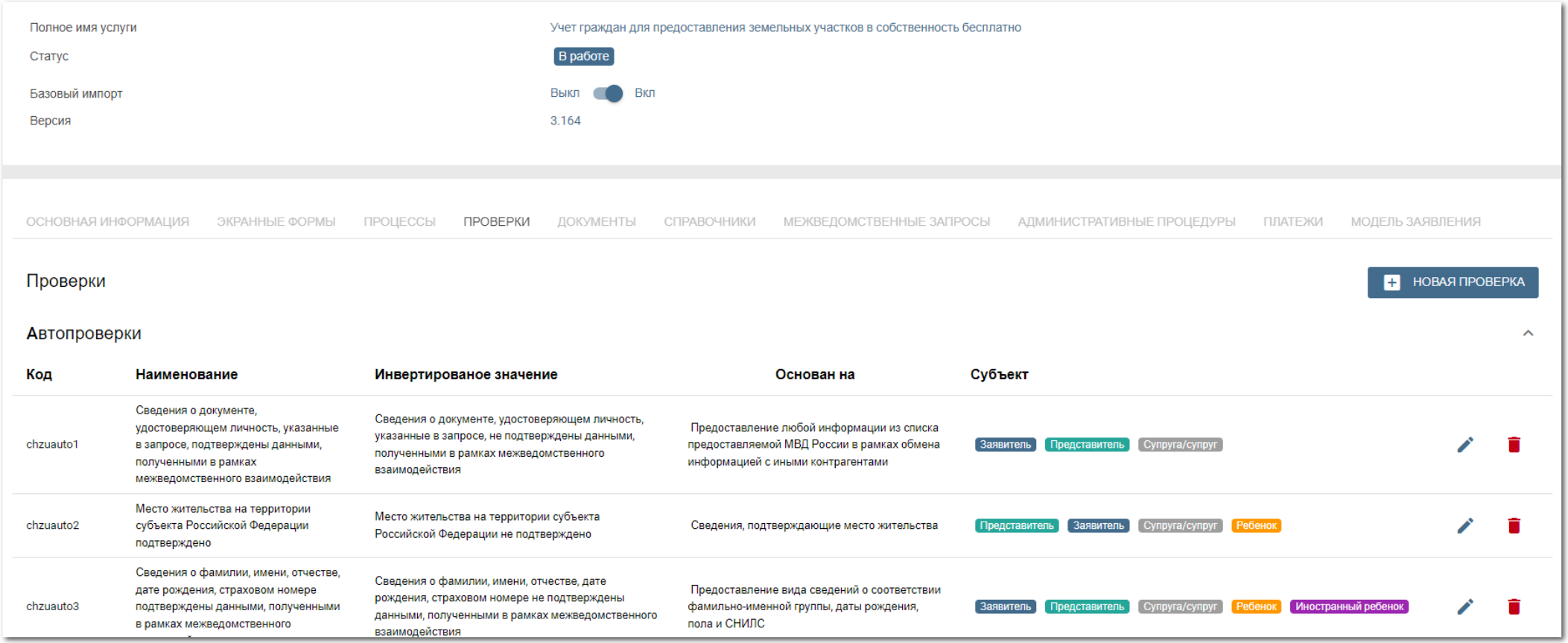


Рисунок 108 — Проверки на вкладке карточки услуги «Проверки».

Для добавления проверки следует нажать кнопку «Новая проверка» (рисунок 108) и на открывшейся странице (рисунок 109):

– задать:

а) код проверки (уникальное в рамках услуги значение, состоящее из латинских букв, цифр и символов) — в поле «Код»;

б) наименование проверки — в поле «Наименование»;

в) значение, которое выводится в качестве результата проверки, если проверка не пройдена, — в поле «Инвертированное значение»;

г) условия, при которых проверка должна выполняться автоматически или отображаться специалисту (см. ниже);

– установить или снять отметку «С комментарием» (установленная отметка означает, что при выполнении проверки обязателен комментарий специалиста, исполняющего процесс предоставления услуги);

– выбрать:

а) режим выполнения проверки: вручную или автоматически — в выпадающем списке «Тип»;

б) сущность, на которой основана проверка: заявление о предоставлении услуги, документ или МЗ — в выпадающем списке «Основан на — Тип» (значения: document, mvz и application соответственно);

в) если проверка основана на заявлении, указать код и наименование атрибута заявления, участвующего в проверке; если проверка основана на документе или МЗ, выбрать требуемый документ или МЗ в выпадающем списке «Основан на — Наименование»[1];

г) код отказа — в выпадающем списке «Основан на — Код отказа».

Выполнение автоматических проверок настраивается в процессе подготовки данных (см. подраздел 3.13). Ручные проверки осуществляются специалистом, исполняющим процесс предоставления услуги, при соблюдении всех указанных в проверке условий.

Значениями в выпадающем списке «Основан на — Код отказа» (рисунок 110) являются основания для отказа в приеме документов и в предоставлении услуги, указанные на вкладке «Справочники» (см. подраздел 3.7). В случае если проверка будет не пройдена, заявителю будет отказано в приеме документов или предоставлении услуги с указанным основанием.

[1] Выпадающие списки МЗ и документов состоят из МЗ и документов, добавленных на вкладках «Межведомственные запросы» и «Документы» соответственно (см. подразделы 3.8 и 3.10 соответственно).

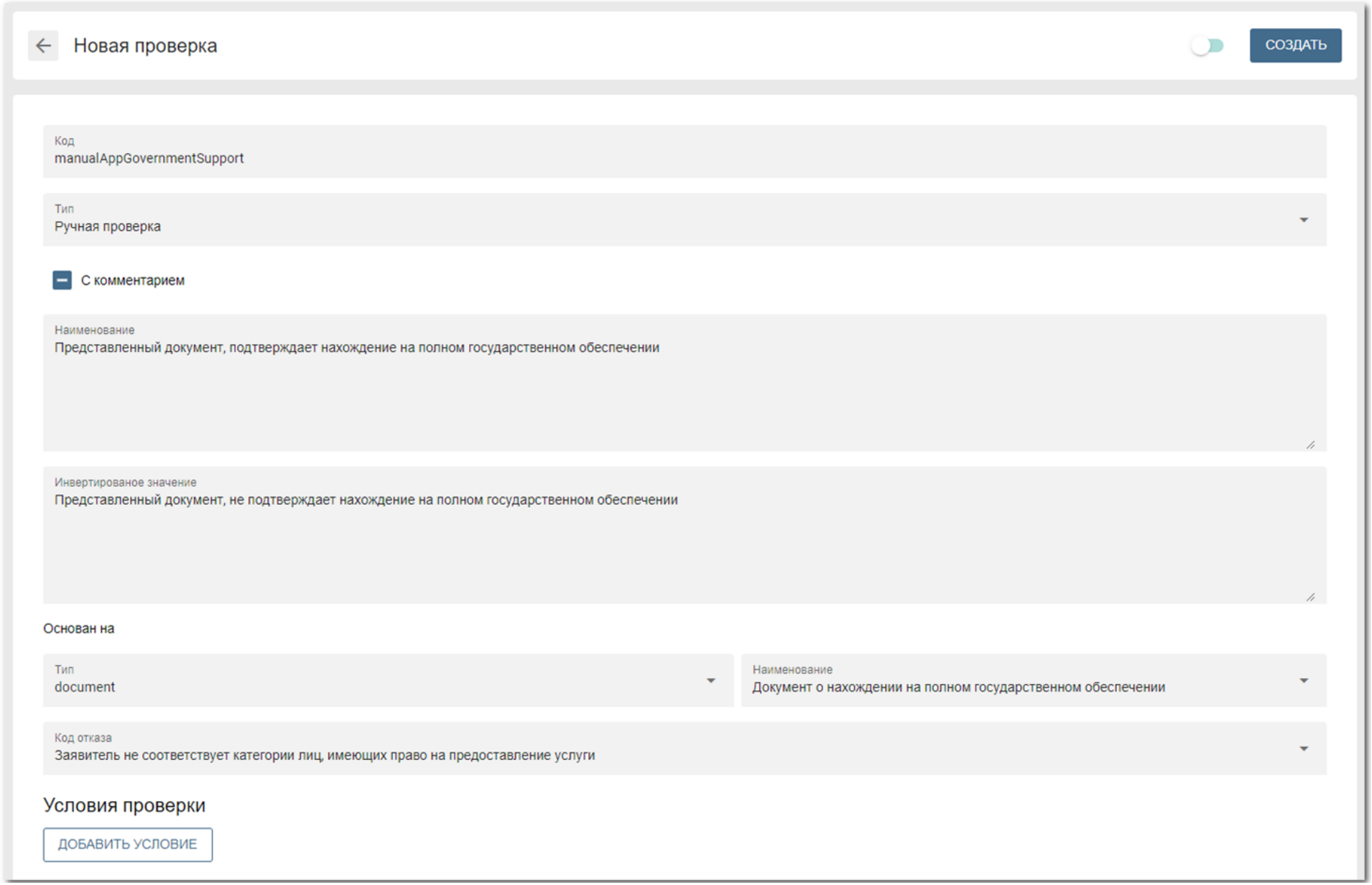


Рисунок 109 — Страница добавления проверки

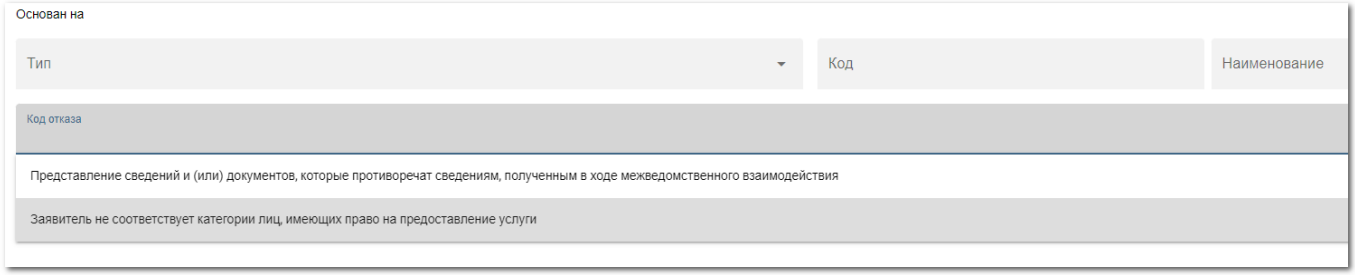


Рисунок 110 — Выпадающий список оснований для отказа

Для добавления условия выполнения проверки необходимо нажать кнопку «Добавить условие». В появившейся форме следует:

– выбрать тип условия в выпадающем списке «Тип условия»:

а) application — проверка применяется к атрибуту заявления;

б) subject — проверка применяется к субъекту заявления;

– если проверка должна применяться к субъекту заявления, выбрать требуемого субъекта в выпадающем списке «Субъект»[1];

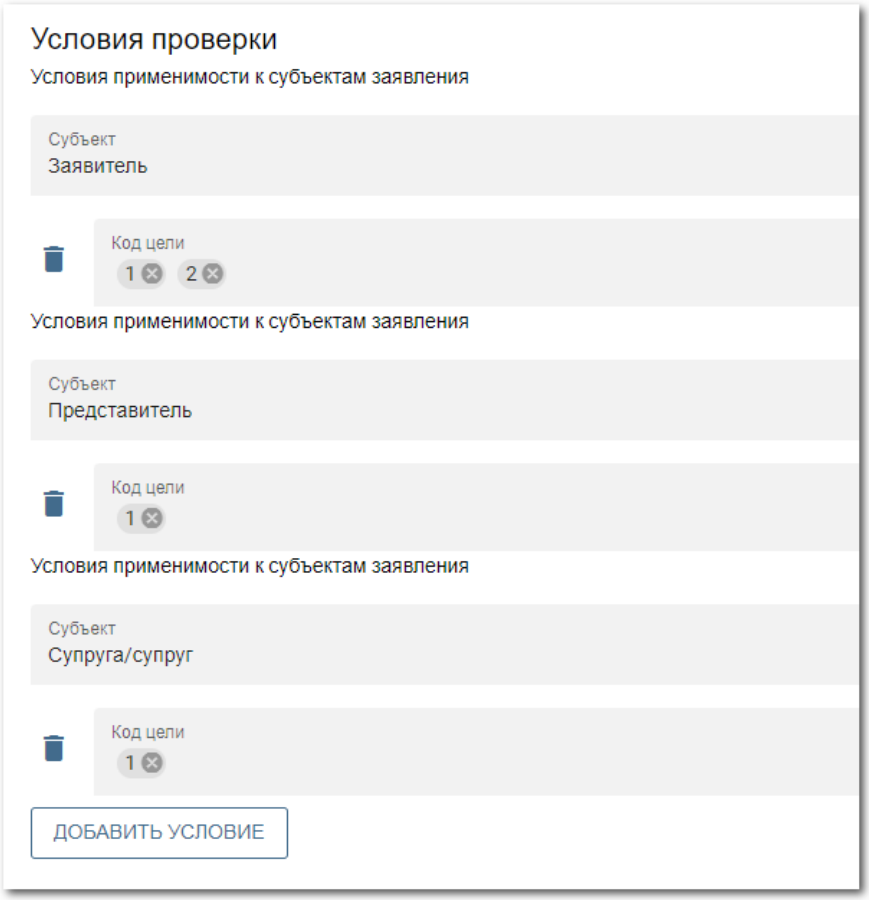
– выбрать логическое выражение для условия в выпадающем списке «Тип выражения»: «И», «Или», «Исключающее или» или «Базовое выражение»;

– если проверка должна применяться к той или иной подуслуге, то установить отметку «Является целью» и в появившемся выпадающем списке выбрать соответствующую цель;

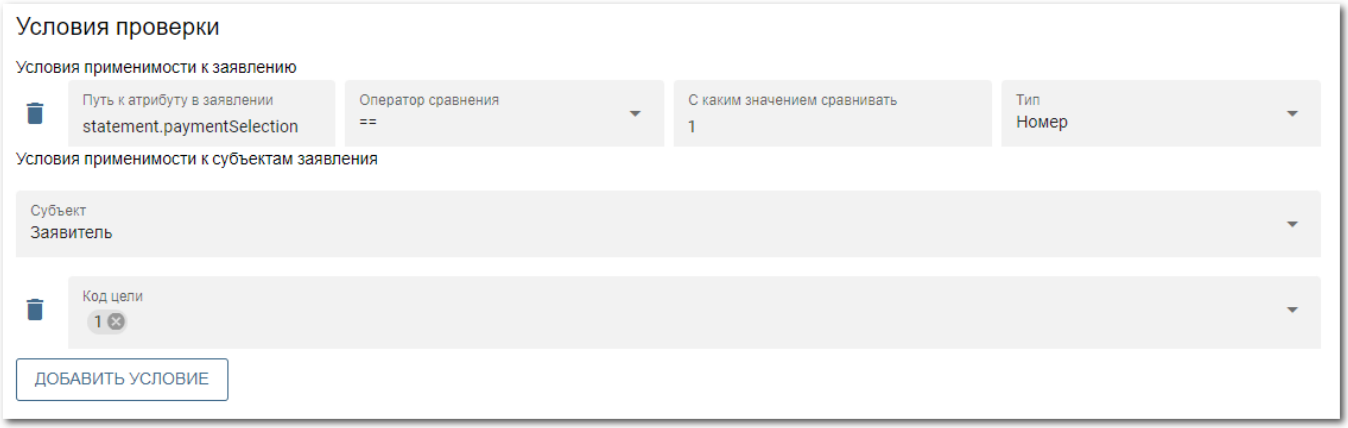
– если проверка не должна применяться к подуслуге, то в полях «Атрибут субъекта», «Оператор сравнения», «С каким значением сравнивать» и «Тип» задать выражение, являющееся условием применения проверки.

На рисунке 111 приведены примеры условий. Совокупность условий на рисунке 111, *а* означает, что проверка применяется к заявителю для подуслуг 1 и 2 и к представителю заявителя, супруге/супругу для подуслуги 1. Совокупность условий на рисунке 111, *б* означает, что проверка применяется к заявителю для подуслуги 1, если выполняется указанное выражение.

[1] Выпадающий список состоит из субъектов, указанных на вкладке «Основная информация» (см. подраздел 3.3).



(а)



(б)

Рисунок 111 — Примеры условий для проверок

По окончании заполнения формы условия следует нажать кнопку «Добавить».

Для проверки может быть задано любое количество условий.

Ошибочно добавленное условие может быть удалено с помощью кнопки («Удалить»).

По окончании заполнения формы проверки и добавления условий следует нажать кнопку «Создать». После этого проверка будет добавлена в услугу.

Для проверок доступны операции редактирования (кнопка — «Редактировать») и удаления (кнопка — «Удалить»).

На страницах создания и редактирования проверок слева от кнопки «Создать» и «Сохранить» соответственно присутствует переключатель (рисунок 109), позволяющий перейти к режиму редактирования JSON-кода проверки.

3.12. Создание DMN-таблицы проверок

После добавления проверок необходимо сформировать DMN-таблицы проверок для каждой из подуслуг, предусмотренных для услуги.

DMN-таблица будет отображаться специалисту, исполняющему процесс предоставления услуги, в заявлении о предоставлении услуги в зависимости от предоставляемой подуслуги (рисунок 112).

Для того чтобы сгенерировать DMN-таблицы для подуслуг, необходимо:

1. Перейти на вкладку карточки услуги «Процессы».
2. Нажать кнопку «Сгенерировать таблицу» (рисунок 113).
3. В открывшемся окне (рисунок 114):
    б) выбрать код цели (подуслуги) в выпадающем списке «Код цели»;
    в) при необходимости отредактировать присвоенное автоматически наименование DMN-таблицы;
    г) Чек-бокс «Разбить по видам заявителя» устанавливается, если в качестве заявителя могут выступать ФЛ, ИП или ЮЛ;
    д) Прочие параметры изменять не следует;
    е) нажать кнопку «Добавить».
4. Если услуга имеет более одной подуслуги, то повторить действия 1–3 для остальных подуслуг (целей).

После этого объекты «DMN-таблица» будут сгенерированы и добавлены в услугу (рисунок 115).

Важно: Если после генерации DMN-таблиц необходимо произвести модификацию или добавление проверок по услуге, то после корректировки проверок необходимо произвести генерацию DMN-таблиц по всем подуслугам заново.

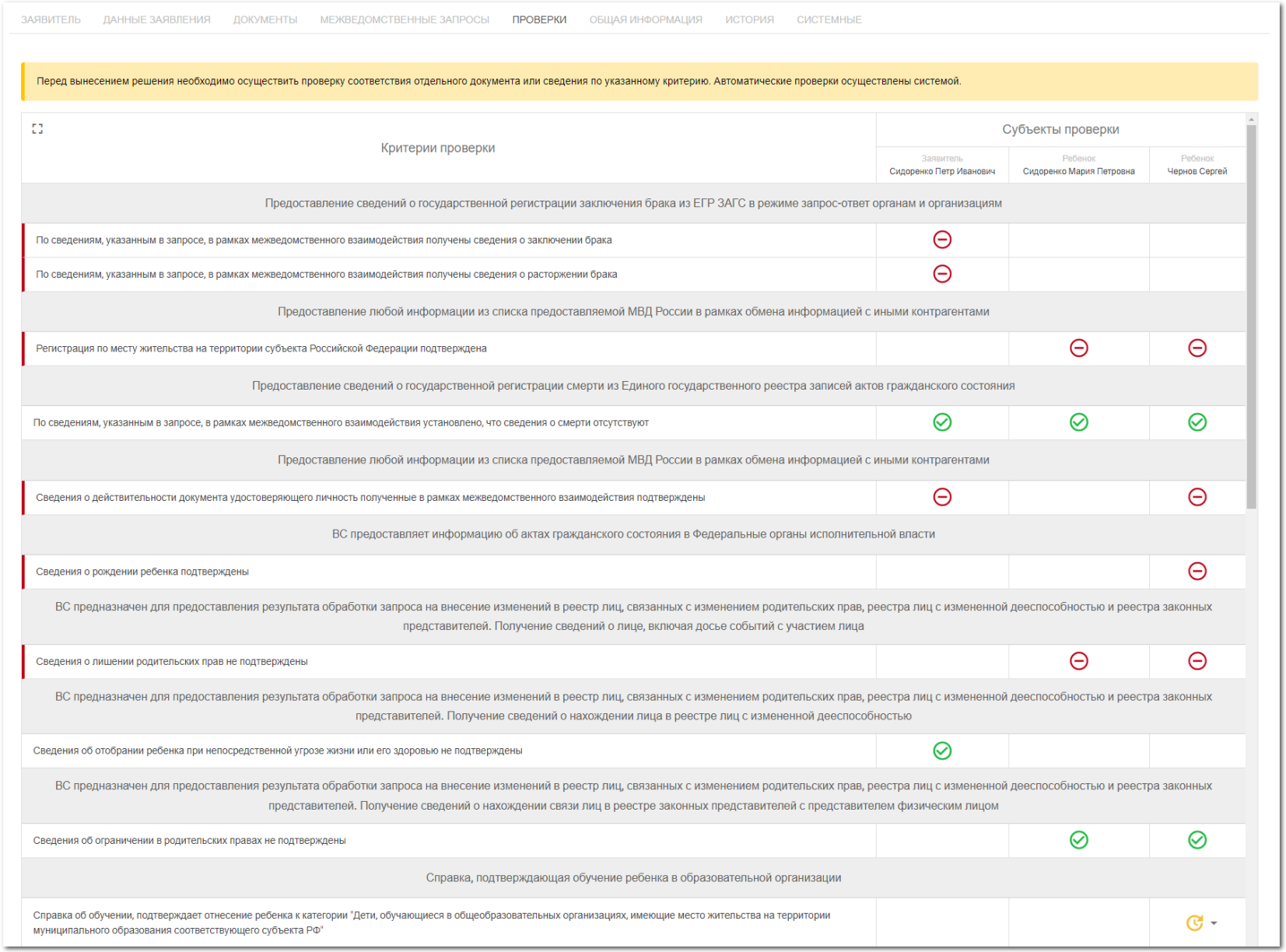


Рисунок 112 — DMN-таблица проверок в карточке заявления о предоставлении услуги (вкладка «Проверки»)

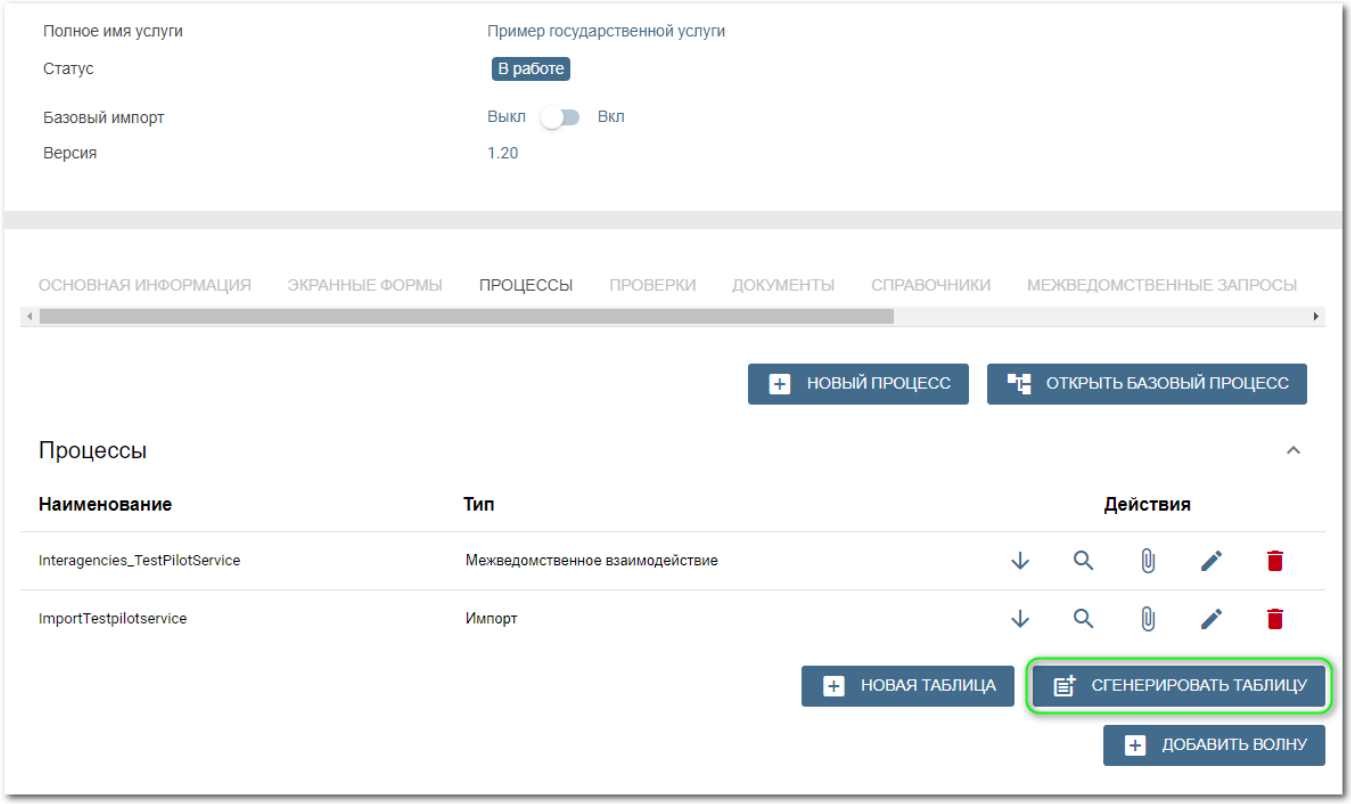


Рисунок 113 — Вкладка карточки услуги «Процессы». Кнопка генерации DMN-таблицы

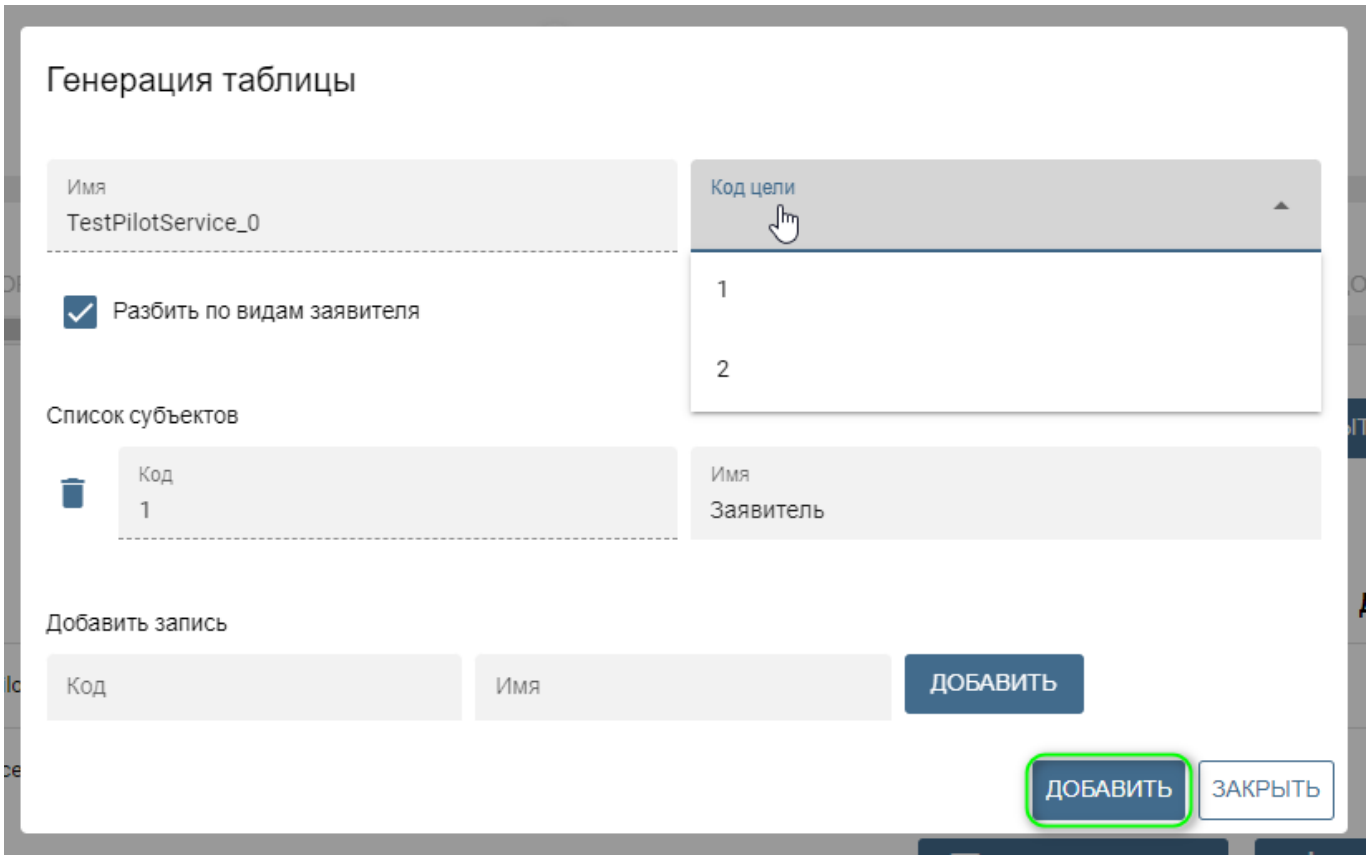


Рисунок 114 — Окно создания DMN-таблицы.

Выпадающий список кодов целей (подуслуг) и кнопка «Добавить»

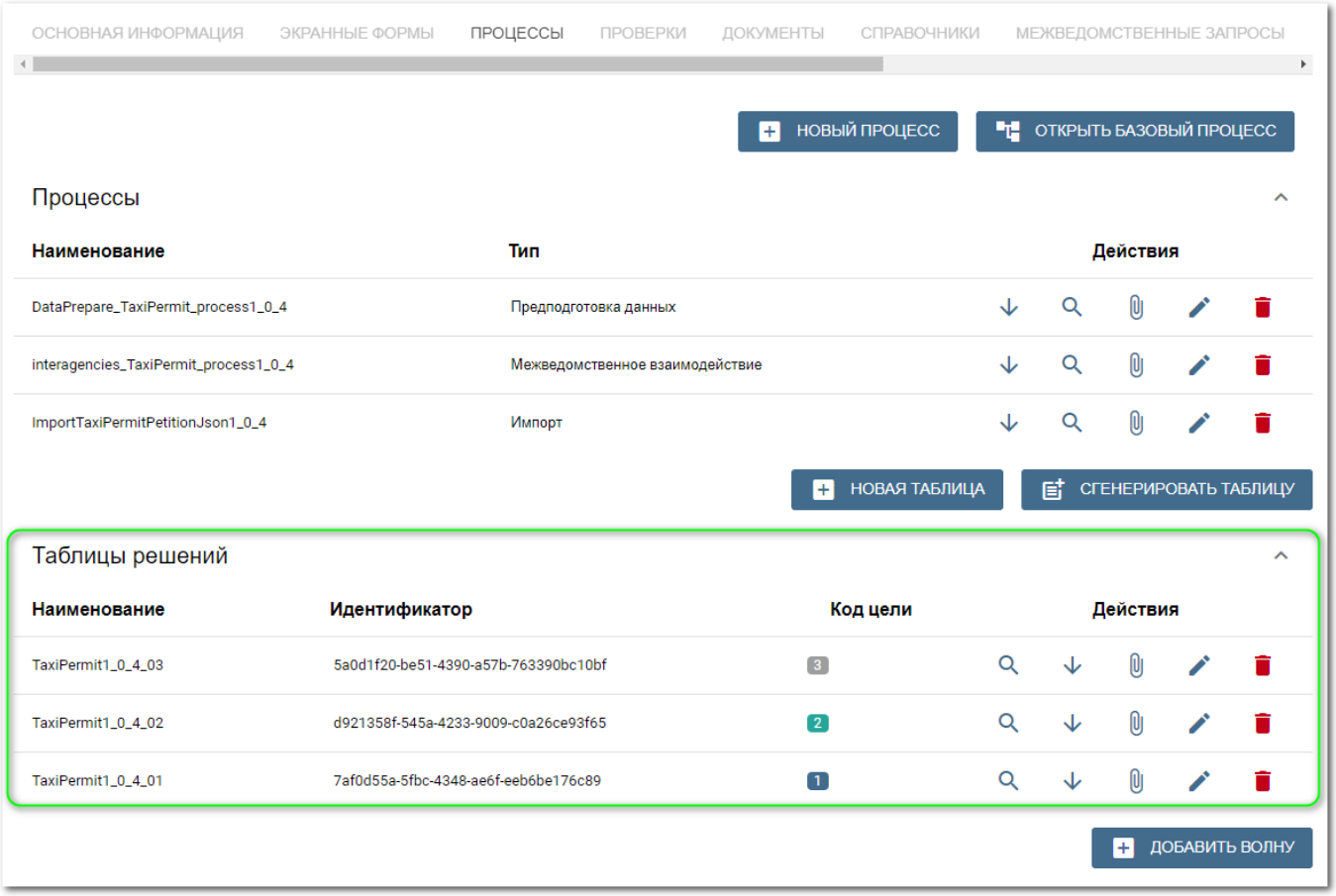


Рисунок 115 — Объекты DMN-таблиц на вкладке карточки услуги «Процессы»

Любую из сгенерированных DMN-таблиц можно при необходимости скачать на компьютер в виде файла с помощью кнопки ↓ («Скачать») и отредактировать в Camunda Modeler (рисунок 116). Отредактированную DMN-таблицу можно загрузить в услугу с помощью кнопки «Новая таблица» (рисунок 117). В окне загрузки файла DMN-таблицы (рисунок 118) необходимо:

– указать наименование DMN-таблицы, отличное от исходного;

– выбрать цель (подуслугу), к которой относится DMN-таблица;

– приложить файл DMN-таблицы: нажать на поле «Загрузить», найти и выбрать файл в открывшемся окне;

– нажать кнопку «Создать».

Кнопки 🗑 («Удалить») позволяют удалять соответствующие DMN-таблицы из услуги.

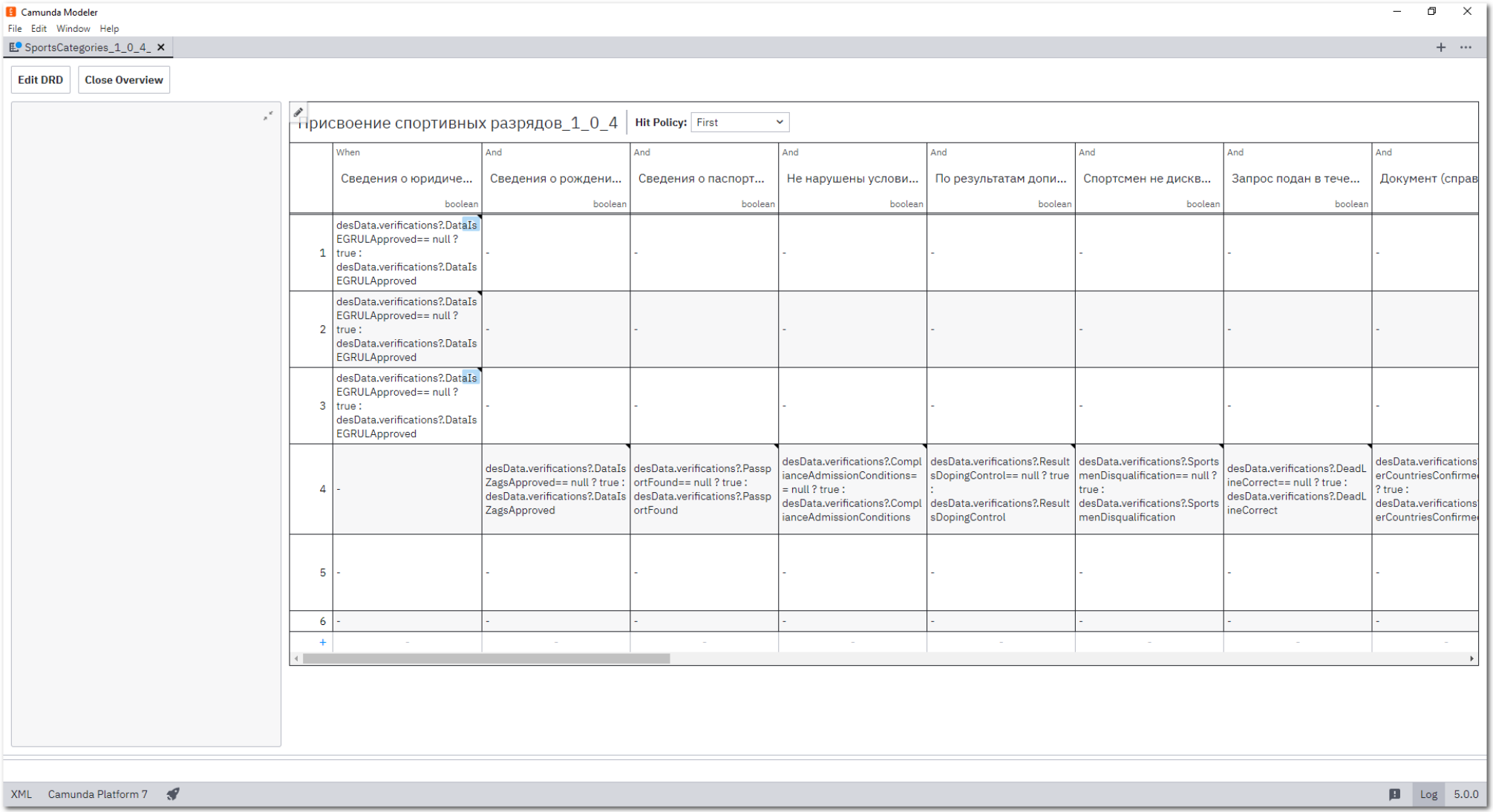


Рисунок 116 — Файл DMN-таблицы, открытый в Camunda Modeler

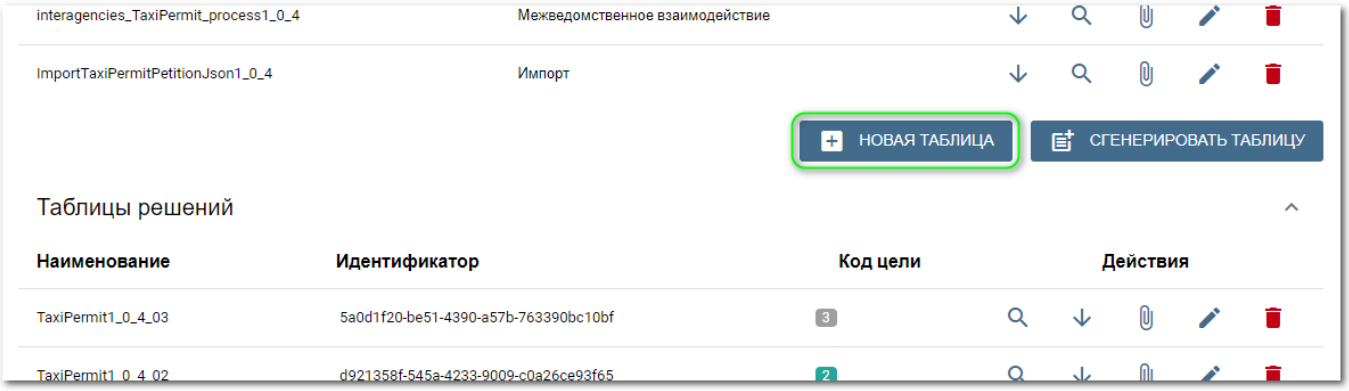


Рисунок 117 — Вкладка карточки услуги «Процессы».

Кнопка загрузки файла DMN-таблицы в услугу

Рисунок 118 — Окно загрузки файла DMN-таблицы в услугу

3.13. Создание процесса подготовки данных

Пример схемы процесса подготовки данных представлена на рисунке 119.

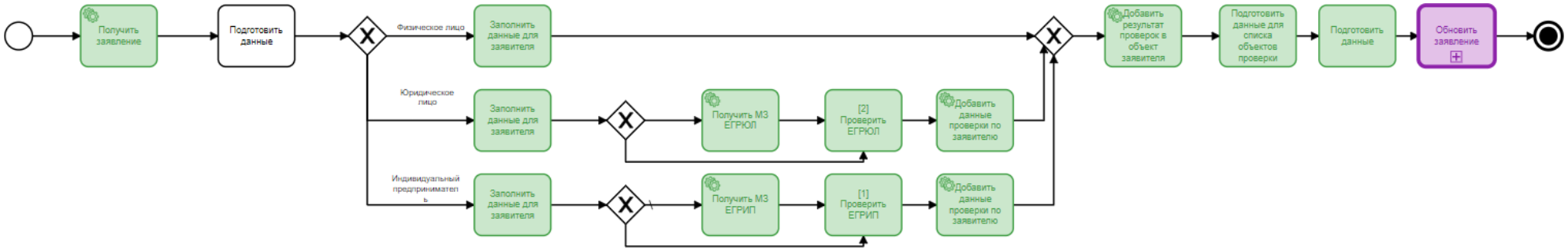


Рисунок 119 — BPMN-схема процесса подготовки данных

Первая задача — «Получить заявление» — отвечает за получение информации о заявлении и аналогична одноименной задаче в процессе МВ (см. подраздел 3.10.2).

Во второй задаче — «Подготовить данные» — для каждого субъекта услуги определяется отдельная переменная, которая будет содержать список проверок для данного субъекта (рисунок 120). Значения этих переменных задаются в последующих задачах.

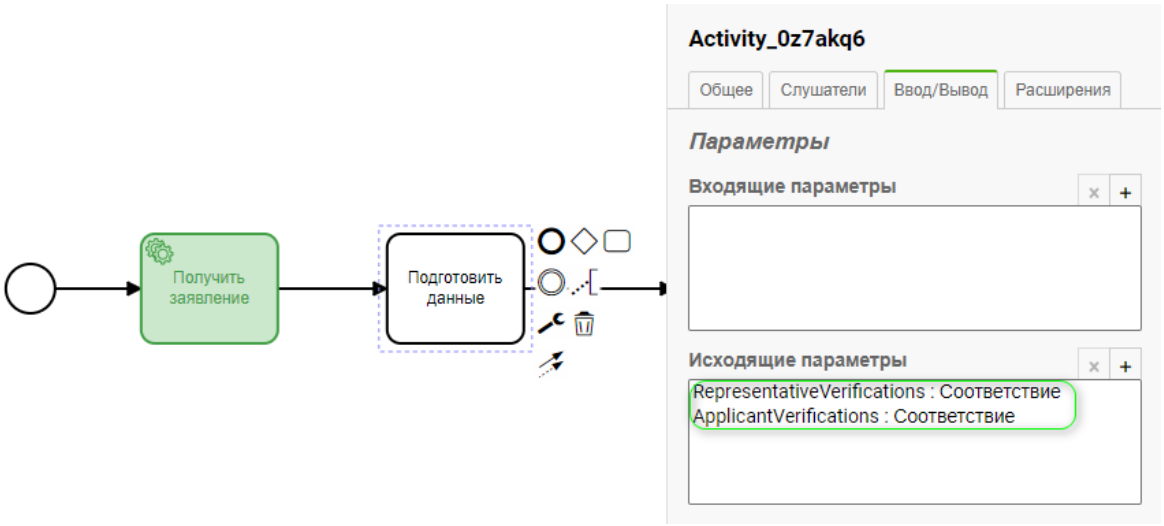


Рисунок 120 — Параметры задачи «Подготовить данные» в ППД.
Вкладка Input/Output («Ввод/Вывод»). Переменные для списков проверок для субъектов услуги

В зависимости от типа субъекта, который является заявителем, в ППД осуществляется сборка данных о субъекте в одну переменную и присвоение ей уникального идентификатора (рисунок 121).

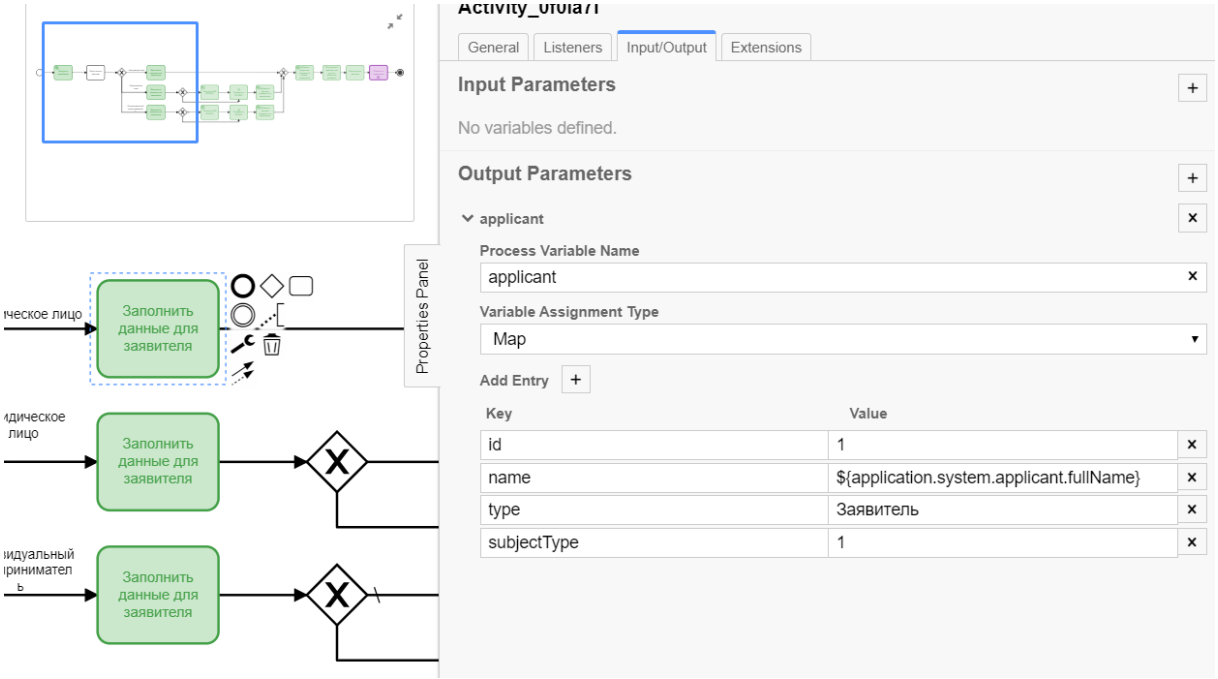


Рисунок 121 — Параметры задачи «Заполнить данные для заявителя» в ППД.
Вкладка Input/Output («Ввод/Вывод»)

Переменные на вкладке Input/Output («Ввод/Вывод»):

– id — уникальный идентификатор субъекта;

– name — поле, отображаемое в таблице решений;

– type — тип субъекта, отображаемый в таблице решений;

– subjectType — код субъекта со вкладки «Основная информация» → «Список субъектов».

Любая проверка состоит минимум из 3 задач. Совокупность задач, определяющих проверку, формируется в ППД для каждой из проверок (рисунок 122).

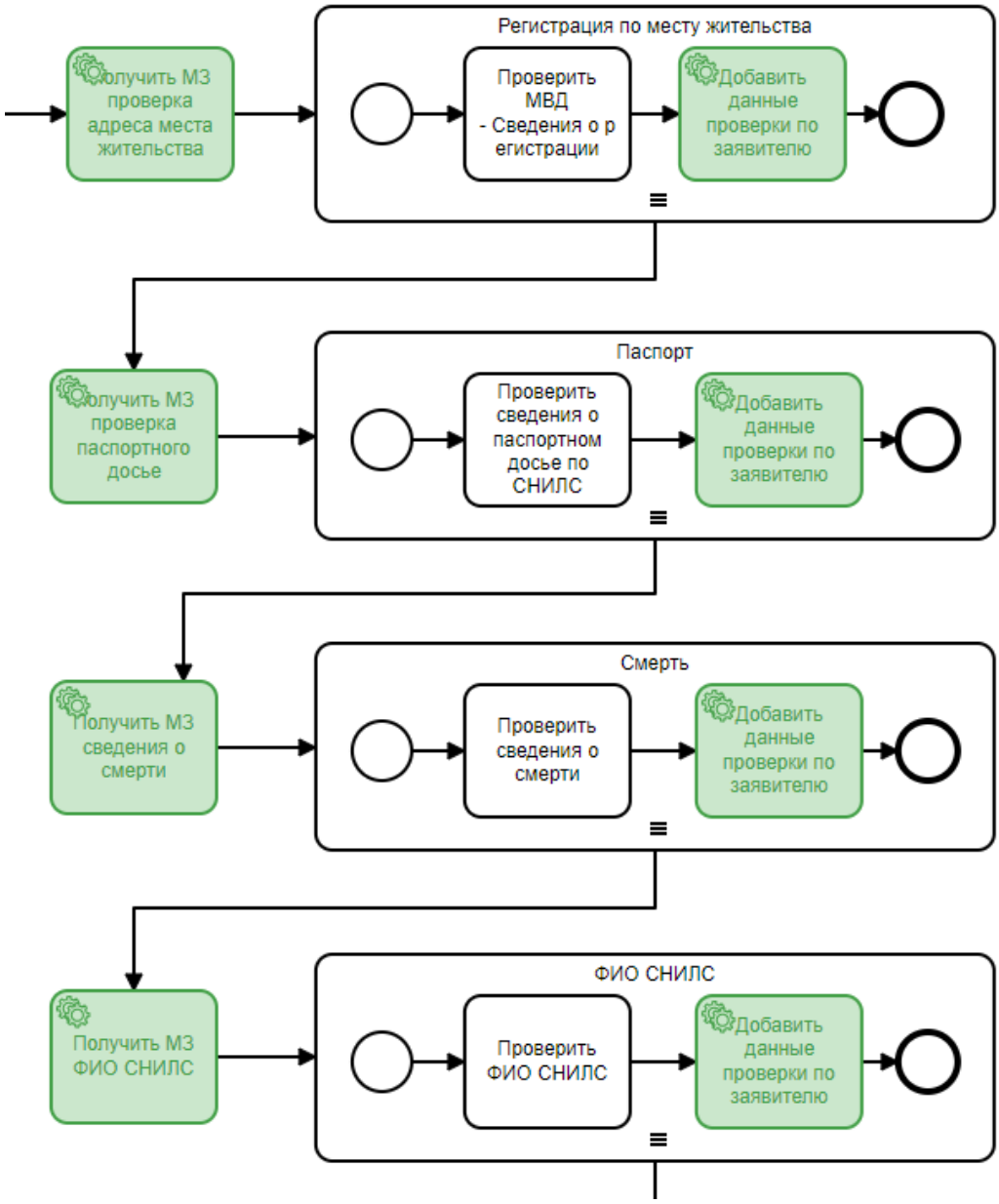


Рисунок 122 — Совокупности задач,
определяющих четыре автоматические проверки, основанные на МЗ

Далее задачи, определяющие проверку, рассматриваются на примере, приведенном на рисунке 123.

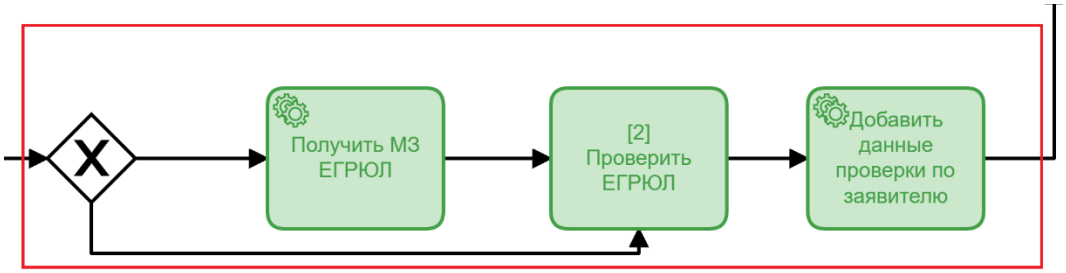


Рисунок 123 — Задачи, определяющие автоматическую проверку, основанную на МЗ

Первая из задач нужна для получения сведений из ответа на МЗ. Данная задача исполняется аналогично задаче «Фильтр» в процессе МВ за исключением параметров, подаваемых на выход. В данном случае коллекцией для фильтрации является interDepReqs, переменной, по которой осуществляется поиск необходимого МЗ, — name, а искомой переменной — egrul-inn (рисунок 124).

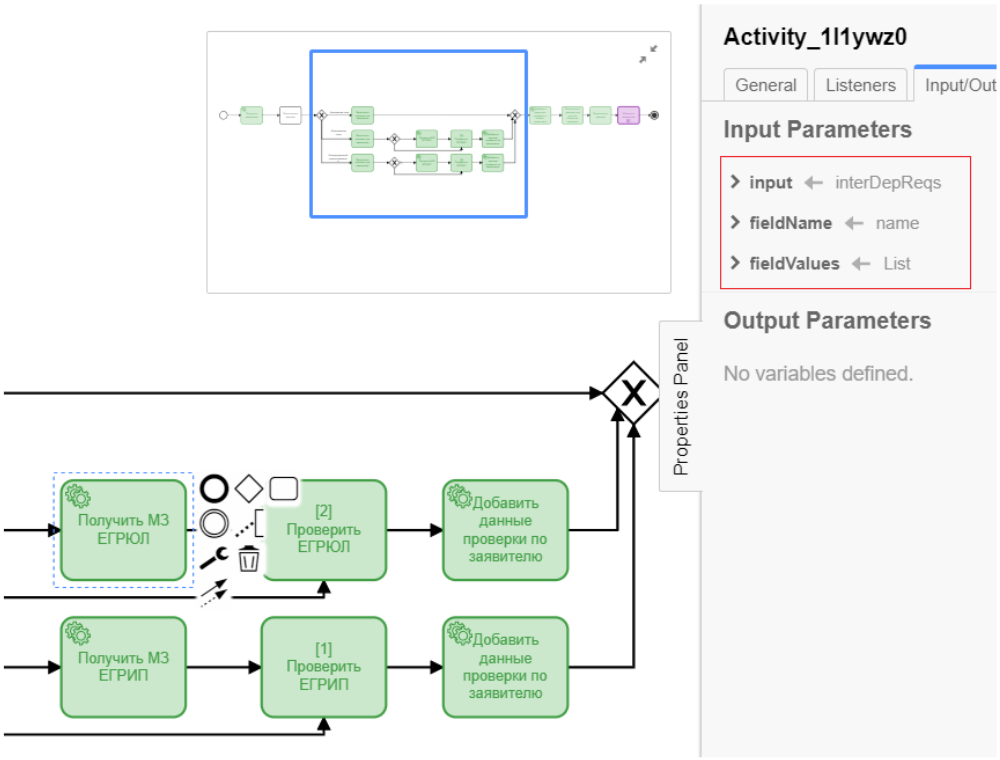


Рисунок 124 — Параметры задачи получения сведений из ответа на МЗ.
Вкладка Input/Output («Ввод/Вывод»)

После получения сведений из ответа на МЗ и записи их в новую переменную стартует задача по проверке сведений (рисунок 125).

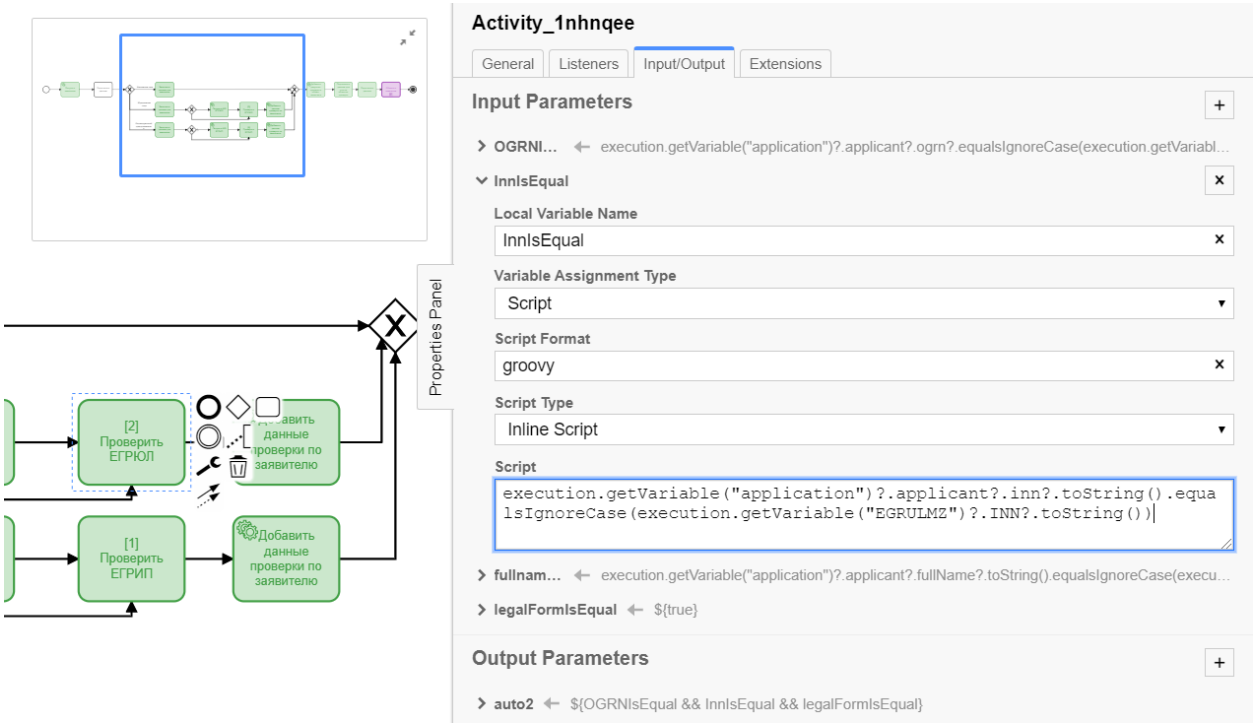


Рисунок 125 — Параметры задачи проверки сведений, полученных из ответа на МЗ. Вкладка Input/Output («Ввод/Вывод»)

В данном случае используется внутренняя переменная задачи для написания скрипта проверки сведений.

Пример скрипта для конкретной услуги:

```
execution.getVariable("application")?.applicant?.inn?.toString().equalsIgnore
Case(execution.getVariable("EGRULMZ")?.INN?.toString())
```

Здесь:

– *execution.getVariable("application")?.applicant?.inn?.toString()* — путь к переменной из заявления, которая содержит в себе ИНН и преобразование полученной переменной в строковую;

– ().equalsIgnoreCase — функция сравнения;

– *(execution.getVariable("EGRULMZ")?.INN?.toString())* — путь к переменной из полученного ответа на МЗ и преобразование в строку.

В данной задаче указаны четыре внутренние переменные, на основании которых происходит определение результирующего критерия (исходящий параметр).

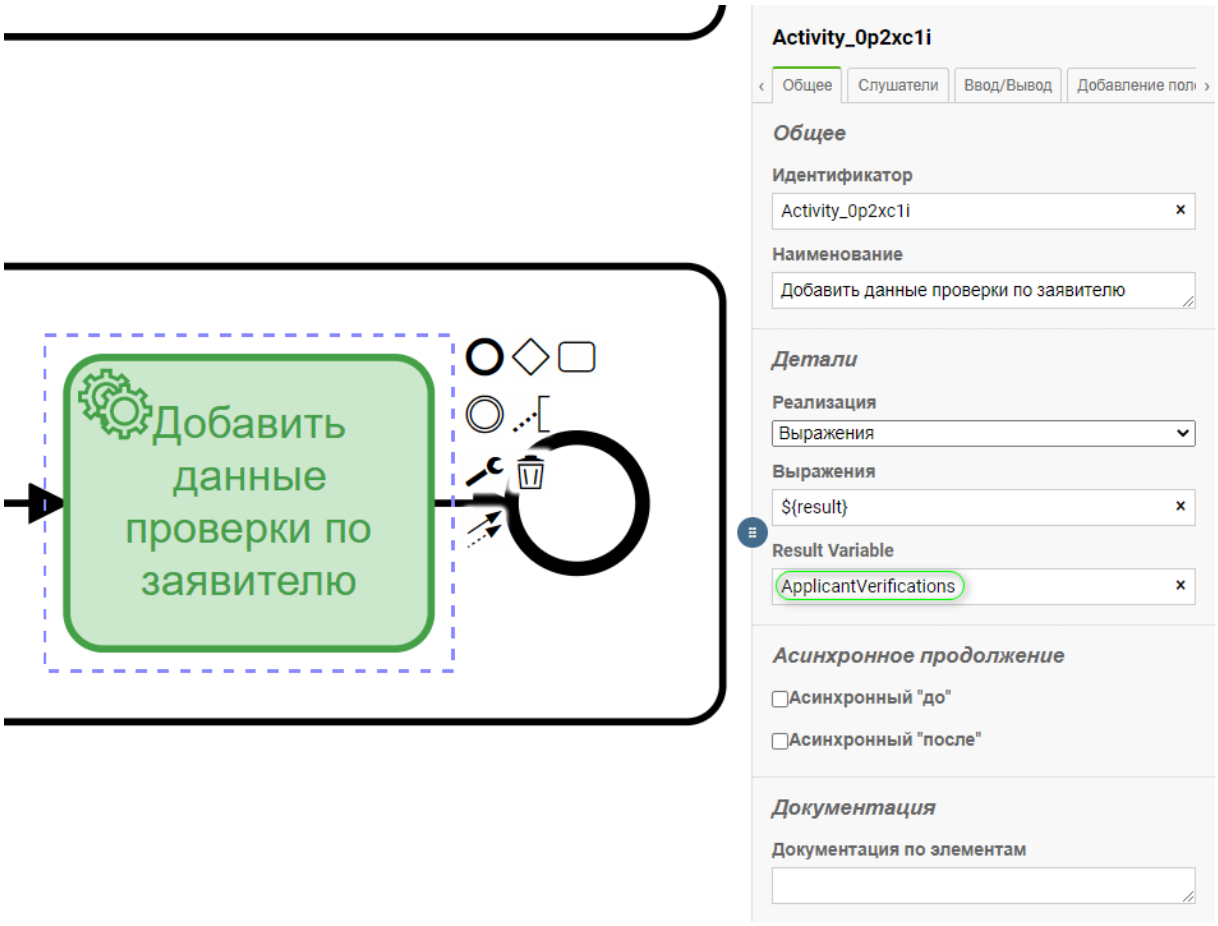
Таким образом, если в услуге (на вкладке «Проверки», см. подраздел 3.11) была создана автоматическая проверка auto2 (название может быть иным), то после исполнения всех скриптов во внутренних параметрах происходит передача результирующих значений в исходящую переменную auto2. В данном случае все переменные, полученные в результате, для прохождения данной проверки должны иметь значение true («истина», «да»). Тем самым по булевой логике auto2 == true.

Также пример скрипта:

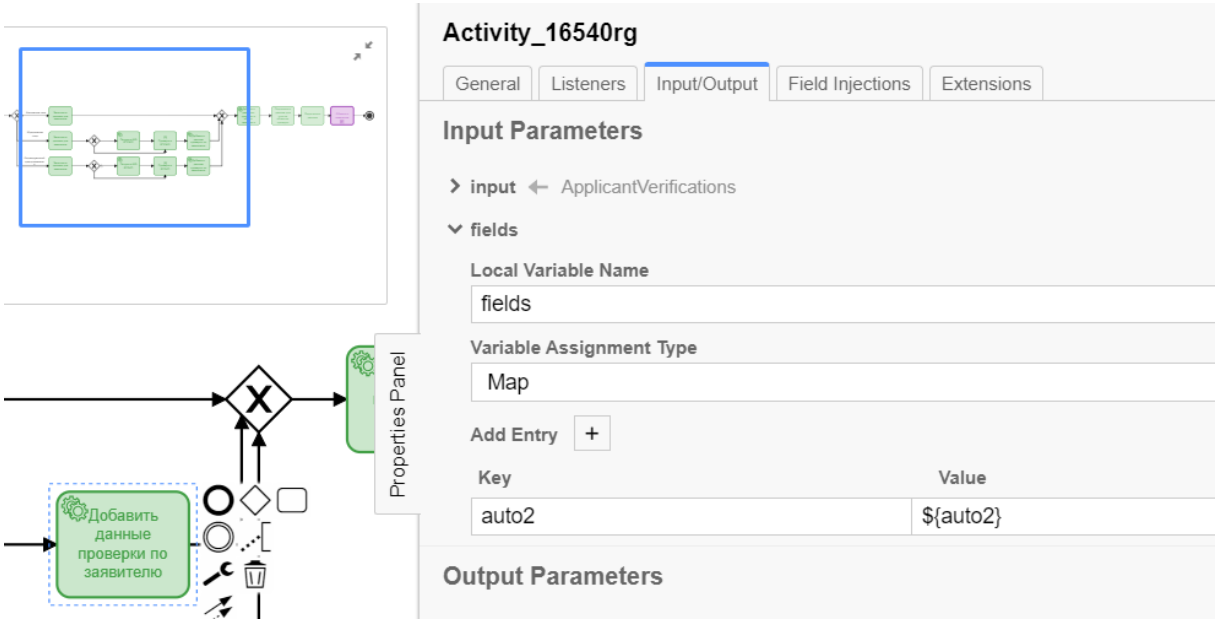
```
def                                applicantSnils                                =
application?.applicant?.snils?.toString()?.replaceAll("[^0-9]", "");

for(responseDoc in execution.getVariable("ZagsDeath"))
{
if(responseDoc?.response == null){continue;}

if(responseDoc?.request?.reqSvedZapros?.reqSvedFlSnils?.toString()?.replaceAl
l("[^0-9]", "")?.equalsIgnoreCase(applicantSnils)){

if(responseDoc?.response?.svedOtverAGS?.getAt(0)?.rezObrab    !=    null    &&
responseDoc?.response?.svedOtverAGS?.getAt(0)?.rezObrab.toString() != "1"){
            return true;
        }
    }
}
return false
```

Данный скрипт проверяет, что заявитель не числится умершим.

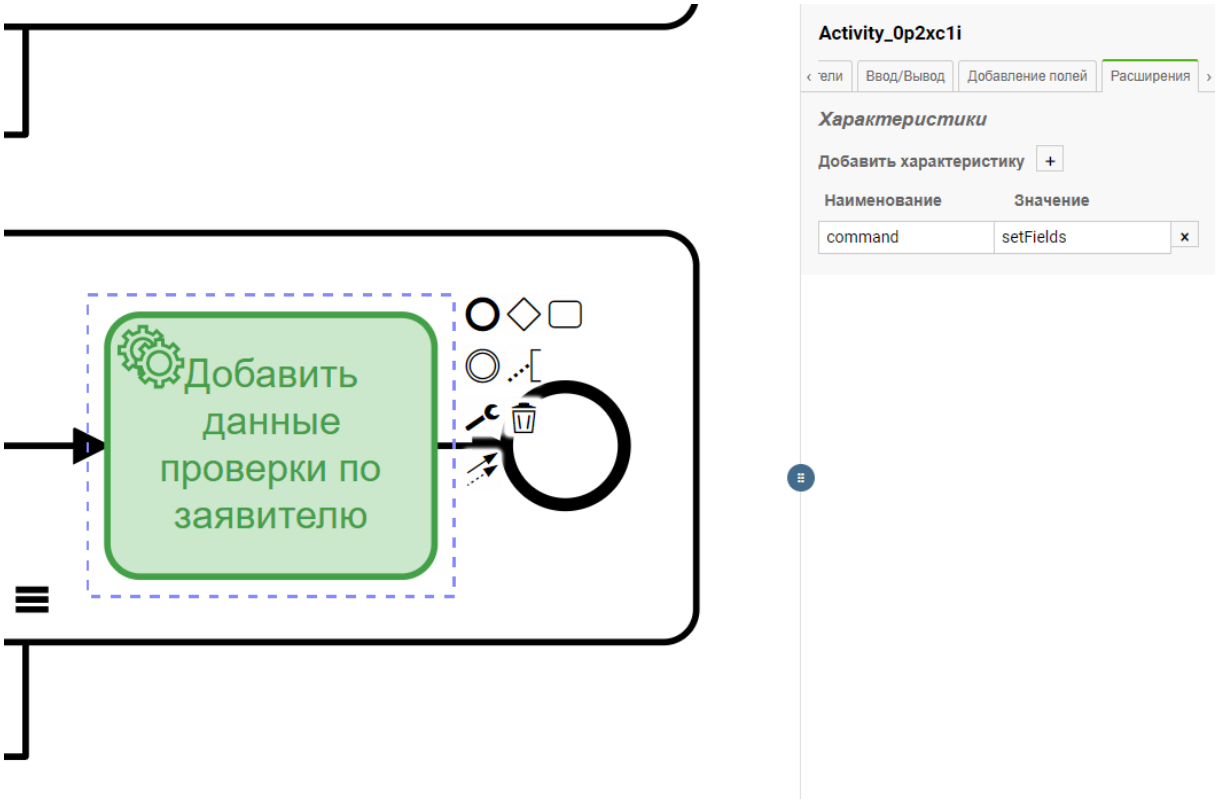
Результирующую переменную auto2 необходимо добавить в общий список проверок для субъекта, который был создан в задаче «Подготовить данные». Это осуществляется в задаче, приведенной на рисунке 126, с помощью команды setFields.



(а)



(б)



(в)

Рисунок 126 — Параметры задачи добавления данных проверки в общий список для субъекта. а) Вкладка «Общее» (General). б) Вкладка Input/Output («Ввод/Вывод»). в) Вкладка «Расширения» (Extensions)

Далее следует блок стандартных задач, завершающих процесс (рисунок 127).

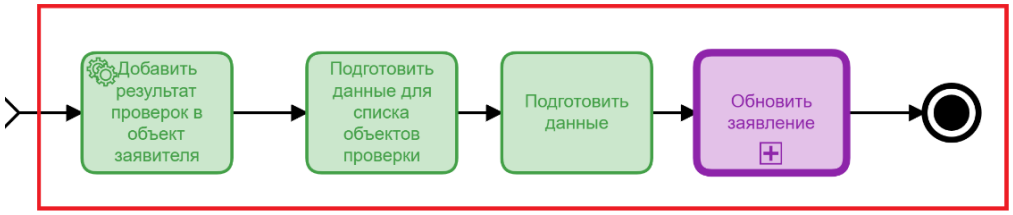


Рисунок 127 — Завершающие задачи ППД

Задача «Добавить результат проверок…» с помощью команды setFields добавляет результат проверок в переменную verifications для субъекта (рисунок 128).

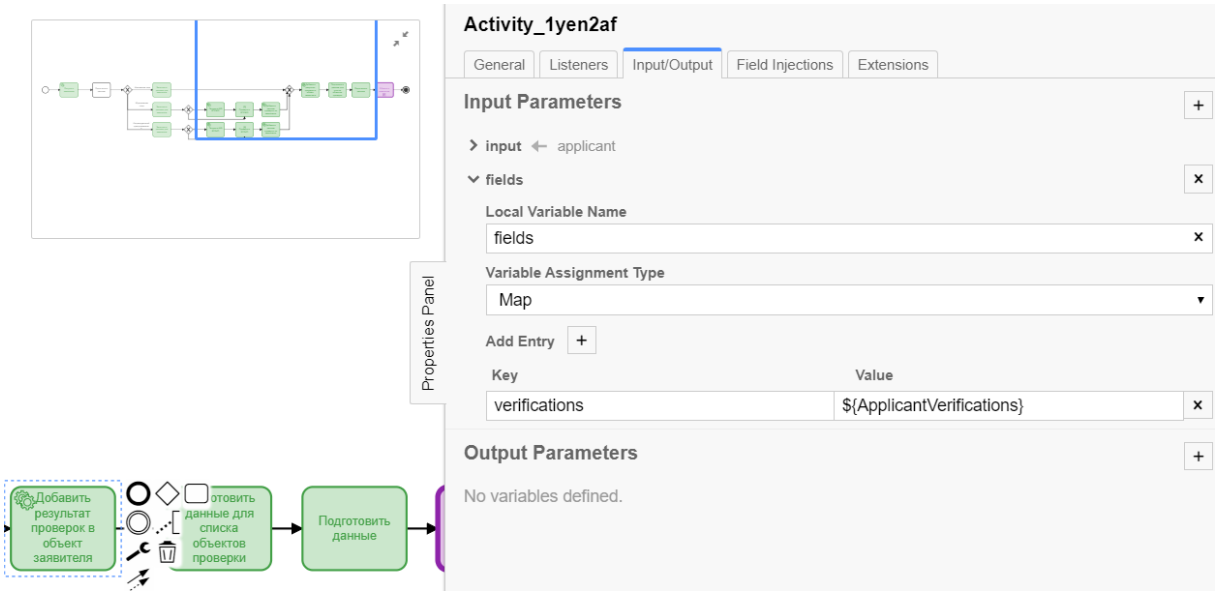


Рисунок 128 — Параметры задачи добавления результата проверок в переменную субъекта. Вкладка Input/Output («Ввод/Вывод»)

Задачи «Подготовить данные…» добавляют переменную applicant со всеми проверками в новую переменную для передачи в подпроцессы (рисунки 129 и 130).

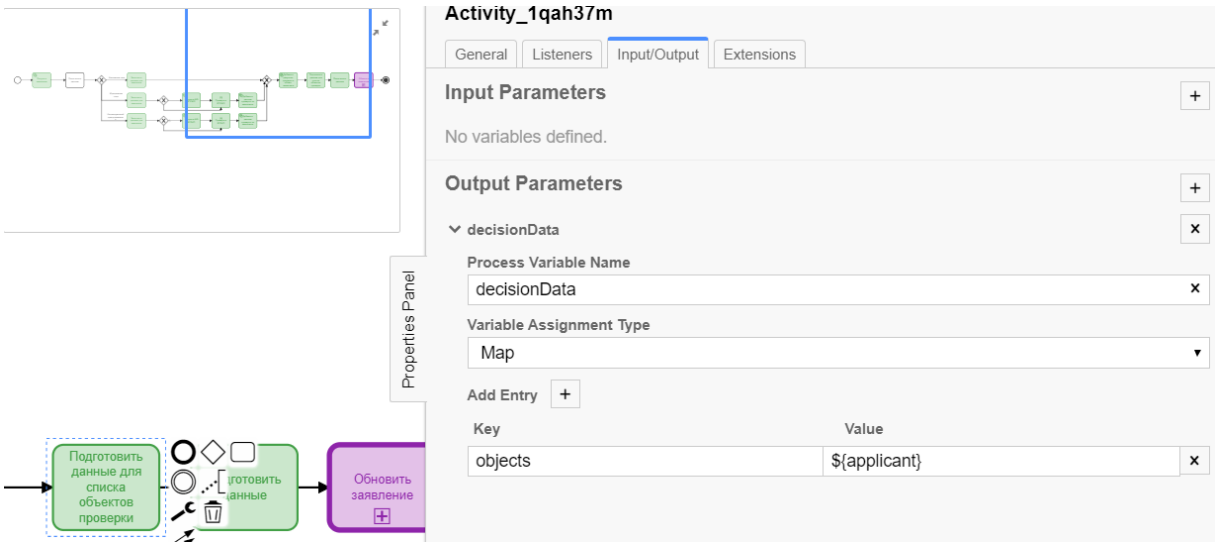


Рисунок 129 — Параметры задачи подготовки данных для списка субъектов проверок. Вкладка Input/Output («Ввод/Вывод»)

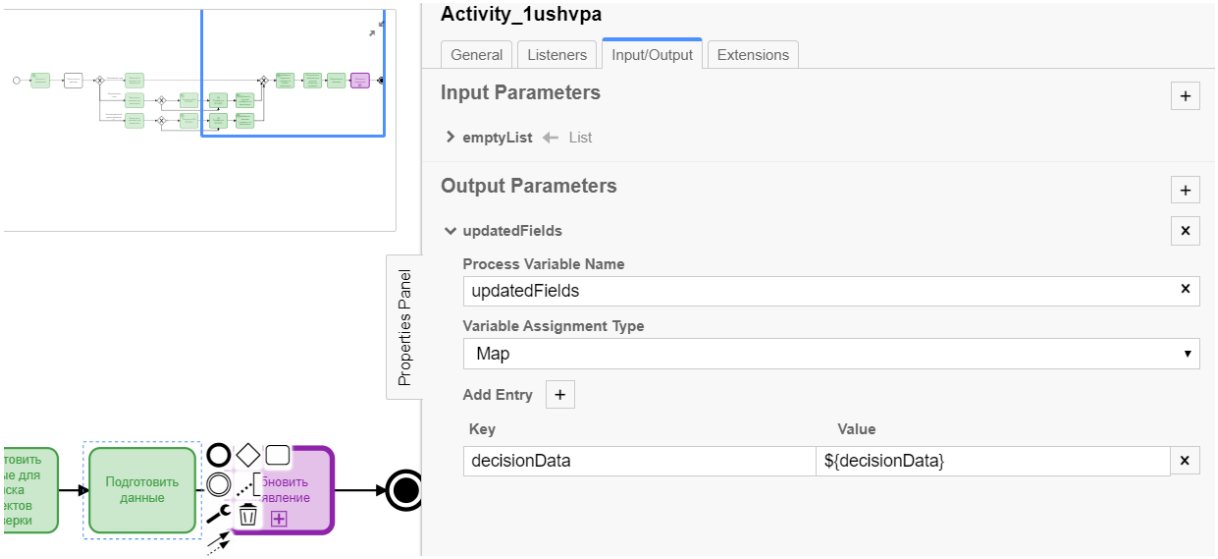


Рисунок 130 — Параметры задачи подготовки данных. Вкладка Input/Output («Ввод/Вывод»)

Последняя задача (рисунок 131) осуществляет отправку данных в заявление через подпроцесс для генерации таблицы решений (DMN).

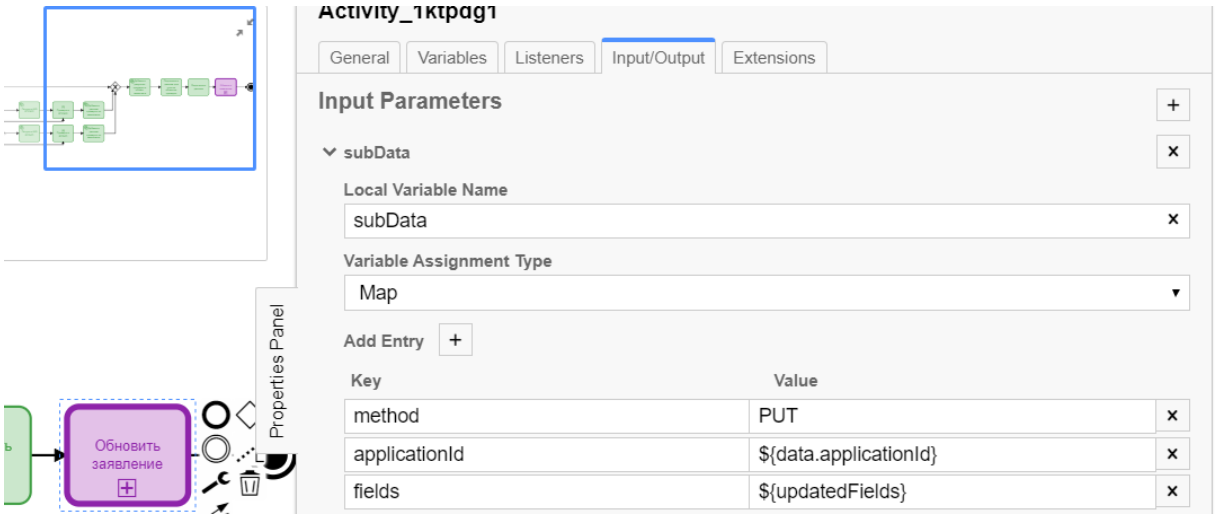


Рисунок 131 — Параметры задачи обновления заявления. Вкладка Input/Output («Ввод/Вывод»)

3.14. Настройка экранных форм

3.14.1. Общие сведения об экранных формах. Операции с экранными формами

Экранные формы настраиваются на вкладке карточки услуги «Экранные формы».

Экранные формы отображают данные по услуге в понятном для сотрудника ведомства (организации) виде. Например, экранная форма заявления отображает все уникальные данные из XML-файла заявления. То есть, все данные, которые при импорте были занесены в ПГС, будут отображаться в карточке заявления на вкладке «Данные заявления» (рисунок 132). Таким образом, в экранной форме заявления должны быть описаны все атрибуты из XSD-схемы.

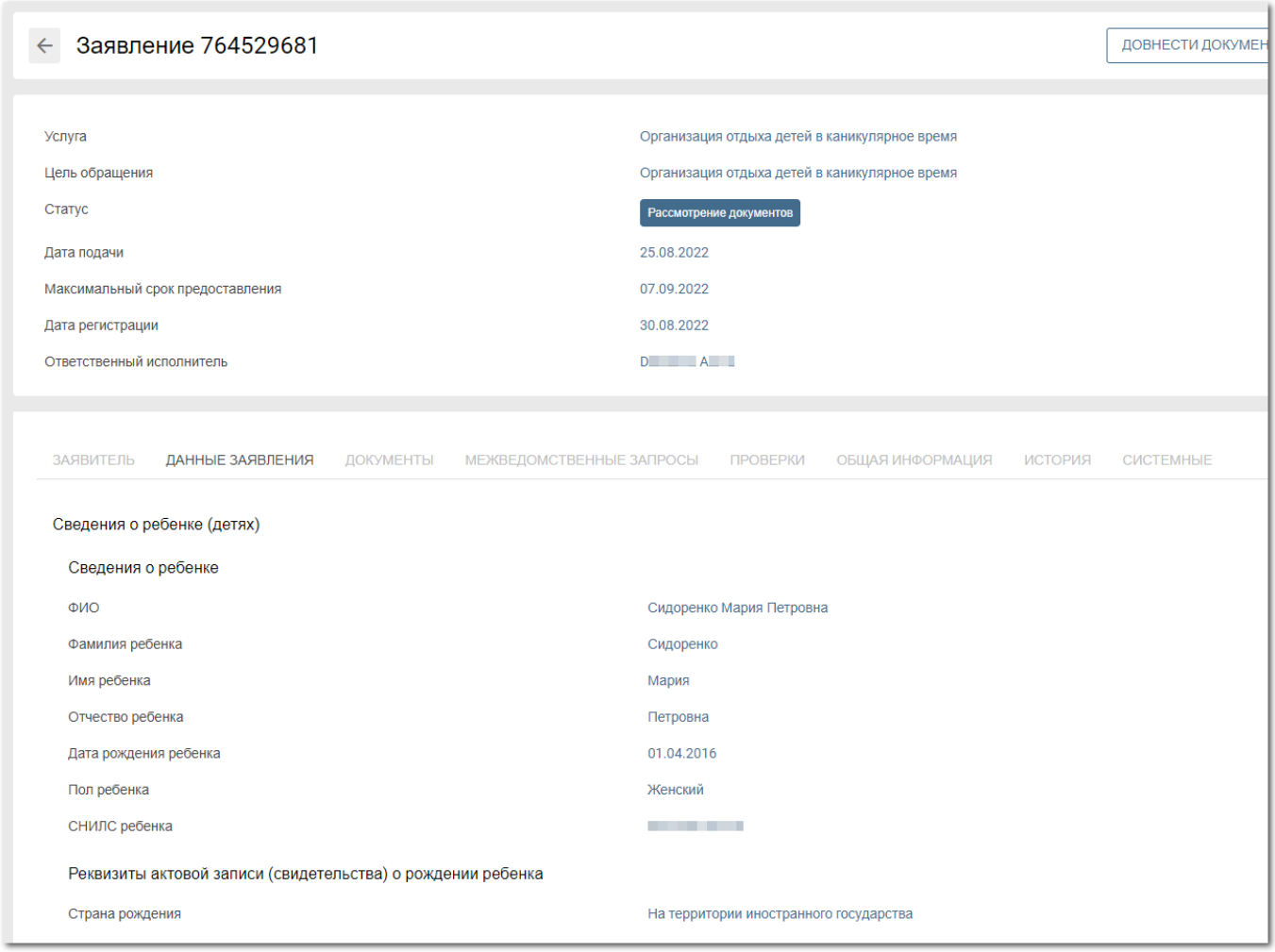


Рисунок 132 — Карточка заявления о предоставлении услуги. Данные заявления

Экранные формы могут как добавляться в услугу вручную, так и создаваться автоматически при получении процесса импорта (см. подраздел 3.5) и добавлении в услугу документов (см. подразделы 3.8 и 3.15) и МЗ (см. подраздел 3.10).

В большинстве случаев автоматически создается только объект «Экранная форма», а сама экранная форма при этом не настроена. В некоторых случаях может создаваться преднастроенная экранная форма, которая требует корректировки и (или) донастройки.

На рисунке 133 приведены примеры преднастроенной экранной формы (в группе «Данные заявления») и объектов «Экранная форма», для которых сами экранные формы не созданы (в группах «Заявитель» и «Очный прием»). Примеры сформированных экранных форм приведены также на рисунке 134.

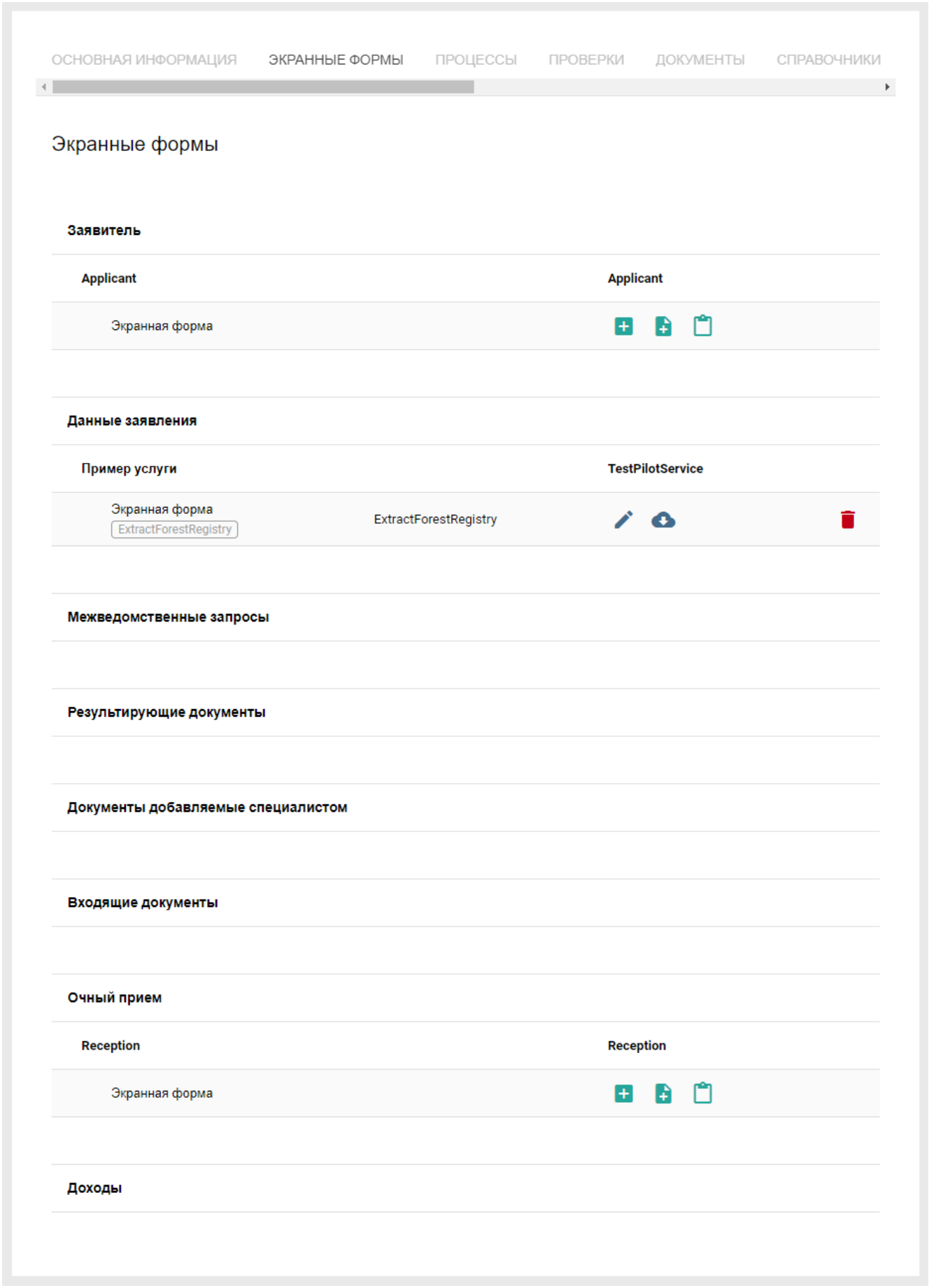


Рисунок 133 — Объекты «Экранная форма» и сформированная экранная форма на вкладке карточки услуги «Экранные формы»

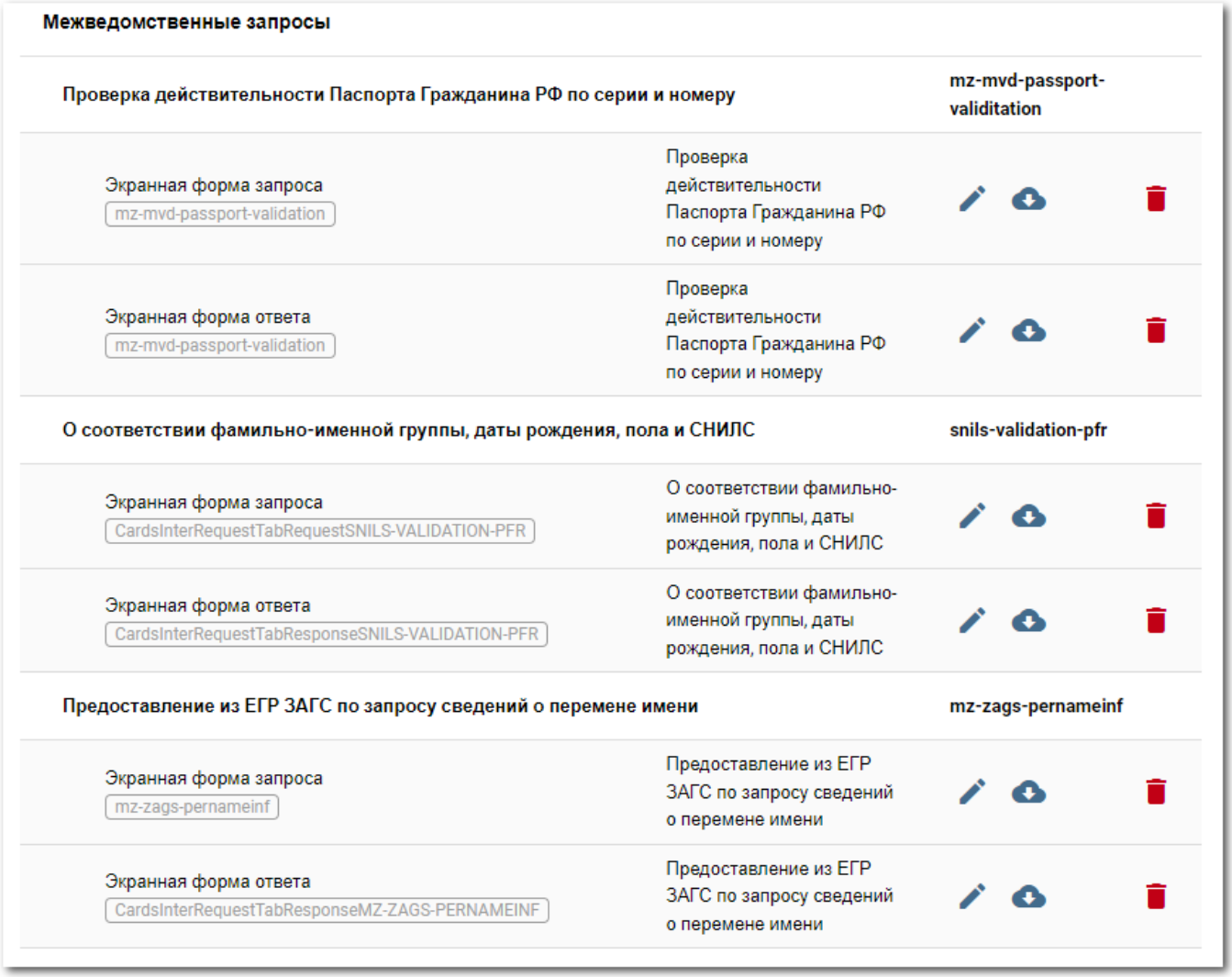


Рисунок 134 — Примеры сформированных экранных форм
на вкладке карточки услуги «Экранные формы»

Для объектов «Экранная форма» без созданных экранных форм доступны операции (рисунок 133):

– «Создать пустую форму» — кнопка ;

– «Создать из файла» — кнопка ;

– «Создать из JSON» — кнопка .

После создания экранной формы для нее доступны операции (рисунок 133):

– «Редактировать форму» — кнопка ;

– «Скачать форму» — кнопка ;

– «Удалить» — кнопка .

Предупреждение. Удаление экранной формы происходит без запроса подтверждения удаления. Следует соблюдать осторожность при выполнении данной операции.

Настройка любой экранной формы осуществляется посредством редактирования ее кода в формате JSON.

При нажатии на кнопку создается и открывается для редактирования пустая экранная форма (рисунок 135). При этом на странице редактирования экранной формы присутствует код, добавляемый автоматически, задающий основные параметры экранной формы. Под полем, в котором осуществляется редактирование JSON-кода, присутствует раздел «Расширенные настройки», представляющий собой простейший конструктор экранной формы, позволяющий автоматически добавлять в нее определенные компоненты. Тем не менее экранная форма требует донастройки специалистом вручную.

Для сохранения кода экранной формы в услуге следует использовать кнопку «Сохранить схему».

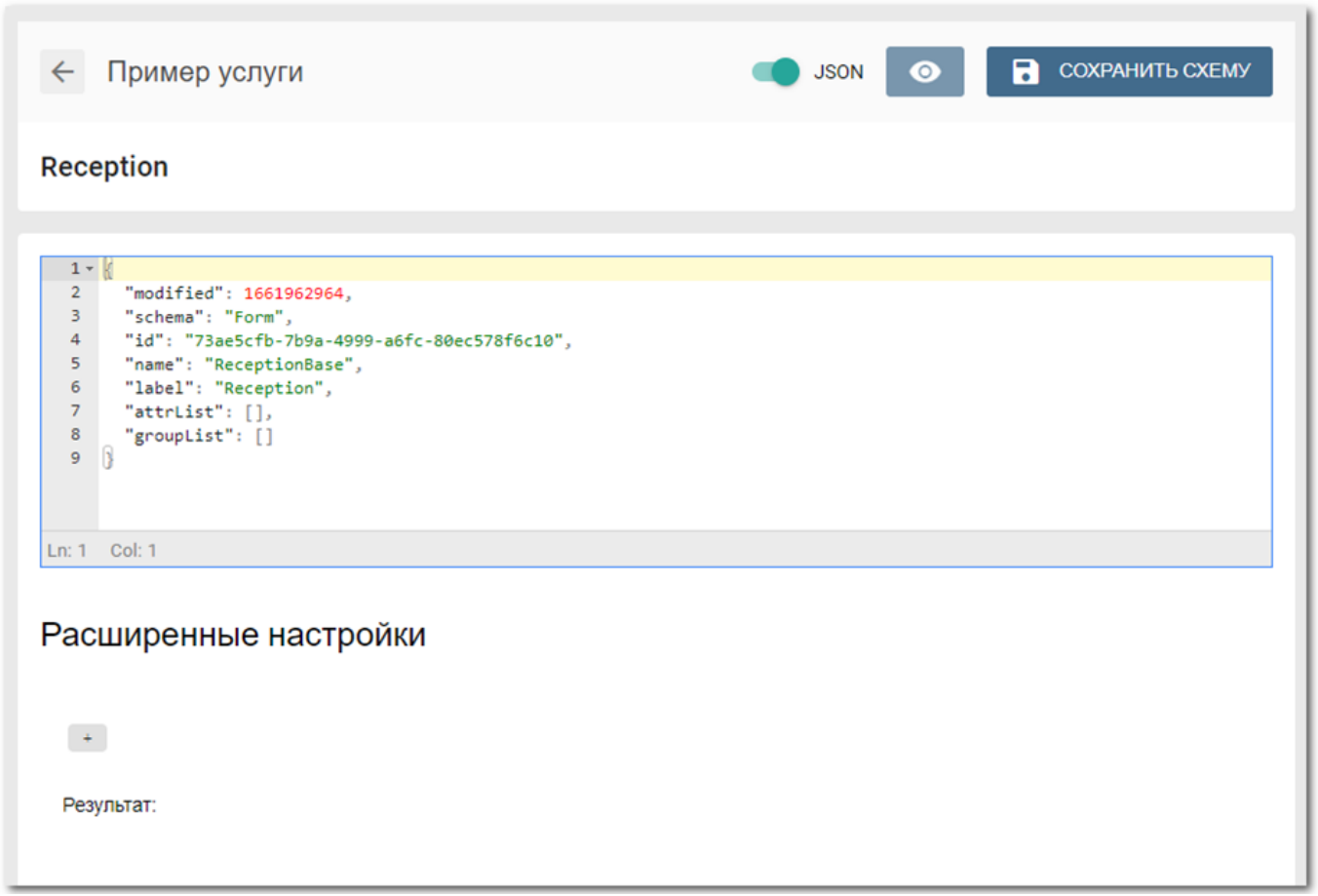


Рисунок 135 — Страница ручного создания экранной формы.
JSON-код, добавляемый в пустую экранную форму по умолчанию

При нажатии на кнопку открывается окно загрузки файла в ПГС. Операция позволяет создать экранную форму путем загрузки предварительно подготовленного файла, содержащего JSON-код экранной формы.

При нажатии на кнопку открывается окно, представленное на рисунке 136. Операция позволяет вставить заранее подготовленный JSON-код из буфера обмена и отредактировать вручную. Экранная форма создается при нажатии на кнопку «Создать».

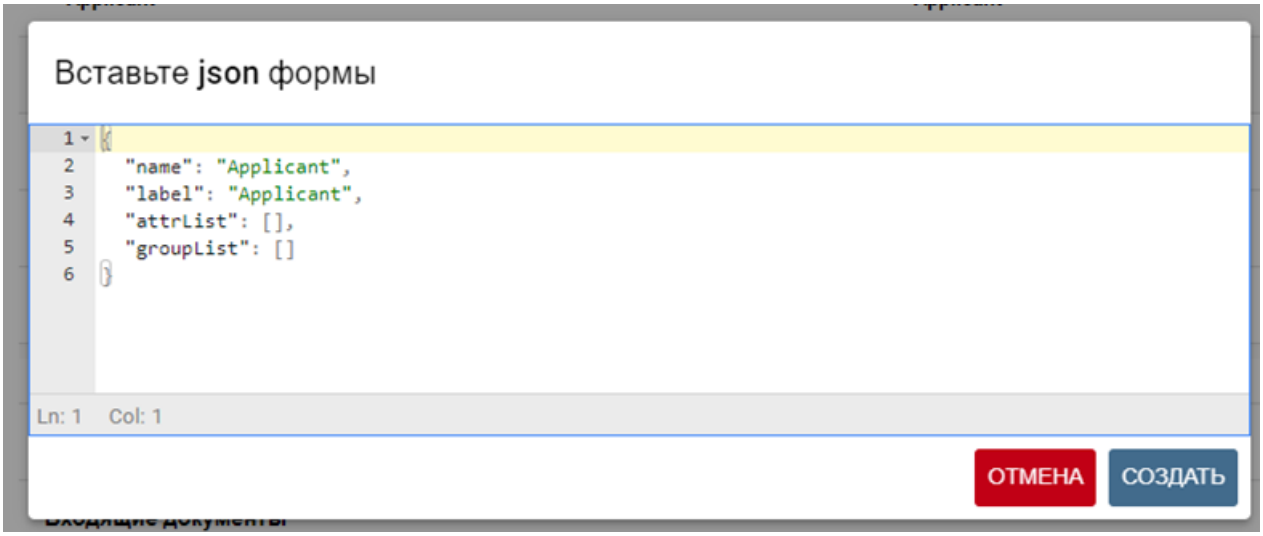


Рисунок 136 — Окно создания экранной формы путем вставки ее JSON-кода из буфера обмена

При нажатии на кнопку открывается страница редактирования экранной формы, аналогичная странице создания (ср. рисунки 135 и 137).

При переводе переключателя «JSON» в левое положение экранная форма переключается в режим визуального конструктора (рисунок 138).

Кнопка («Предпросмотр») позволяет открыть сформированную экранную форму для предварительного просмотра.

Для сохранения кода экранной формы в услуге следует использовать кнопку «Сохранить схему».

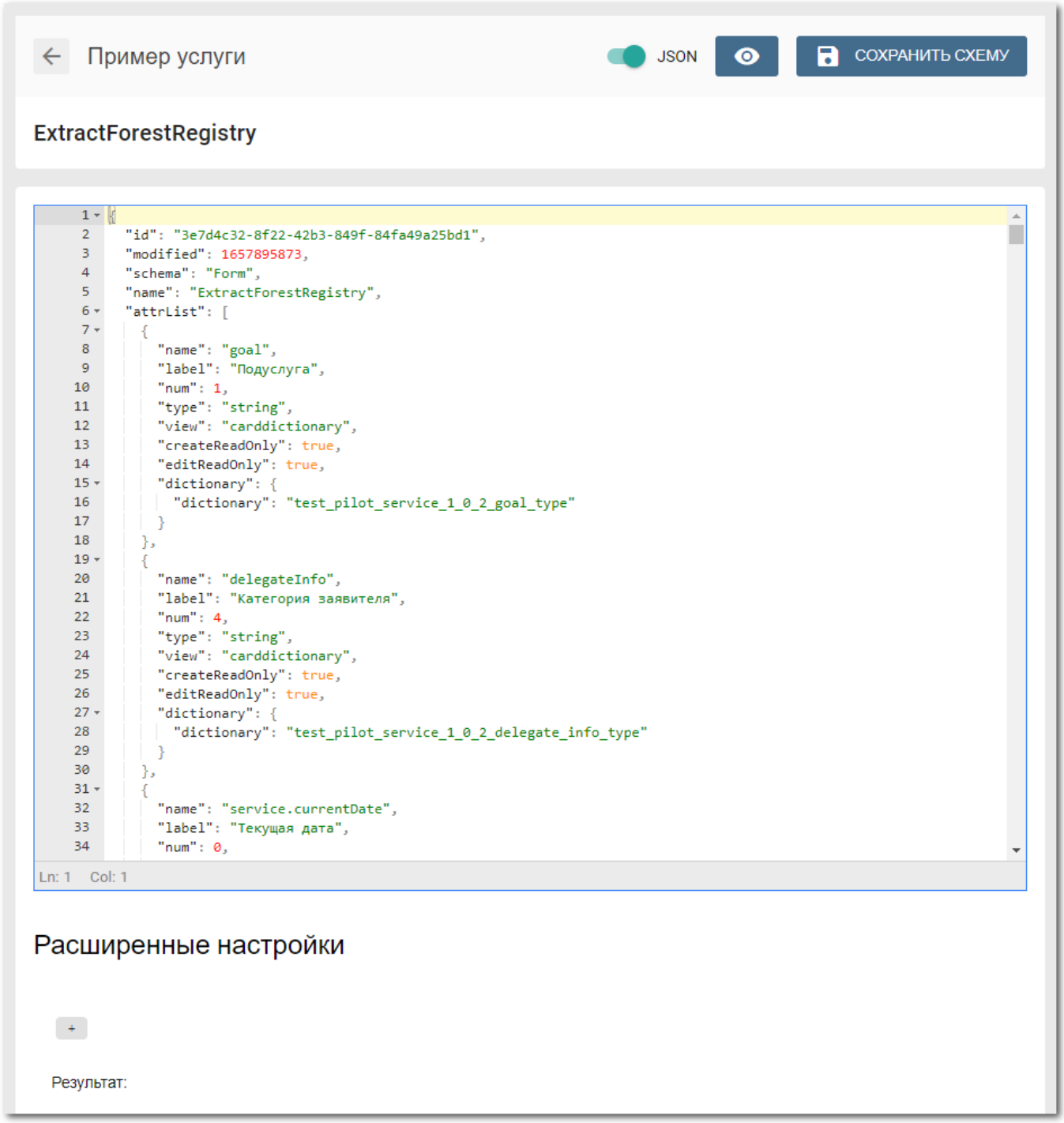


Рисунок 137 — Страница редактирования экранной формы

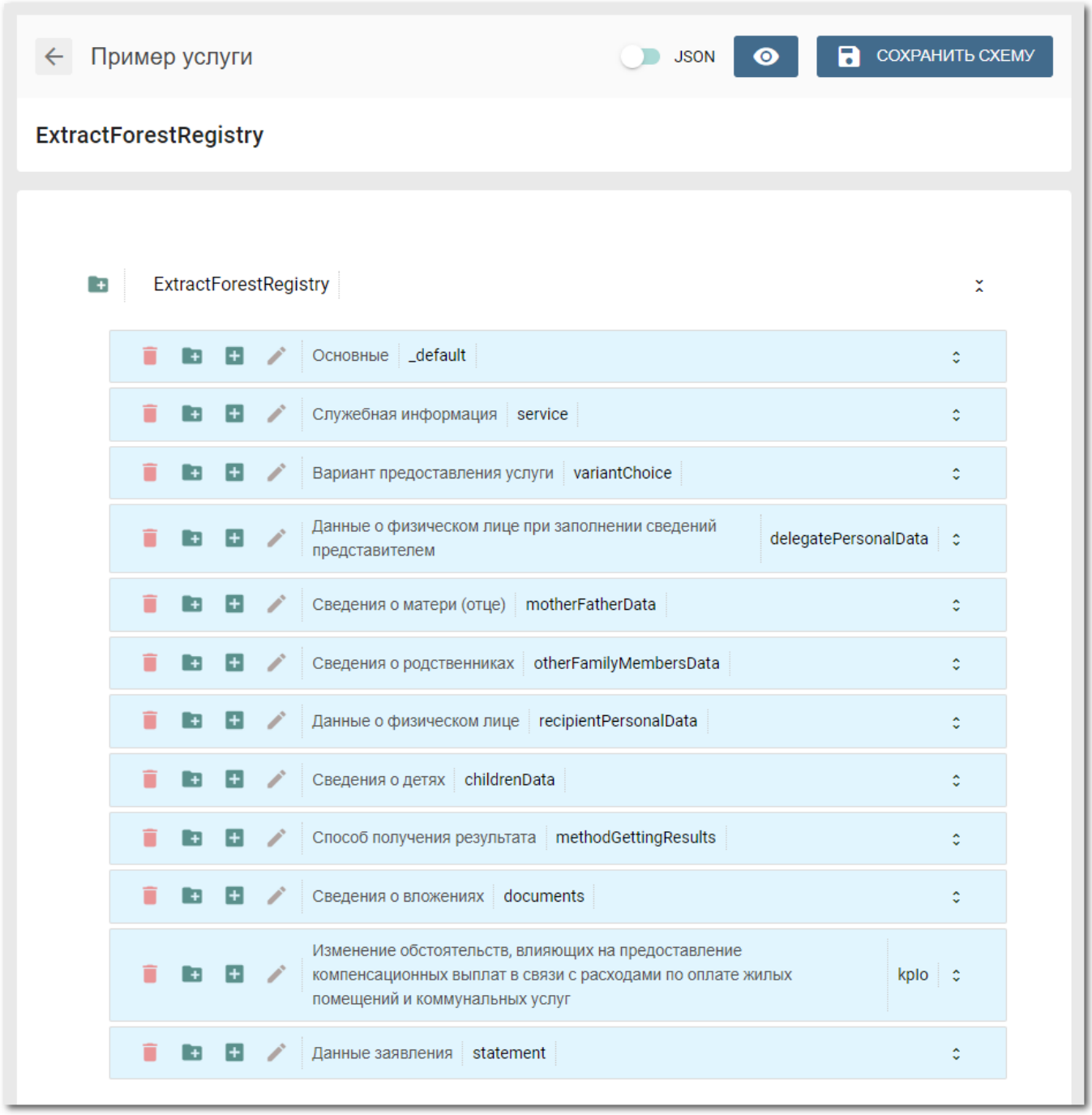


Рисунок 138 — Страница редактирования экранной формы. Режим визуального конструктора

При создании и редактировании экранных форм с помощью встроенного редактора рекомендуется всегда сохранять копию кода на АРМ.

3.14.2. Описание основных компонентов кода экранной формы

Экранная форма состоит их двух блоков двух видов: groupList и attrList (рисунки 139 и 140).

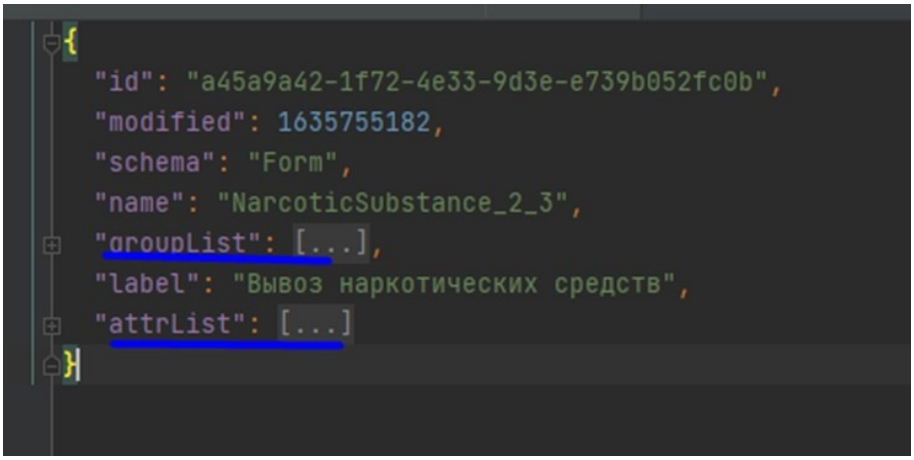


Рисунок 139 — Код экранной формы. Блоки groupList и attrList (в свернутом виде)

groupList предназначен для группировки атрибутов в группы данных. Эти группы должны полностью соответствовать XSD-схеме, иметь такой же порядок (рисунок 140).

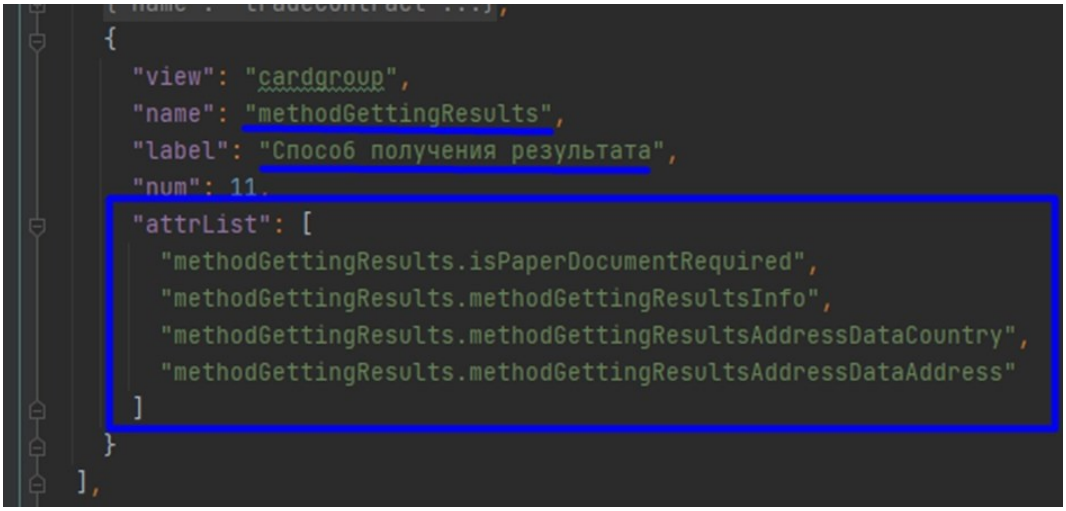


Рисунок 140 — Код экранной формы. Группировка атрибутов: блок attrList внутри groupList

Компонент groupList имеет тип cardgroup. *card* в начале каждого типа говорит о том, что эта группа может быть необязательной для заявления, т. е. в случае отсутствия этого блока в XML, он не появится в экранной форме.

Компонент должен иметь такое же наименование, как в таблице соответствия атрибутов для процесса импорта (левая колонка, см. подраздел 3.5) или как в описании (коде) заявления в ПГС (без application).

Примечание. Для просмотра описания заявления в ПГС необходимо перейти в раздел ПГС «Заявления», открыть карточку требуемого заявления и открыть в браузере Инструменты разработчика (в браузере Chrome открываются нажатием клавиши F12 или комбинации клавиш Ctrl +Shift + I).

label соответствует наименованию блока данных в XSD-схеме (ср. рисунки 140 и 141).

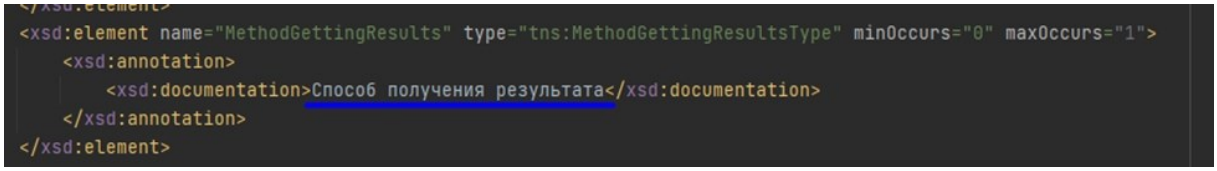


Рисунок 141 — Комментарий в XSD-схеме услуги,
которому соответствует label в коде экранной формы

attrList внутри groupList описывает все атрибуты, которые объединены этой группой, согласно XSD-схеме (ср. рисунки 140, 141 и 142).

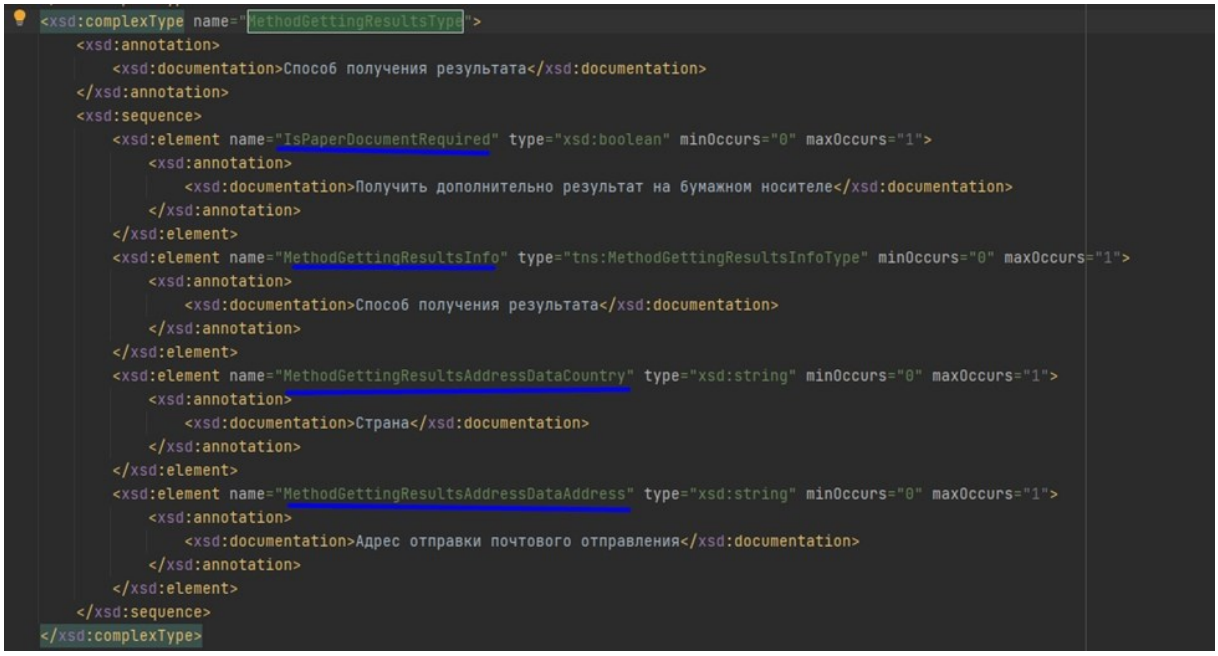


Рисунок 142 — Атрибуты в XSD-схеме услуги, которым соответствует набор атрибутов в attrList

Далее в отдельном attrList необходимо описать все компоненты из каждого attrList, присутствующего в groupList (рисунок 143).

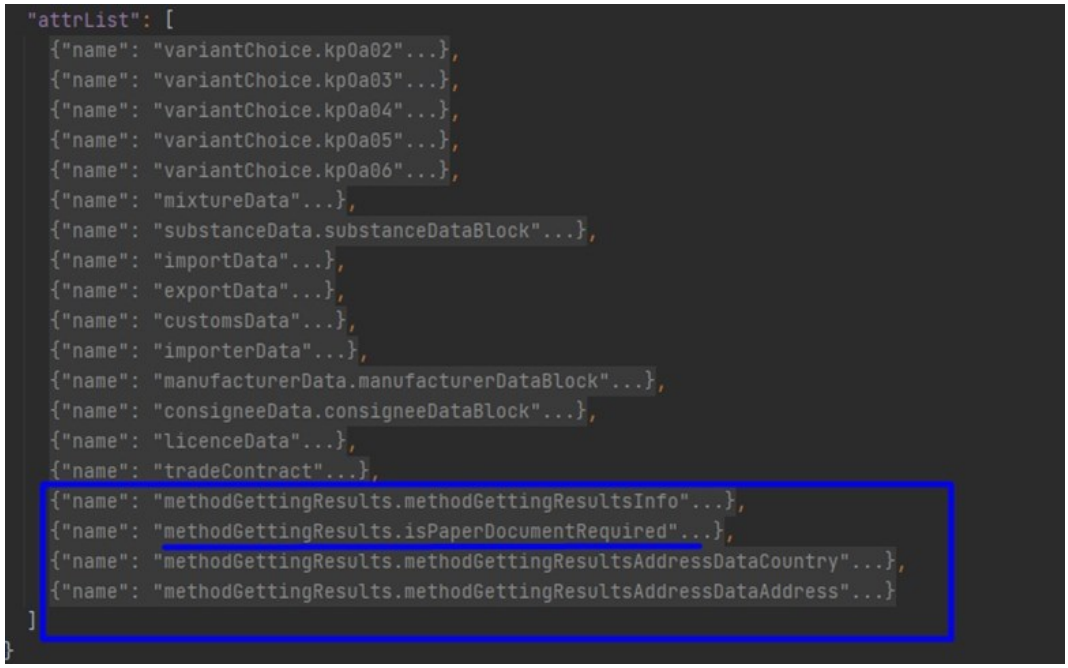


Рисунок 143 — Атрибуты attrList, приведенного на рисунке 140, в отдельном attrList

3.14.3. Кодирование свойств атрибутов

С помощью определения свойств атрибутов можно настраивать, как тот или иной атрибут должен отображаться в экранной форме.

Свойство name задает наименование атрибута экранной формы. Оно должно совпадать с наименованием в таблице соответствия атрибутов для процесса импорта (левая колонка) или в описании (коде) заявления в ПГС (без application), т. е. должно описывать путь к данным (рисунок 144).

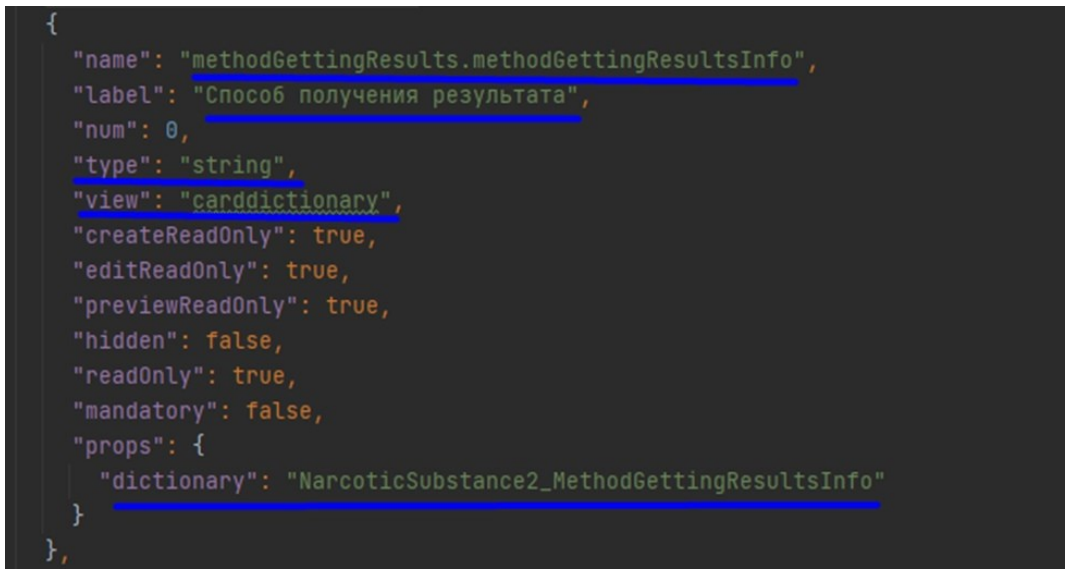


Рисунок 144 — Свойства атрибута, связанного со справочником

В примере на рисунке 144 представлен *атрибут, который связан со справочником*, заведенным в ПГС на вкладке «Справочники» (рисунок 145). Он получает из заявления числовое

значение кода записи справочника и выводит на экран текстовое значение записи справочника, соответствующее данному коду (рисунок 146). Атрибут имеет тип carddictionary. props задает свойства для этого атрибута.

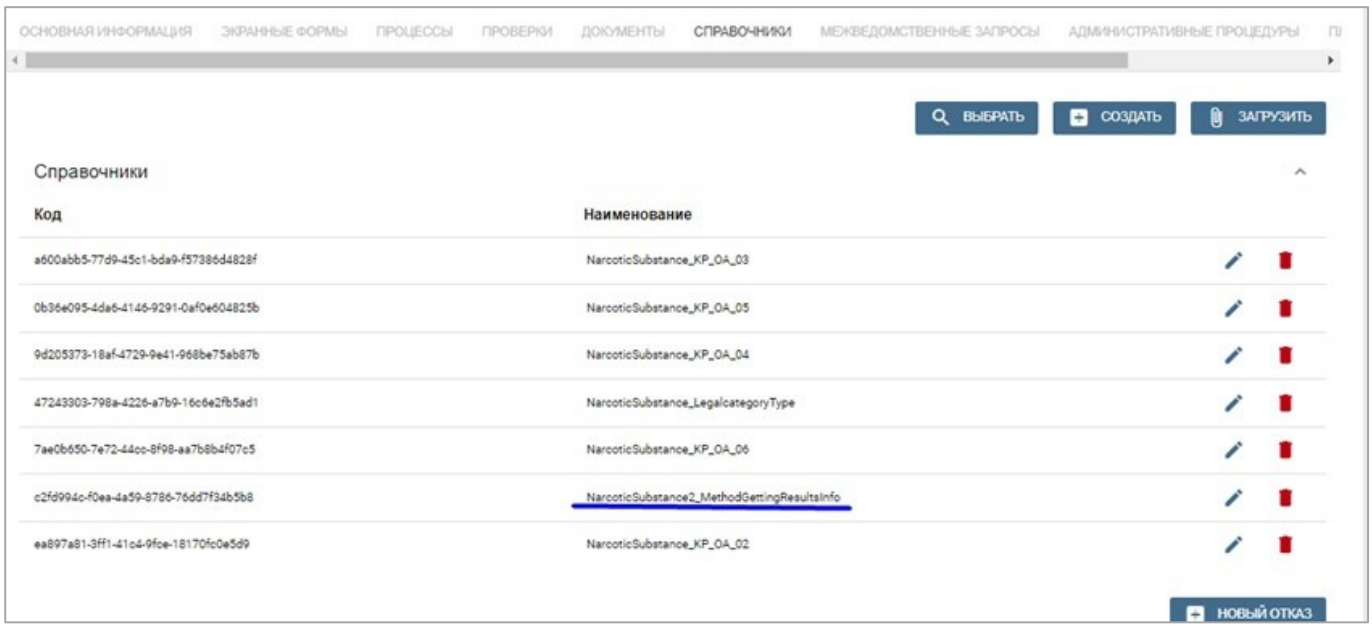


Рисунок 145 — Справочник, с которым связан атрибут на рисунке 144, в списке справочников

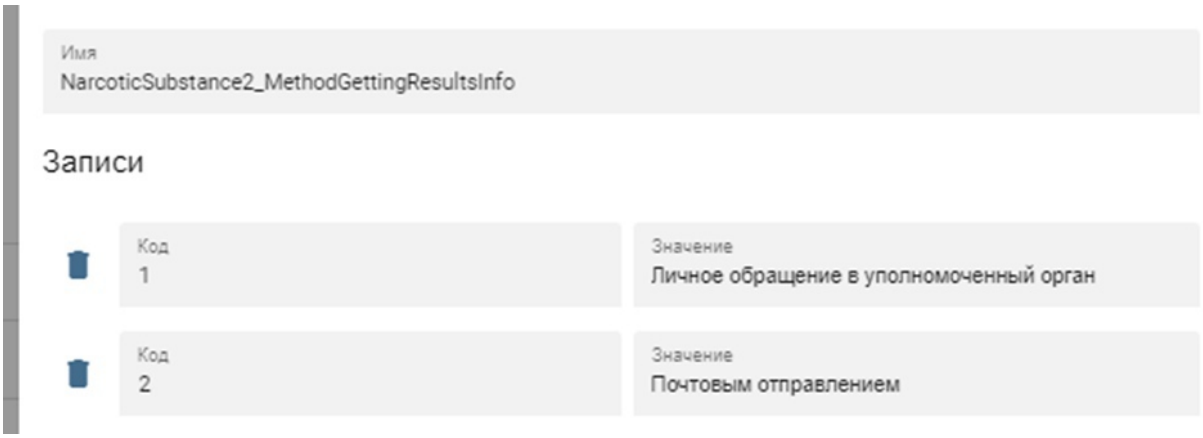


Рисунок 146 — Значения справочника, с которым связан атрибут на рисунке 144

Атрибут на рисунке 147 описывает ответ на вопрос, который может иметь значения только true («истина», «да») и false («ложь», «нет»).

Во избежание перегрузки экранной формы информацией рекомендуется выводить такие атрибуты на экранную форму, только если их значением является true. Для этого необходимо добавить в props свойство *"hideOnFalsy": "true"*, а также указать для атрибута свойство *"view": "string"*. В этом случае такие атрибуты отображаются в виде ответа «Да» на вопрос в заявлении, которому соответствует атрибут (рисунок 148).

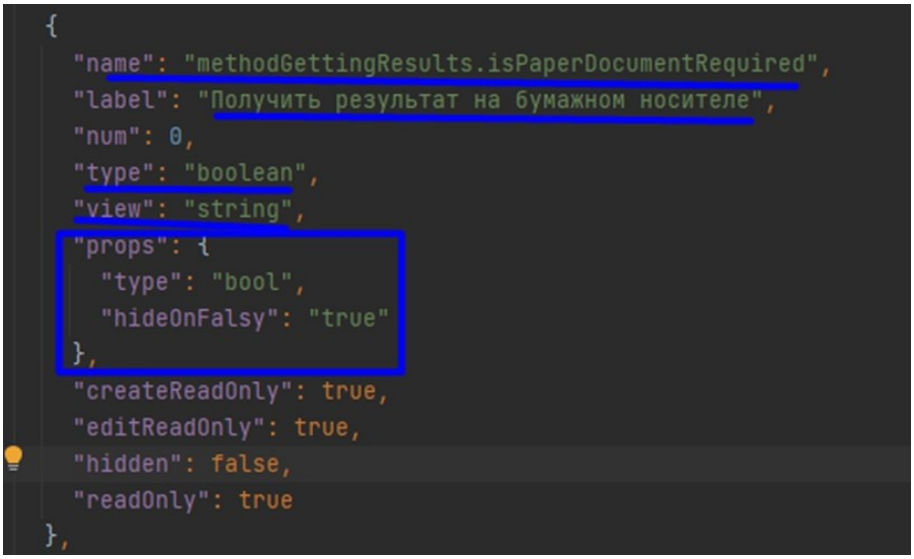


Рисунок 147 — Свойства атрибута булевского типа

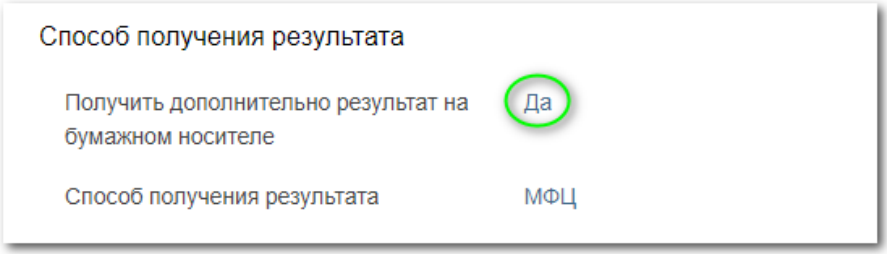


Рисунок 148 — Представление атрибута булевского типа на экране

Тип cardstring (рисунок 149) используется для необязательных атрибутов, которые могут как присутствовать, так и отсутствовать в заявлении, значения которых выводятся на экран как текст. Для описания дат также может использоваться атрибут типа cardstring.

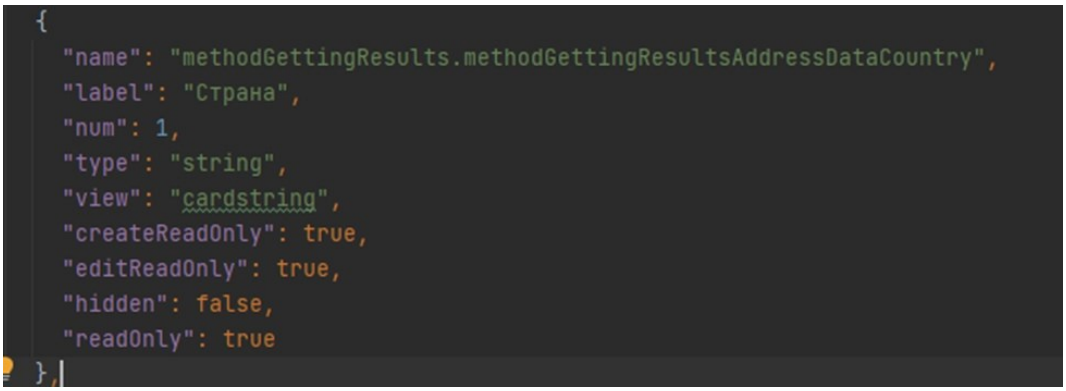


Рисунок 149 — Карточка заявления о предоставлении услуги. Данные заявления

Если в XSD-схеме услуги предусмотрено, что данные могут быть массивом, т. е. блоков ConsigneeDataBlock может быть сколько угодно внутри ConsigneeDataType (свойство *maxOccurs="unbounded"* — рисунок 150), то для описания в экранной форме заявления удобно использовать компонент типа cardtable.

Для этого необходимо определить компонент groupList и указать в его attrList путь до повторяющихся элементов[1]. Путь наиболее просто получить, скопировав его из таблицы соответствия атрибутов для процесса импорта (см. подраздел 3.6) от начала до символа «*»[2] не включительно (ср. рисунки 151 и 152).

```
<xsd:complexType name="ConsigneeDataType">
    <xsd:annotation>
        <xsd:documentation>Данные грузополучателя</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="ConsigneeDataBlock" type="tns:ConsigneeDataBlockType" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ConsigneeDataBlockType">
    <xsd:annotation>
        <xsd:documentation>Данные грузополучателя</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="ConsigneeName" type="xsd:string" minOccurs="0" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>Полное наименование</xsd:documentation>
```

Рисунок 150 — Массив данных в XSD-схеме услуги

manufacturerData.manufacturerDataBlock*	ManufacturerData.ManufacturerDataBlock*
consigneeData.consigneeDataBlock*.consi	ConsigneeData.ConsigneeDataBlock*.Con
consigneeData.consigneeDataBlock*.consi	ConsigneeData.ConsigneeDataBlock*.Con
consigneeData.consigneeDataBlock*.consi	ConsigneeData.ConsigneeDataBlock*.Con
consigneeData.consigneeDataBlock*.consi	ConsigneeData.ConsigneeDataBlock*.Con
consigneeData.consigneeDataBlock*.consi	ConsigneeData.ConsigneeDataBlock*.Con
consigneeData.consigneeDataBlock*.consi	ConsigneeData.ConsigneeDataBlock*.Con
licenceData.licenceNumber	LicenceData.LicenceNumber

Рисунок 151 — Таблица соответствия атрибутов
в задаче «Собираем бизнес-данные» процесса импорта

```
{
  "view": "cardtable",
  "num": 8,
  "name": "consigneeData.consigneeDataBlock",
  "label": "Данные грузополучателя",
  "attrList": [
    "consigneeData.consigneeDataBlock"
  ]
},
```

Рисунок 152 — Компонент groupList, содержащий массив данных, в JSON-коде экранной формы

Далее внутри groupList следует описать один компонент attrList, в нем указать свойство listAttrs, в котором описать все элементы данных из клонируемого блока (массива) — рисунок 153.

[1] Имя компонента attrList "name" допускается любое, однако для удобства рекомендуется в качестве имени также указать путь до повторяющихся элементов.

[2] Символом «*» в таблице соответствия атрибутов обозначаются атрибуты, которых может быть более одного внутри блока данных (элементы массива).

Рисунок 153 — Компонент attrList. Свойство listAttrs, содержащее несколько атрибутов

Затем необходимо детализировать описание атрибутов в listAttrs, добавив свойства (рисунок 154). Каждый атрибут описывается отдельно. Таким же образом можно описать даты и справочники.

Рисунок 154 — Свойства атрибутов внутри listAttrs

Далее приведен пример реализации экранной формы.

```
groupList
{  "view": "cardgroup",
  "num": 8,
  "name": "consigneeData.consigneeDataBlock",
  "label": "Данные грузополучателя",
  "attrList": [
    "consigneeData.consigneeDataBlock"
  ]
},attrList{
  "name": "consigneeData.consigneeDataBlock",
  "num": 0,
  "type": "linkMN",
  "view": "cardtable",
  "createReadOnly": true,
  "editReadOnly": true,
  "previewReadOnly": true,
  "reference": {
    "mode": "grid",
    "listAttrs": [
      {
        "name": "consigneeName",
        "label": "Полное наименование",
        "hidden": false,
        "sortable": false
      },
```

```
      {
        "name": "consigneeCountry",
        "label": "Страна",
        "hidden": false,
        "sortable": false
      },
      {
        "name": "consigneeCountryCode",
        "label": "Код страны",
        "hidden": false,
        "sortable": false
      },
      {
        "name": "consigneeAddressFiasFullCode",
        "label": "Адрес, код ФИАС",
        "hidden": false,
        "sortable": false
      },
      {
        "name": "ConsigneeAddress",
        "label": "Адрес",
        "hidden": false,
        "sortable": false
      },
      {
        "name": "consigneeFax",
        "label": "Факс",
        "hidden": false,
        "sortable": false
      },
      {
        "name": "consigneePhone",
        "label": "Телефон",
        "hidden": false,
        "sortable": false
      },
      {
        "name": "consigneeEmail",
        "label": "Электронная почта",
        "hidden": false,
        "sortable": false
      }
    ]
  },
```

3.14.3.1 Реализация в вложенных массивов в экранных формах

Помимо простых массивов, в услугах также могут присутствовать блоки данных, содержащие внутри себя как отдельные блоки данных, так и клонируемые блоки.

ChildrenData в XSD-схеме услуги на рисунке 155 является клонируемым блоком (массивом) и внутри себя содержит блок DeathData, который в свою очередь также может быть клонируемым (массивом), — рисунок 156. После импорта данных из XSD-схемы услуги эти вложенные массивы в описании (коде) заявления в ПГС выглядят, как на рисунке 157.

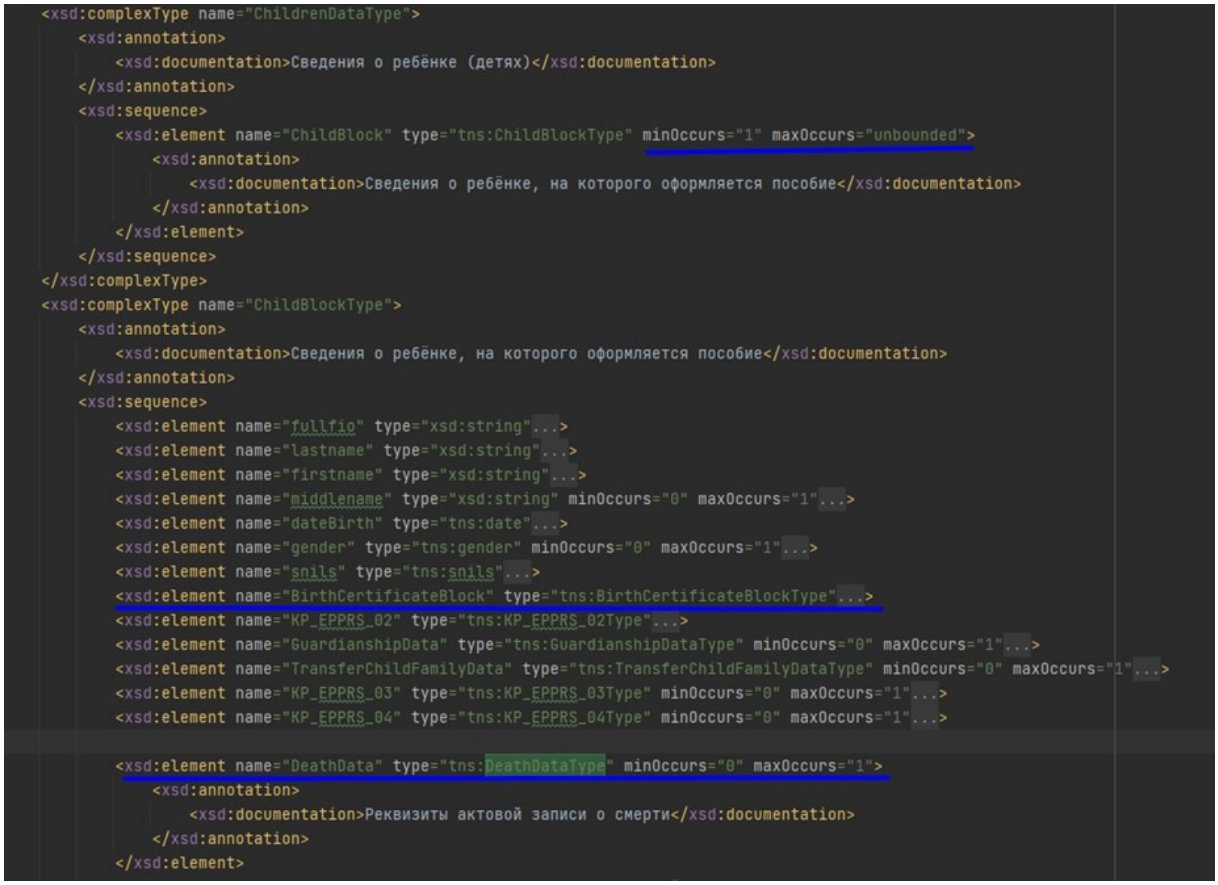


Рисунок 155 — Вложенные массивы данных в XSD-схеме услуги

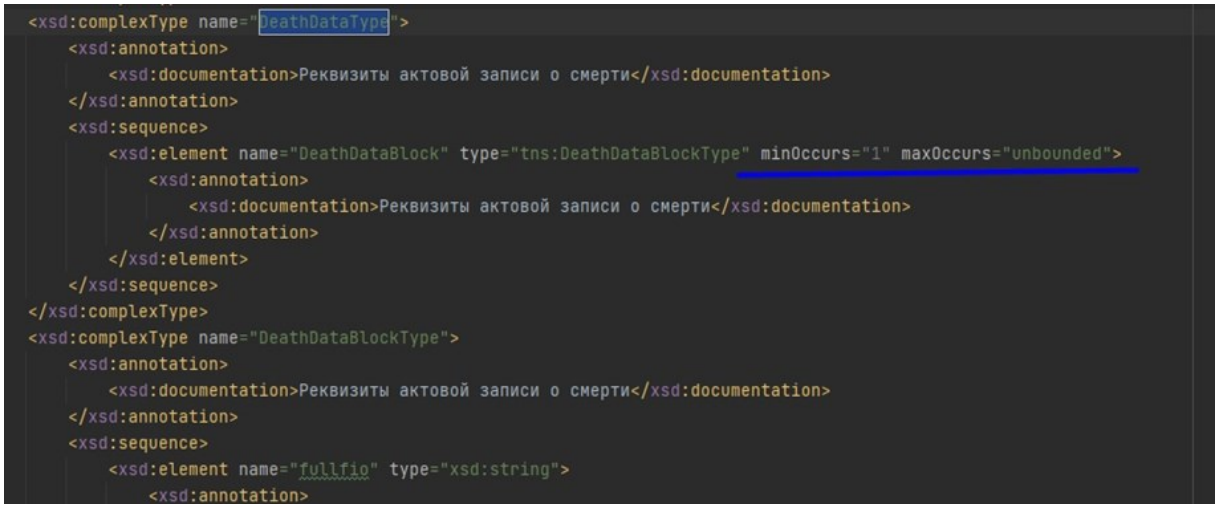


Рисунок 156 — Описание типа DeathDataType в той же XSD-схеме услуги, что и на рисунке 155

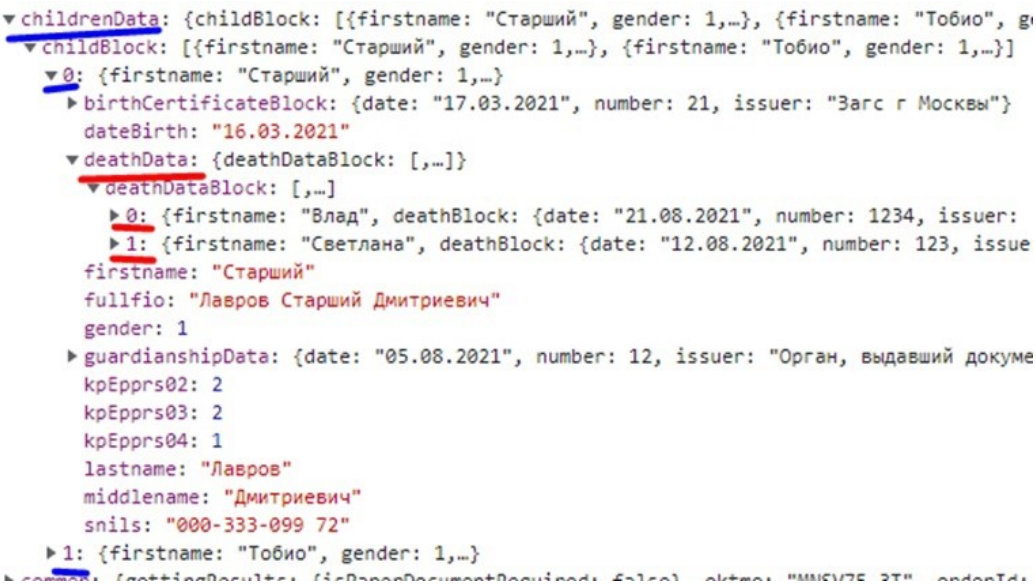


Рисунок157 — Вложенные массивы данных в описании заявления в ПГС

Кодирование вложенных массивов осуществляется следующим образом.

В groupList следует описать атрибут childrenData типа linkMN, в его attrList описать атрибут также типа linkMN, наименованием которого является путь до повторяющихся элементов: childrenData.childBlock (указан в таблице соответствия атрибутов в процессе импорта, в левой колонке от начала до символа «*» не включительно), — рисунок 158.

Рисунок 158 — Компонент groupList, содержащий вложенные массивы данных, в JSON-коде экранной формы

Для атрибута в attrList необходимо задать свойство-ссылку reference на компонент linked, как показано на рисунке 159. В свойстве entity внутри reference необходимо указать наименование компонента linked (наименование может отличаться от приведенного на рисунке).

Ниже attrList описывается компонент linkedSchema (рисунок 160).

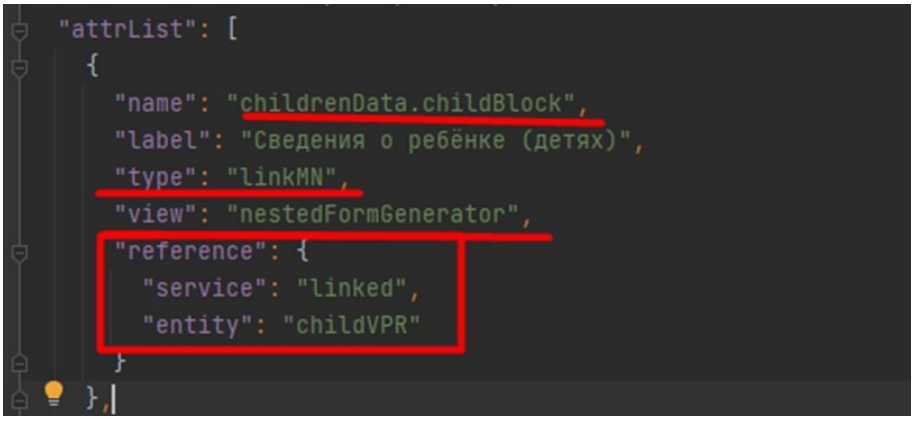


Рисунок 159 — Свойство reference в описании атрибута

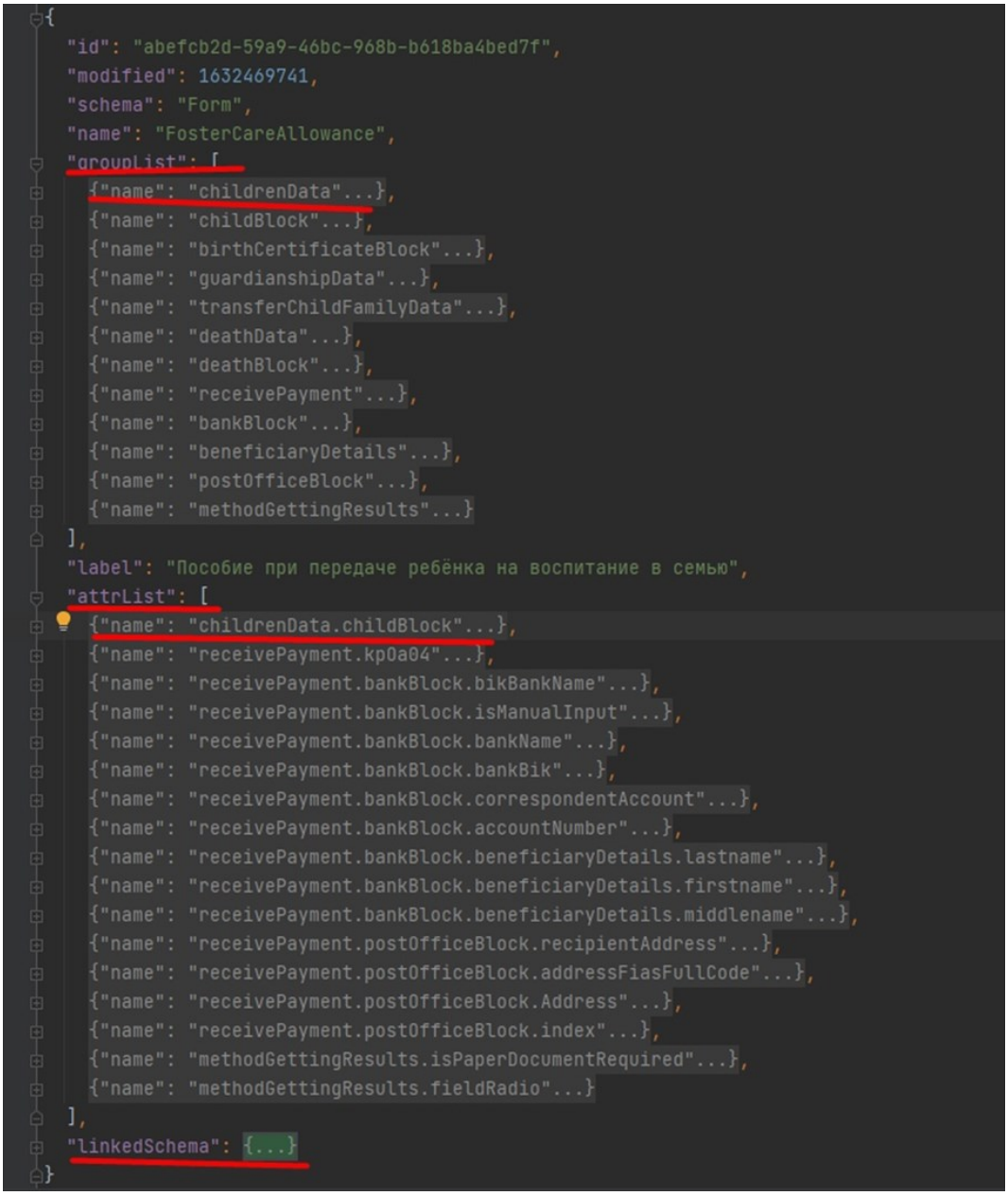


Рисунок 160 — Последовательность компонентов groupList, attrList и linkedSchema в коде экранной формы

linkedSchema должна также содержать в себе компонент groupList, ниже которого следует добавить компонент linked (рисунок 161).

Внутри linked необходимо описать компонент, наименование которого должно совпадать с указанным в свойстве entity внутри описанного выше свойства reference, а внутрь данного компонента добавить компонент attrList (рисунок 161). Атрибуты в attrList описываются так же, как и в обычном attrList экранной формы (рисунок 162). В groupList необходимо указать наименования всех компонентов и блоков, описанных в данном attrList.

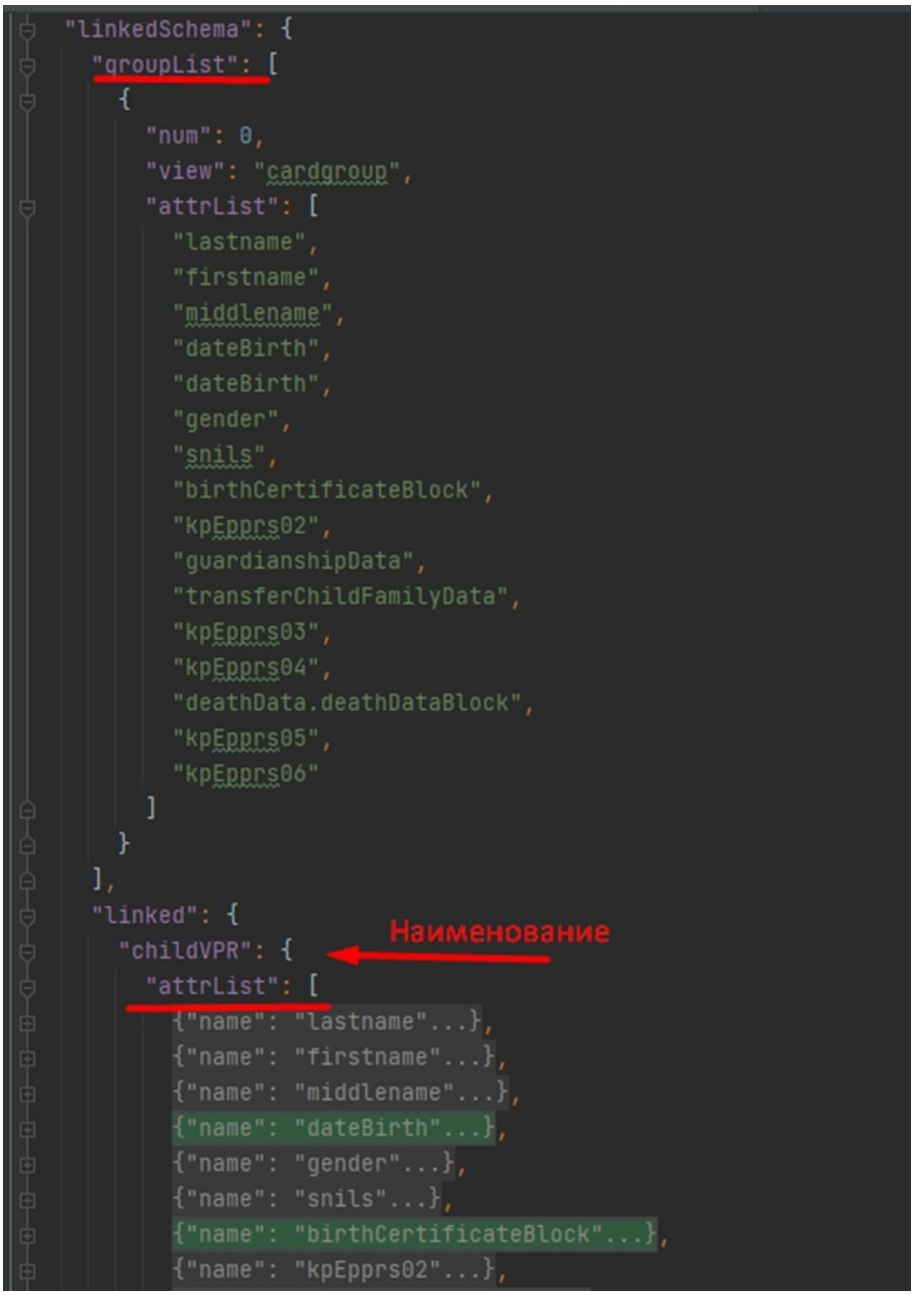


Рисунок 161 — Описание компонента linked

Рисунок 162 — Описание атрибутов в attrList внутри linked

В качестве значения для свойства name в клонируемом блоке deathData необходимо указать путь до повторяющихся элементов (см. в таблице соответствия атрибутов в процессе импорта, в левой колонке от начала до символа «*» не включительно) без учета той части пути, которая указывается
в вышестоящем клонируемом блоке childrenData.

Далее внутри deathData описываются все его атрибуты обычным образом.

Рисунок 163 — Описание клонируемого блока в attrList внутри linked

Далее приведен пример реализации компонента linkedSchema.

```
"linkedSchema": {
  "groupList": [
    {
      "num": 0,
      "view": "cardgroup",
      "attrList": [
        "lastname",
        "firstname",
        "middlename",
        "dateBirth",
        "dateBirth",
        "gender",
        "snils",
        "birthCertificateBlock",
        "kpEpprs02",
        "guardianshipData",
        "transferChildFamilyData",
        "kpEpprs03",
        "kpEpprs04",
        "deathData.deathDataBlock",
        "kpEpprs05",
        "kpEpprs06"
      ]
    }
  ],
  "linked": {
    "childVPR": {
      "attrList": [
        {
          "num": 1,
          "label": "Фамилия",
          "name": "lastname",
          "type": "string",
          "view": "cardstring",
          "previewHidden": false
        },
        {
          "num": 2,
          "label": "Имя",
          "name": "firstname",
          "type": "string",
          "view": "cardstring",
          "previewHidden": false
        },
        {
          "num": 3,
          "label": "Отчество",
          "name": "middlename",
          "type": "string",
          "view": "cardstring",
          "previewHidden": false
        },
        {
          "num": 4,
          "label": "Дата рождения",
```

```
      "name": "dateBirth",
      "type": "string",
      "view": "cardstring",
      "previewHidden": false,
      "props": {
        "type": "date",
        "readonlyAsText": true
      }
    },
    {
      "num": 5,
      "label": "Пол",
      "name": "gender",
      "type": "string",
      "view": "carddictionary",
      "previewHidden": false,
      "editReadOnly": true,
      "createReadOnly": true,
      "props": {
        "dictionary": "FosterCareAllowance_genderChild"
      }
    },
    {
      "num": 6,
      "label": "СНИЛС",
      "name": "snils",
      "type": "string",
      "view": "cardstring",
      "previewHidden": false
    },
    {
      "name": "birthCertificateBlock",
      "label": "Реквизиты актовой записи о рождении ребёнка",
      "type": "linkMN",
      "view": "cardtable",
      "reference": {
        "mode": "grid",
        "listAttrs": [
          {
            "num": 1,
            "label": "Номер актовой записи о рождении ребёнка",
            "name": "number",
            "type": "string",
            "view": "cardstring",
            "previewHidden": false
          },
          {
            "num": 2,
            "label": "Дата",
            "name": "date",
            "type": "string",
            "view": "cardstring",
            "previewHidden": false
          },
          {
            "num": 3,
            "label": "Место государственной регистрации",
```

```
                "name": "issuer",
                "type": "string",
                "view": "cardstring",
                "previewHidden": false
              }
            ]
          }
        },
        {
          "num": 7,
          "label": "Кем является заявитель?",
          "name": "kpEpprs02",
          "type": "string",
          "view": "carddictionary",
          "previewHidden": false,
          "editReadOnly": true,
          "createReadOnly": true,
          "props": {
            "hideOnFalsy": true,
            "dictionary": "FosterCareAllowance_KP_EPPRS_02"
          }
        },
        {
          "name": "guardianshipData",
           "label":  "Реквизиты документа, подтверждающего установление опеки
(попечительства) над ребёнком",
          "type": "linkMN",
          "view": "cardtable",
          "reference": {
            "mode": "grid",
            "listAttrs": [
              {
                "num": 1,
                "label": "Номер документа",
                "name": "number",
                "type": "string",
                "view": "cardstring",
                "previewHidden": false
              },
              {
                "num": 2,
                "label": "Дата",
                "name": "date",
                "type": "string",
                "view": "cardstring",
                "previewHidden": false
              },
              {
                "num": 3,
                "label": "Орган, выдавший документ",
                "name": "issuer",
                "type": "string",
                "view": "cardstring",
                "previewHidden": false
              }
            ]
          }
```

```
    },
    {
      "name": "transferChildFamilyData",
      "label": "Реквизиты документа о передаче ребёнка в семью",
      "type": "linkMN",
      "view": "cardtable",
      "reference": {
        "mode": "grid",
        "listAttrs": [
          {
            "num": 1,
            "label": "Номер документа",
            "name": "number",
            "type": "string",
            "view": "cardstring",
            "previewHidden": false
          },
          {
            "num": 2,
            "label": "Дата",
            "name": "date",
            "type": "string",
            "view": "cardstring",
            "previewHidden": false
          },
          {
            "num": 3,
            "label": "Орган, выдавший документ",
            "name": "issuer",
            "type": "string",
            "view": "cardstring",
            "previewHidden": false
          }
        ]
      }
    },
    {
      "num": 8,
      "label": "По какой причине ребёнок передан на воспитание в семью?",
      "name": "kpEpprs03",
      "type": "string",
      "view": "carddictionary",
      "previewHidden": false,
      "editReadOnly": true,
      "createReadOnly": true,
      "props": {
        "hideOnFalsy": true,
        "dictionary": "FosterCareAllowance_KP_EPPRS_03"
      }
    },
    {
      "num": 8,
      "label": "Документ, подтверждающий смерть родителей",
      "name": "kpEpprs04",
      "type": "string",
      "view": "carddictionary",
      "previewHidden": false,
```

```
      "editReadOnly": true,
      "createReadOnly": true,
      "props": {
        "hideOnFalsy": true,
        "dictionary": "FosterCareAllowance_KP_EPPRS_04"
      }
    },
    {
      "name": "deathData.deathDataBlock",
      "label": "Реквизиты актовой записи о смерти",
      "type": "linkMN",
      "view": "cardtable",
      "reference": {
        "mode": "grid",
        "listAttrs": [
          {
            "num": 1,
            "label": "Фамилия",
            "name": "lastname",
            "type": "string",
            "view": "cardstring",
            "previewHidden": false
          },
          {
            "num": 2,
            "label": "Имя",
            "name": "firstname",
            "type": "string",
            "view": "cardstring",
            "previewHidden": false
          },
          {
            "num": 3,
            "label": "Отчество",
            "name": "middlename",
            "type": "string",
            "view": "cardstring",
            "previewHidden": false
          },
          {
            "num": 4,
            "label": "Дата рождения",
            "name": "dateBirth",
            "type": "string",
            "view": "cardstring",
            "previewHidden": false
          },
          {
            "num": 1,
            "label": "Номер документа",
            "name": "deathBlock.number",
            "type": "string",
            "view": "cardstring",
            "previewHidden": false
          },
          {
            "num": 2,
```

```
                "label": "Дата",
                "name": "deathBlock.date",
                "type": "string",
                "view": "cardstring",
                "previewHidden": false
              },
              {
                "num": 3,
                "label": "Орган, выдавший документ",
                "name": "deathBlock.issuer",
                "type": "string",
                "view": "cardstring",
                "previewHidden": false
              }
            ]
          }
        },
        {
          "num": 9,
          "label": "Документ, подтверждающий отсутствие родителей",
          "name": "kpEpprs05",
          "type": "string",
          "view": "carddictionary",
          "previewHidden": false,
          "editReadOnly": true,
          "createReadOnly": true,
          "props": {
            "hideOnFalsy": true,
            "dictionary": "FosterCareAllowance_KP_EPPRS_05"
          }
        },
        {
          "num": 10,
          "label": "Подтверждающий документ",
          "name": "kpEpprs06",
          "type": "string",
          "view": "carddictionary",
          "previewHidden": false,
          "editReadOnly": true,
          "createReadOnly": true,
          "props": {
            "hideOnFalsy": true,
            "dictionary": "FosterCareAllowance_KP_EPPRS_06"
          }
        }
      ],
      "label": "Ребенок"
    }
  }
}
```

Вид экранной формы, соответствующей приведенному выше примеру, представлен на рисунке 164.

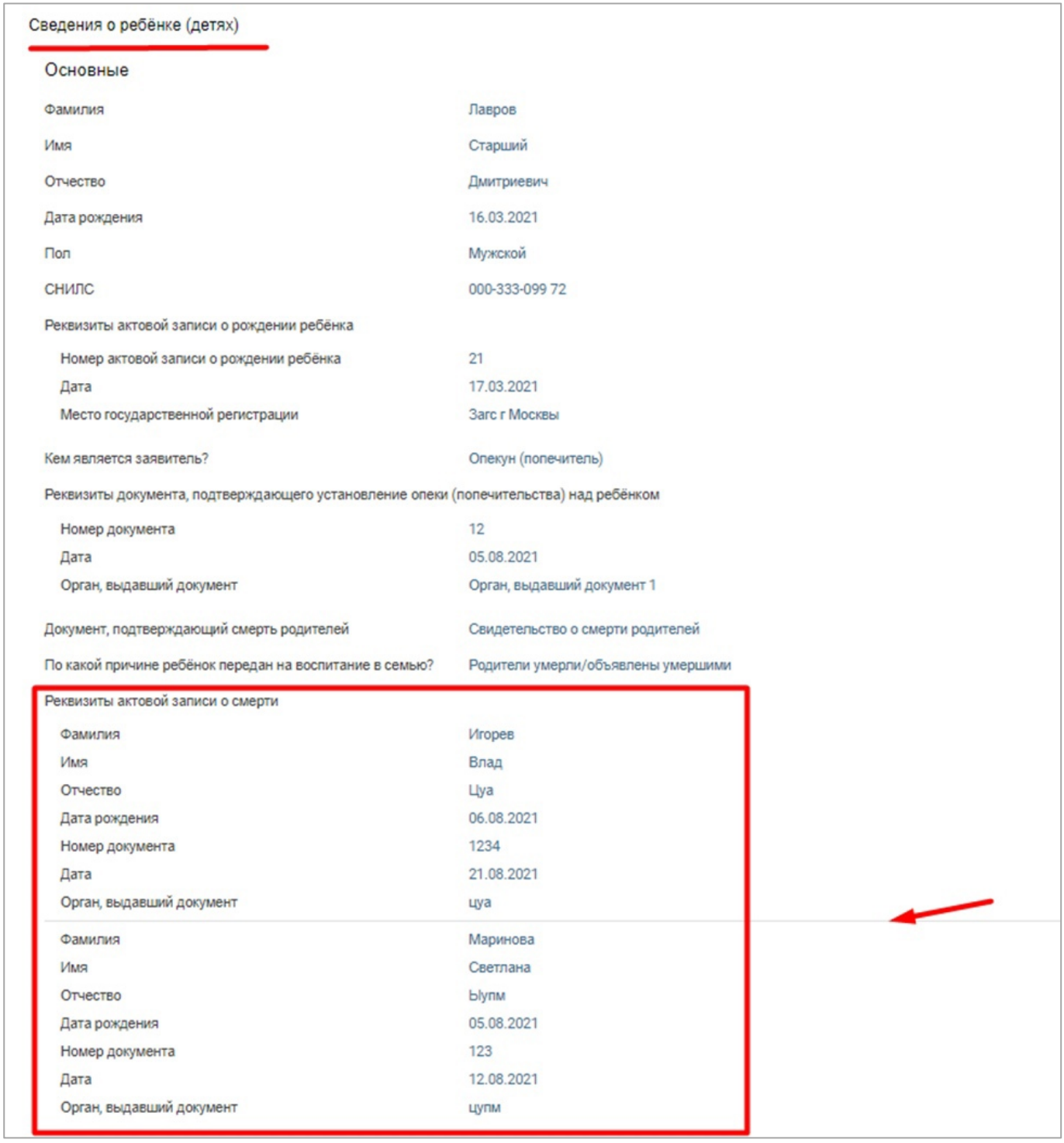


Рисунок 164 — Пример экранной формы, кодированной с использованием атрибутов linkedSchema и linked (вложенные массивы)

3.15. Добавление шаблонов выходных документов для предоставления услуги

Для каждого предусмотренного в услуге вида выходного документа необходимо предварительно подготовить и загрузить в услугу шаблон, в соответствии с которым ПГС будет генерировать документы данного вида.

Как правило, в документации на услугу содержатся формы *результирующих документов*. Пример такой формы представлен на рисунке 165. В них кодами, заключенными в двойные угловые

скобки (например, <<P.126.00.06>> и аналогичные), обозначаются атрибуты, которые берутся из заявления о предоставлении услуги или были получены в ходе предоставления услуги. Например, на рисунке 166 код P.126.00.06 обозначает ф. и. о. заявителя, указанные в заявлении. Коды в форме размещаются в тех местах, в которые должны быть подставлены соответствующие значения атрибутов.

Форма решения о предоставлении государственной услуги

<<P.126.00.01>>

(наименование уполномоченного органа исполнительной власти
субъекта Российской Федерации или органа местного самоуправления)

Кому <<P.126.00.06>>
<<P.126.00.11>>
<<P.126.00.12>>

РЕШЕНИЕ
о предоставлении услуги «Назначение и выплата единовременного пособия при передаче ребёнка на воспитание в семью»

от <<P.126.00.02>> № <<P.126.00.03>>

По результатам рассмотрения заявления от <<P.126.00.04>> № <<P.126.00.05>> и приложенных к нему документов, на основании Федерального закона от 19.05.1995 №81-ФЗ «О государственных пособиях гражданам, имеющим детей», приказа Минтруда России от 29.09.2020 № 668н «Об утверждении Порядка и условий назначения и выплаты государственных пособий гражданам, имеющим детей», принято решение предоставить государственную услугу по назначению и выплате единовременного пособия при передаче ребёнка на воспитание в семью:

<<P.126.00.06>>, <<P.126.00.09>>
(Ф.И.О, дата рождения)

в размере <<P.126.01.01>> рублей.

Рисунок 165 — Пример формы результирующего документа

6	P.126.00.06	ФИО заявителя	Системные данные	Строка	250	Нет

Рисунок 166 — Соответствие кода в форме результирующего документа сведениям из заявления о предоставлении услуги

Шаблон вида документа должен представлять собой документ Microsoft Word, подготовленный на основе данной формы или разработанный специалистом специально для услуги,

в котором для обозначения присутствующих атрибутов вместо кодов в угловых скобках используются соответствующие теги в нотации Mustache (https://mustache.github.io/mustache.5.html , англ.).

Используются следующие теги:

– *простая переменная.* Имеет вид: *{{name}}*. В качестве наименований тегов такого вида, заключаемых в двойные фигурные скобки, следует указывать наименования соответствующих атрибутов в описании (коде) заявления в ПГС с учетом пути (пример: {{resolutionDocument.resDate}}). Формируя результирующий документ по шаблону, ПГС будет искать атрибут с указанным наименованием в коде заявления в ПГС. Если атрибут будет найден, то в результирующем документе вместо тега будет подставлено его значение. Если атрибут не будет найден, то в результирующем документе вместо тега значение будет отсутствовать;

– *блок.* Имеет открывающий тег вида *{{#person}}* и закрывающий вида *{{/person}}*. В качестве наименований таких тегов могут выступать как полные наименования атрибутов, так и первые части наименований составных атрибутов (путь). Наименование в открывающем и закрывающем тегах должно быть одним и тем же. Содержимым блока могут являться как фрагменты текста документа, так и другие теги. Например, для отображения значений составных атрибутов delegate.fullName и delegate.phone следует определить блок {{#delegate}}…{{/delegate}} и внутри него указать теги (в виде простых переменных либо также блоков в зависимости от требований к шаблону) с наименованиями fullName и phone. Вместо тегов, указанных в теле блока, в результирующий документ подставляются значения соответствующих атрибутов. При этом в результирующий документ будут подставлены значения только тех атрибутов, которые будут найдены в коде заявления в ПГС. Если в коде заявления не будет найден ни один атрибут, наименование которого совпадает или содержит в качестве пути наименование блока, то все содержимое блока будет отсутствовать в документе;

– *блок по отсутствию*. Имеет открывающий тег вида *{{#person}}* и закрывающий вида *{{/person}}*. Оформляется аналогично предыдущему, но содержимое блока отображается в случае *отсутствия* атрибута с наименованием, совпадающим с наименованием блока;

– *тег {{.}}*. Используется внутри блоков. Подставляет в документ значение атрибута с наименованием, совпадающим с наименованием блока, если атрибут имеет единственное значение; подставляет все значения массива, если атрибут является массивом.

Пример фрагмента шаблона приведен на рисунке 167.

Шаблон *печати заявления* разрабатывается специалистом также с использованием Mustache.

Добавление видов выходных документов осуществляется, аналогично добавлению входных, на вкладке карточки услуги «Документы» по нажатию на кнопку «Новый документ». При этом в выпадающем списке «Источник документа» в окне добавления вида документа необходимо выбрать «Результирующие документы» или «Шаблон печати заявления» в зависимости от вида выходного документа.

Файл шаблона документа категории «Результирующие документы» загружается в окне добавления вида документа по нажатию на поле «Загрузить шаблон», которое появляется при выборе данной категории в выпадающем списке (рисунок 168). Файл шаблона печати заявления загружается после добавления самого́ вида документа по нажатию на соответствующую кнопку («Загрузить шаблон») на вкладке карточки услуги «Документы» (рисунок 169).

Примечание. Добавляемые результирующие документы должны соответствовать предусмотренным для услуги административным процедурам, добавленным в услугу ранее (см. подраздел 3.9).

{{system.department.fullName}}

Кому:
{{applicant.fullName}}{{#applicant.address.postal.text}}
{{.}}{{/applicant.address.postal.text}}{{#applicant.inn}}
ИНН {{.}}
{{/applicant.inn}}{{^applicant.ogrnip}}{{#applicant.ogrn}}
ОГРН {{.}}
{{/applicant.ogrn}}{{/applicant.ogrnip}}{{#applicant.ogrnip}}
ОГРНИП {{.}}
{{/applicant.ogrnip}}{{#representative.fullName}}
Уполномоченное лицо: {{.}}
Контактные данные: {{#representative.phone}}
тел.
{{.}}{{/representative.phone}}{{#representative.email}}
эл почта
{{.}}{{/representative.email}}{{/representative.fullName}}{{^representative.fullName}}{{#delegate}}
Представитель: {{fullName}}
Контактные данные представителя: {{#phone}}
тел. {{.}}{{/phone}}{{#email}}
эл. почта
{{.}}{{/email}}{{/delegate}}{{^delegate}}
Контактные данные: {{#applicant.phone}}
тел.
{{.}}{{/applicant.phone}}{{#applicant.email}}
эл. почта
{{.}}{{/applicant.email}}{{/delegate}}{{/representative.fullName}}

РЕШЕНИЕ
об отказе в приеме документов

от {{resolutionDocumet.resDate}} № {{resolutionDocumet.resNumber}}

Рисунок 167 — Пример фрагмента шаблона результирующего документа

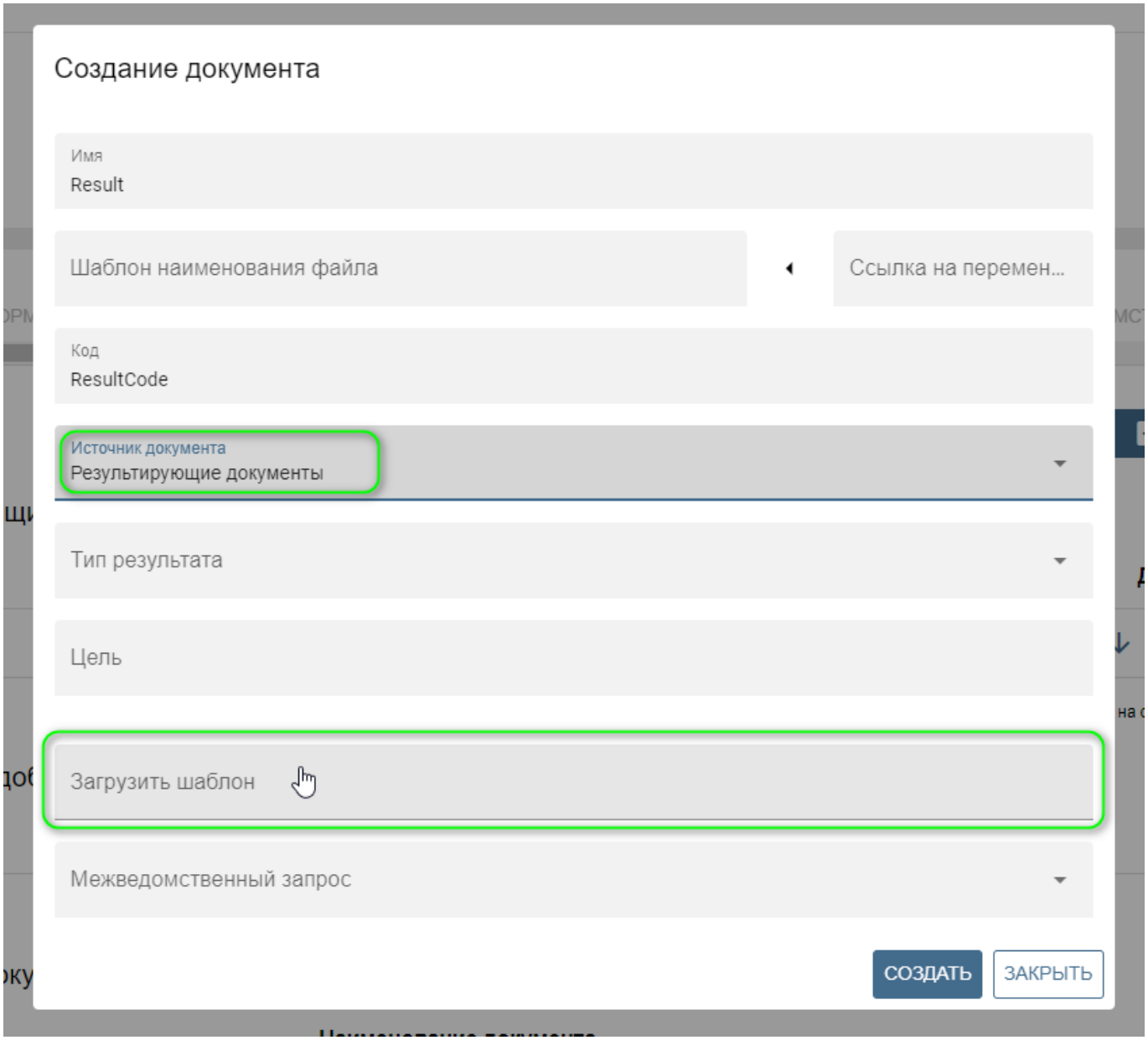


Рисунок 168 — Добавление файла шаблона документа категории «Результирующие документы»

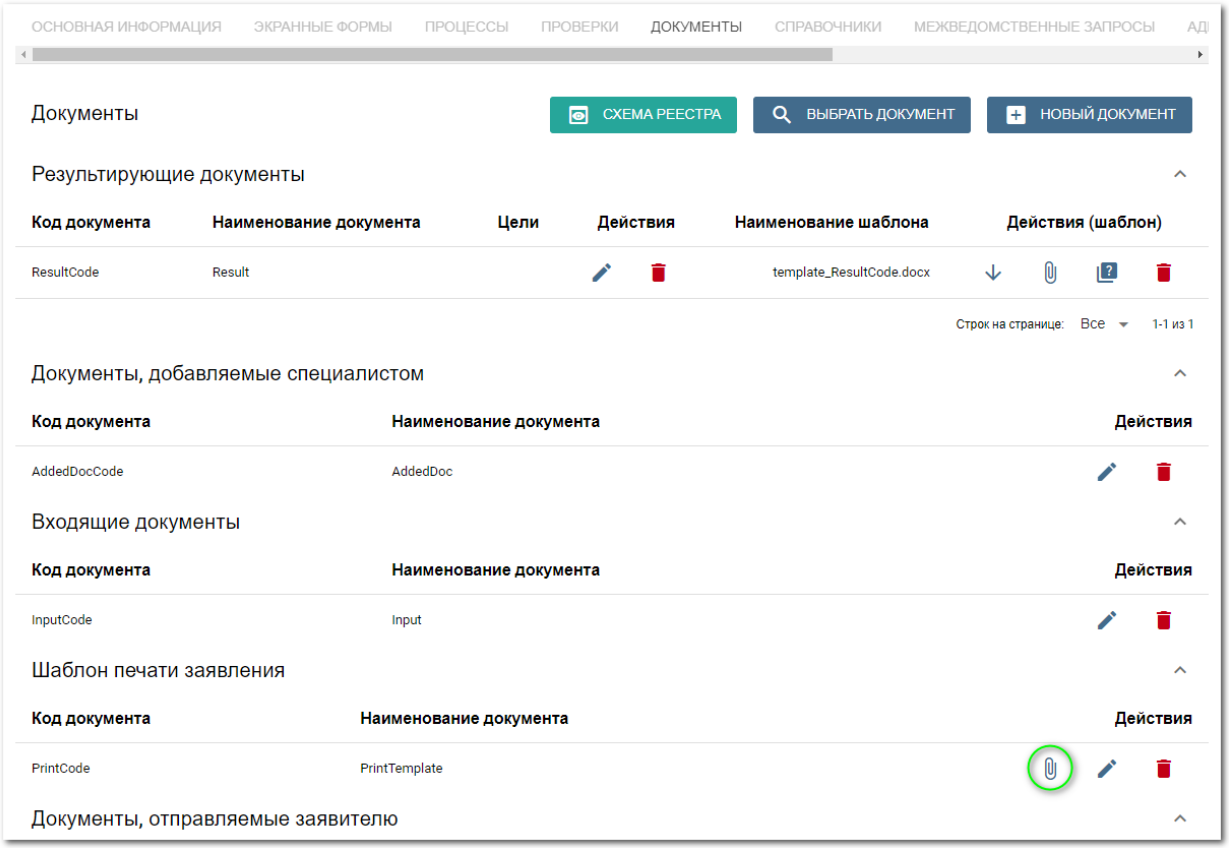


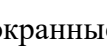
Рисунок 169 — Добавление файла шаблона печати заявления

3.16. Создание пустых экранных форм для результирующих документов

При добавлении в услугу шаблонов документов категории «Результирующие документы» для них автоматически генерируются соответствующие объекты «Экранная форма», которые доступны на вкладке карточки услуги «Экранные формы».

Экранные формы для результирующих документов используются для указания дополнительных сведений, которые будут внесены в соответствующие результирующие документы.

Сами экранные формы для результирующих документов автоматически не создаются, и их необходимо настроить вручную.

Сначала необходимо создать пустые экранные формы для документов, используя кнопки () («Создать пустую форму»), затем выполнить действия, описанные в подразделе 3.17, и после этого завершить настройку экранных форм (см. подраздел 3.18).

По умолчанию для каждого добавленного результирующего документа генерируется только один объект «Экранная форма». Если для документа требуется более одной экранной формы (например , в случае, когда услуга имеет несколько подуслуг), то с помощью кнопки + («Добавить еще одну схему формы»), относящейся к данному документу, можно добавить для него еще любое количество объектов «Экранная форма» и настроить для них экранные формы (рисунок 170).

Примечание. Кнопка для документа появляется только после того, как для него будет создана первая пустая экранная форма.

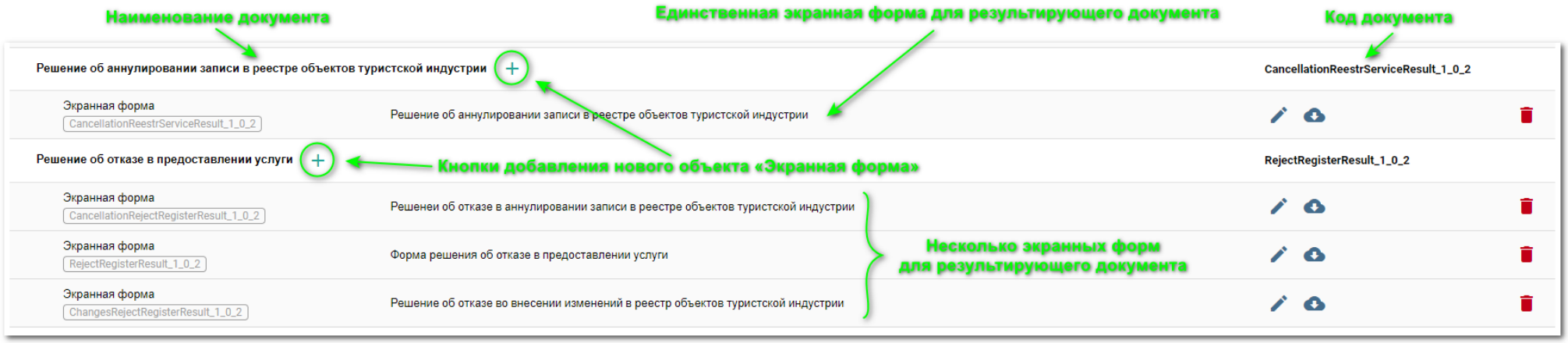


Рисунок 170 — Создание нескольких экранных форм для результирующего документа

3.17. Добавление результирующих документов и соответствующих им экранных форм в административные процедуры

После добавления в услугу шаблонов документов категории «Результирующие документы» необходимо добавить их и соответствующие им экранные формы в административные процедуры, ранее добавленные в услугу (см. подраздел 3.9), предусматривающие предоставление заявителю результирующих документов.

Для этого следует:

1. Перейти на вкладку карточки услуги «Административные процедуры».

2. Нажать кнопку («Редактировать»), соответствующую административной процедуре, в которую следует добавить документы;

3. В открывшемся окне редактирования административной процедуры нажать кнопку «Добавить документ» под заголовком «Результирующие документы» (рисунок 171).

4. После этого над кнопкой появятся четыре поля с выпадающими списками, показанные на рисунке 172.

5. В выпадающем списке «Документ» выбрать требуемый результирующий документ (рисунок 173).

6. В выпадающем списке «Статус» (рисунок 174) выбрать статус, соответствующий документу: положительный (в зависимости от административной процедуры, заявление принято или услуга предоставлена) или отрицательный (отказ в приеме документов или предоставлении услуги).

7. В выпадающем списке «Форма документа» (рисунок 175) выбрать экранную форму, соответствующую документу.

Примечание. В выпадающем списке можно выбрать только одну экранную форму, даже если в нем присутствуют более одной экранной формы. Для того, чтобы выбрать более одной экранной формы, необходимо добавить данный документ в административную процедуру еще

8. В выпадающем списке «Цель» (рисунок 176) выбрать цели (подуслуги), которым соответствуют выбранный документ и экранная форма.

9. При необходимости повторить шаги 3–7 для добавления других результирующих документов в административную процедуру или добавления того же документа с другими подуслугами и (или) экранной формой.

10. При необходимости повторить шаги 2–8 для других административных процедур.

11. Нажать кнопку «Сохранить» в окне редактирования административной процедуры.

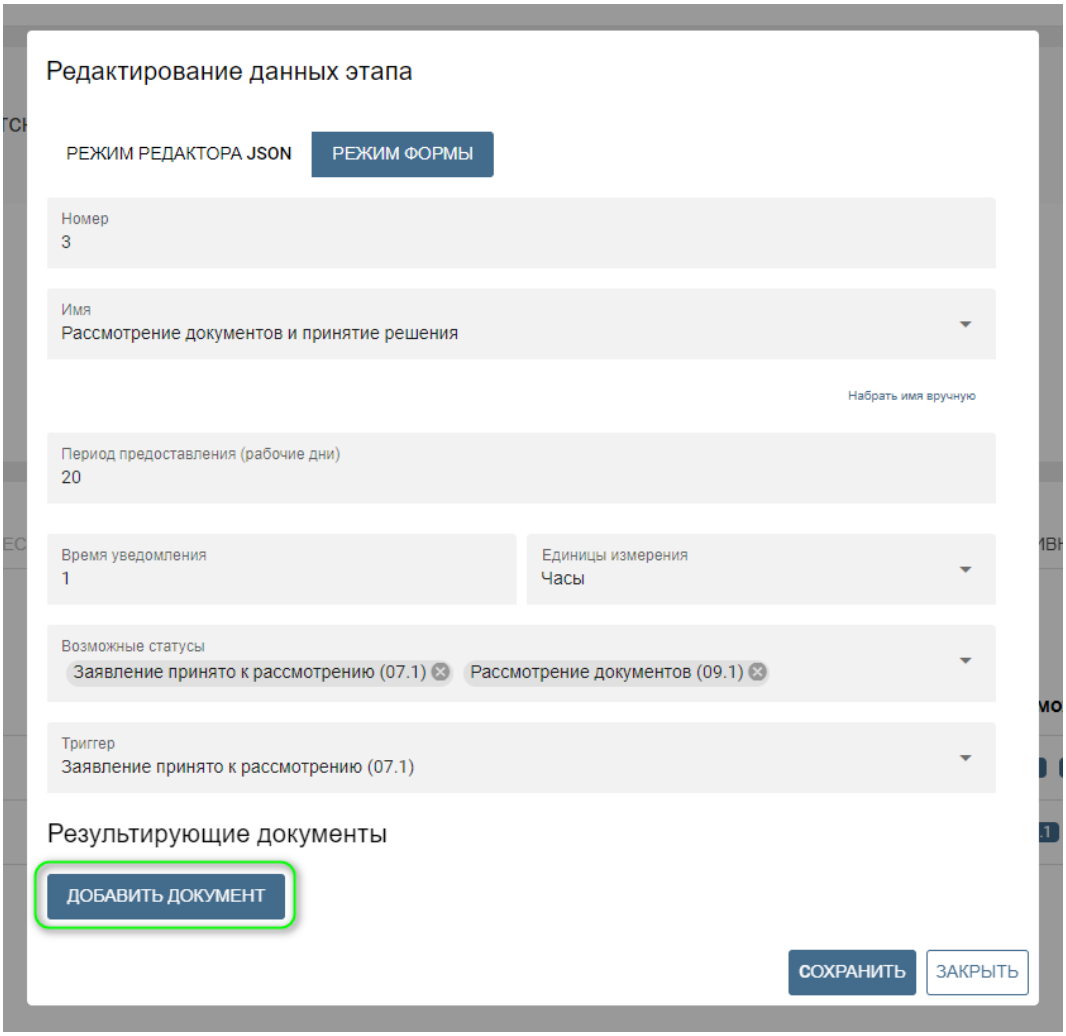


Рисунок 171 — Окно редактирования административной процедуры. Кнопка «Добавить документ»

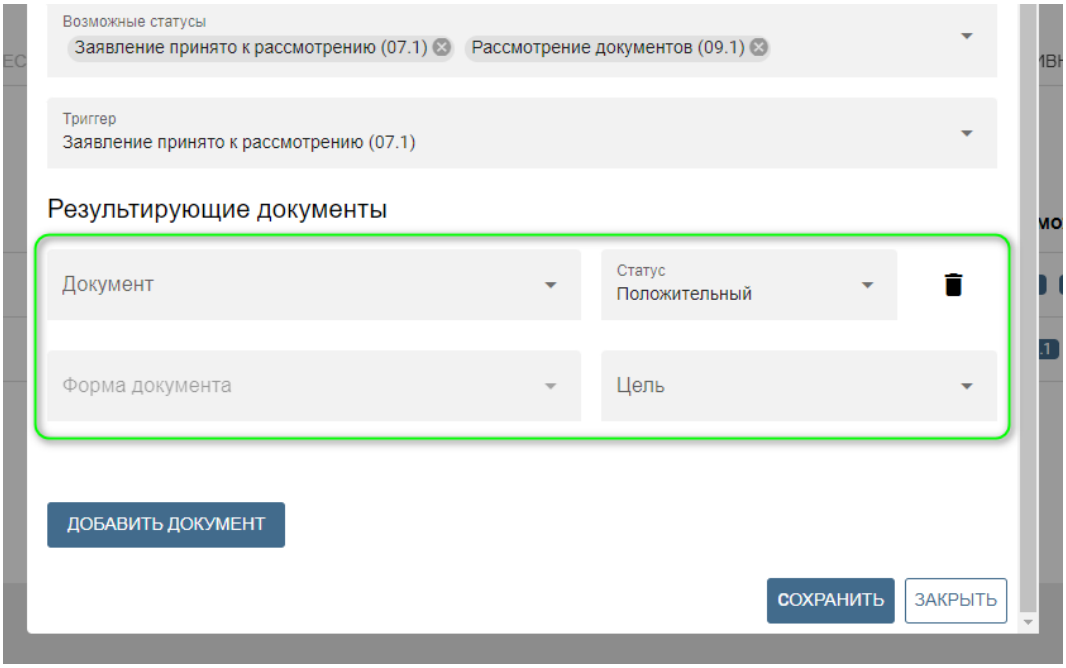


Рисунок 172 — Окно редактирования административной процедуры. Поля для добавления результирующего документа и соответствующей ему экранной формы с указанием статуса и подуслуг

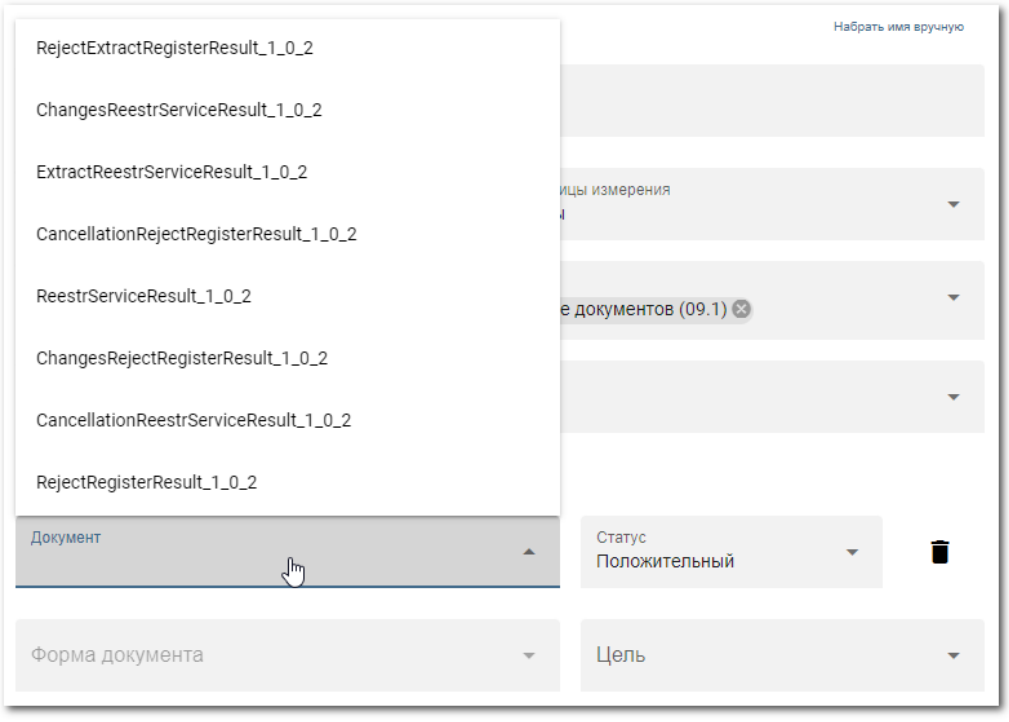


Рисунок 173 — Окно редактирования административной процедуры. Поле «Документ». Выпадающий список документов

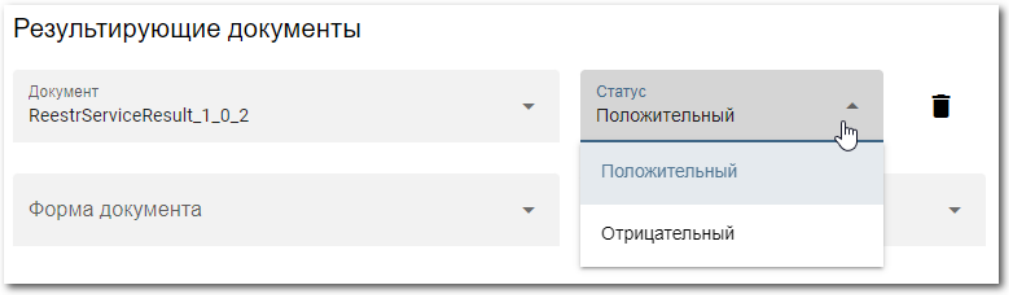


Рисунок 174 — Окно редактирования административной процедуры. Поле «Статус». Выпадающий список статусов для документа

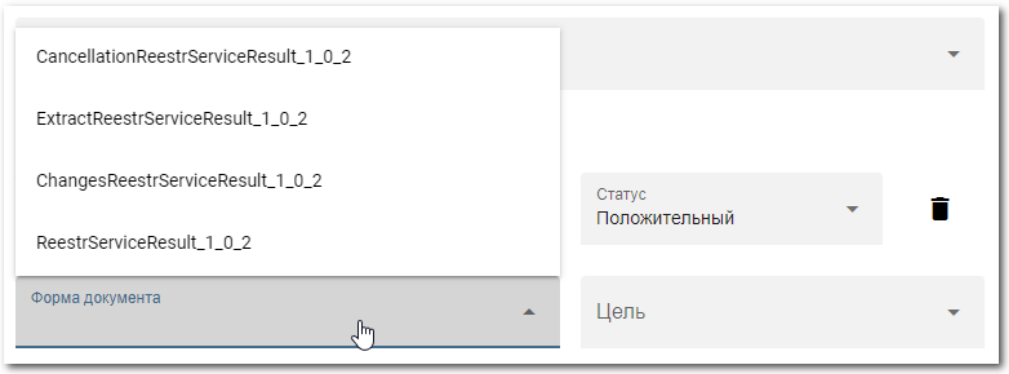


Рисунок 175 — Окно редактирования административной процедуры. Поле «Форма документа». Выпадающий список созданных экранных форм для документа

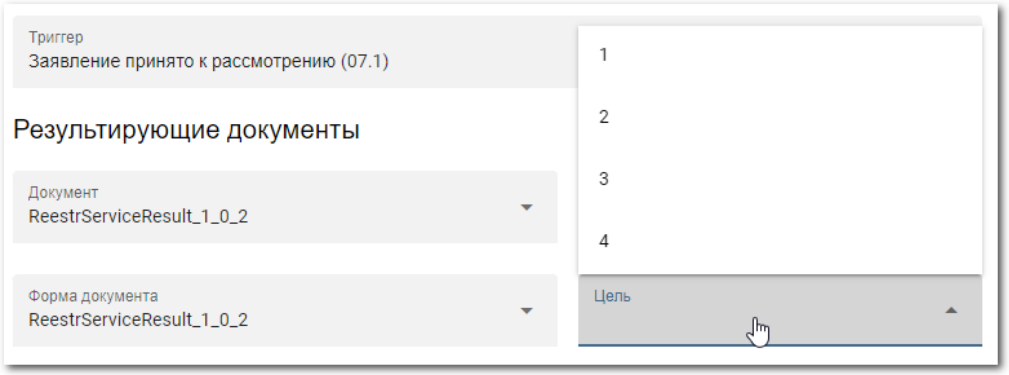


Рисунок 176 — Окно редактирования административной процедуры. Поле «Цель». Выпадающий целей (подуслуг) услуги

3.18. Донастройка экранных форм для результирующих документов

После выполнения действий, описанных в подразделах 3.16 и 3.17, необходимо завершить настройку экранных форм для результирующих документов посредством редактирования кода экранных форм на языке JSON в соответствии с общими принципами настройки экранных форм, изложенными в подразделе 3.14.

3.19. Реестры решений

Для услуги могут создаваться реестры, предназначенные для хранения решений по предоставлению услуги.

Реестры создаются отдельно от услуги в разделе ПГС «Реестры». На вкладке карточки услуги «Документы» могут быть выбраны реестры ПГС, в которых должны храниться решения по предоставлению услуги.

Для того чтобы создать реестр решений, необходимо:

1. В браузере перейти по URL-адресу ПГС (если пользователь уже находится в интерфейсе ПГС, пропустить данный шаг).

2. Авторизоваться в ПГС с ролью администратора услуг или оператора реестров (см подраздел 3.1; если пользователь уже авторизован, пропустить данный шаг).

3. Перейти в раздел «Реестры».

4. На открывшейся странице нажать кнопку «Создать» (рисунок 177).

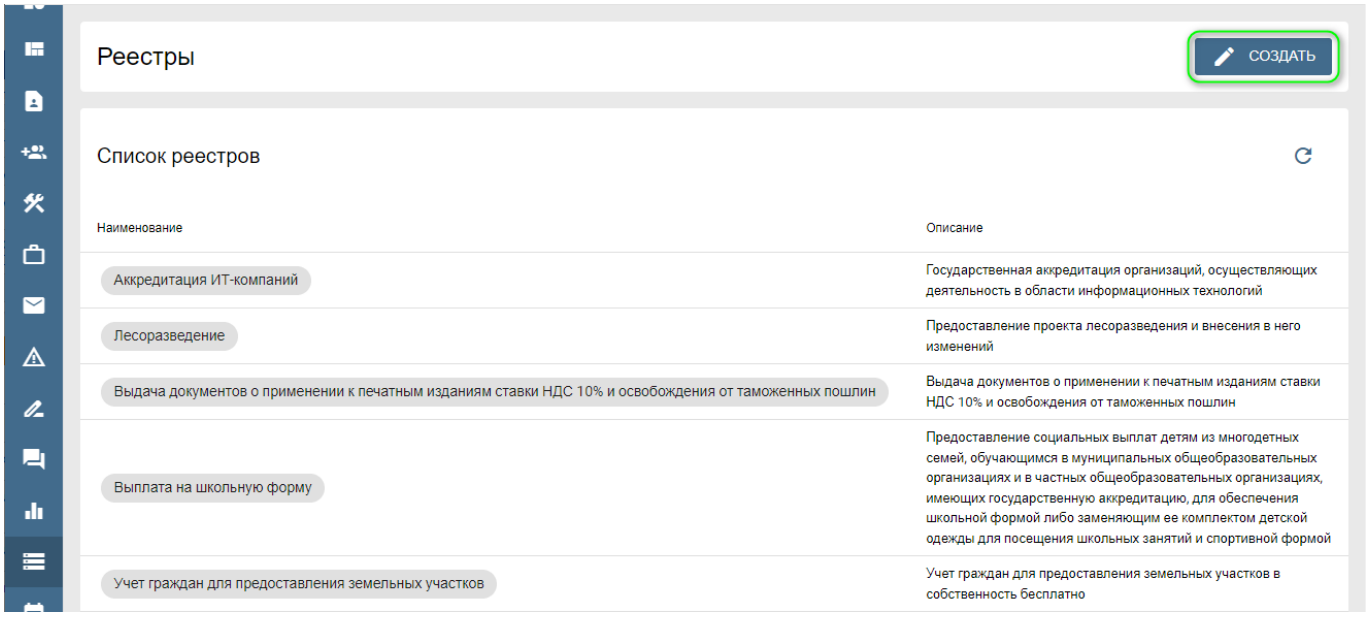


Рисунок 177 — Главная страница раздела ПГС «Реестры». Список реестров, кнопка «Создать»

5. В открывшемся окне (рисунок 178):

– указать:

а) наименование реестра, позволяющее пользователям идентифицировать и находить его в ПГС, — в поле «Название»;

б) описание реестра, сведений, которые он должен содержать, — в поле «Описание»;

в) код реестра (уникальное среди всех реестров в ПГС значение, состоящее из латинских букв, цифр и символов) — в поле «Код»;

– установить отметку «Активность» (если отметка не установлена, реестр будет создан, но будет неактивен);

– снять отметку «Реестр по услугам социального казначейства»;

– выбрать услугу, которой должен соответствовать создаваемый реестр, используя поиск по наименованию и выпадающий список в поле «Услуга» (рисунок 179). Не следует выбирать более одной услуги. Настройка выбираемой услуги необязательно должна быть завершена. После этого под заголовком «Документы» в окне создания реестра автоматически появятся *результирующие* документы, которые были добавлены на вкладке карточки данной услуги «Документы» (рисунок 180);

– посредством установки отметок выбрать документы, соответствующие тем решениям, которые требуется хранить в реестре (рисунок 180). Для большинства услуг необходимо хранить все решения;

– нажать на ссылку «Добавить кастомную форму» и в появившемся поле настроить экранную форму для отображения записей реестра посредством редактирования ее JSON-кода в соответствии с общими принципами настройки экранных форм, изложенными в подразделе 3.14 (по умолчанию в поле присутствует код предварительно настроенной экранной формы, сгенерированный автоматически в соответствии с выбранными документами). Код экранной формы, в частности должен содержать соответствие между полями записи реестра решений и атрибутами в описании (коде) заявления о предоставлении услуги в ПГС;

– нажать кнопку «Создать».

Примечание. Поля «Связанные реестры», «Территориальный уровень» заполнять не требуется.

Примечание. В карточке услуги на вкладке «Документы» по нажатию на кнопку «Схема реестра» также доступна настройка экранной формы для отображения записей реестра. Код настроенной экранной формы необходимо перенести в реестр с помощью буфера обмена. Доступна также функция сохранения кода экранной формы в файл.

После этого реестр будет создан и появится в списке реестров в разделе «Реестры».

Просмотр записей реестра осуществляется в карточке реестра (рисунок 181), для перехода к которой необходимо нажать на наименование реестра в списке реестров.

Кнопка «Редактировать реестр» (рисунок 181) позволяет изменить основные параметры реестра. При ее нажатии открывается окно, аналогичное окну создания реестра.

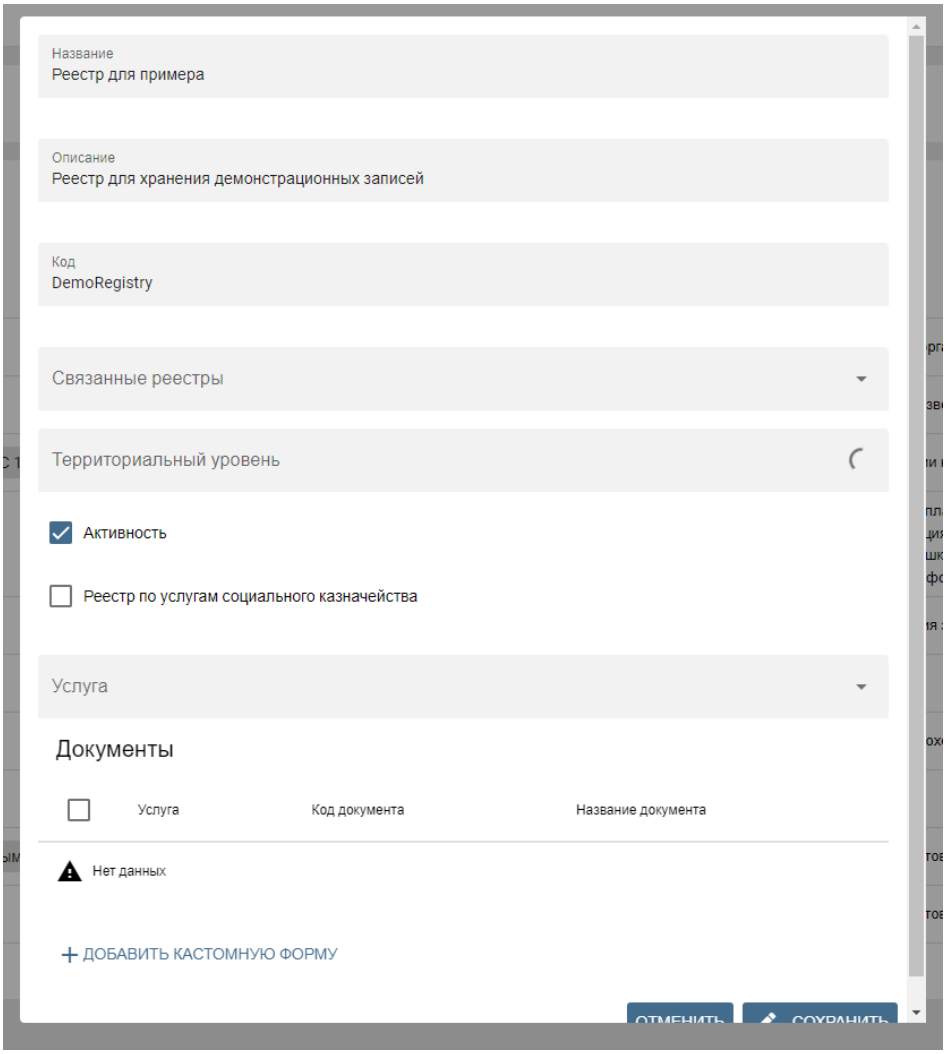


Рисунок 178 — Окно создания реестра

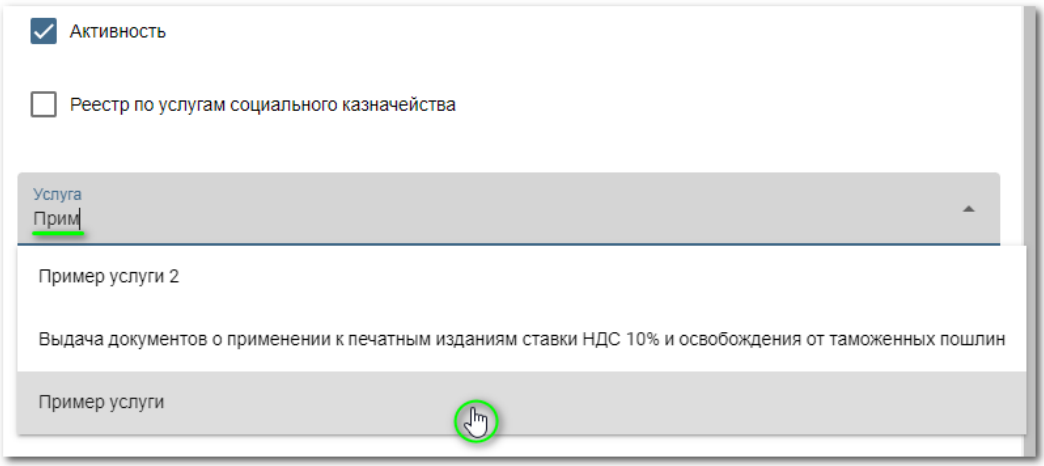


Рисунок 179 — Выбор услуги для привязки к реестру
с использованием функции поиска наименованию и выпадающего списка услуг

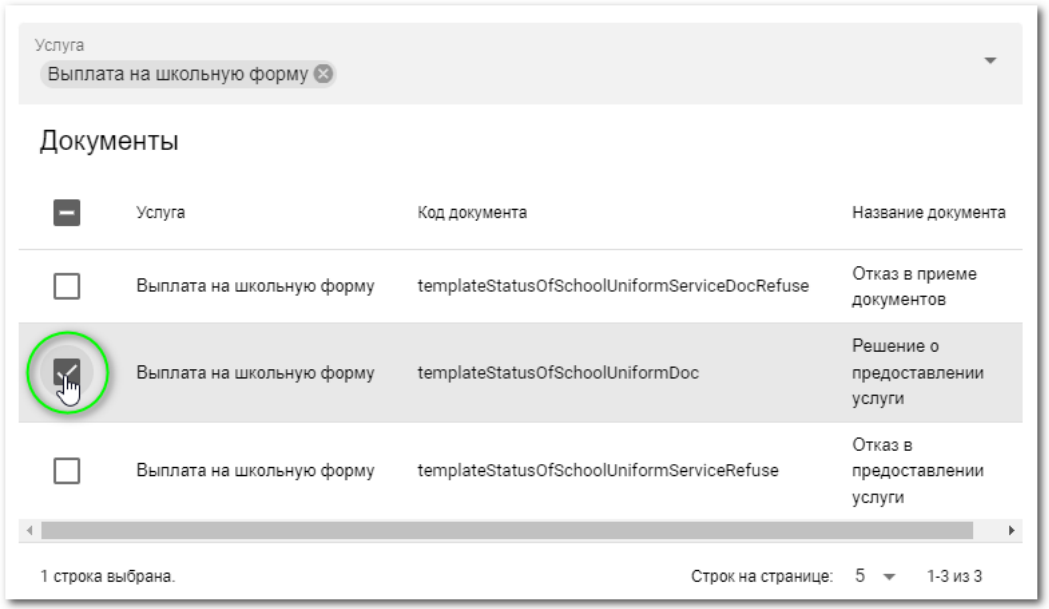


Рисунок 180 — Результирующие документы услуги в окне создания реестра. Выбор документов

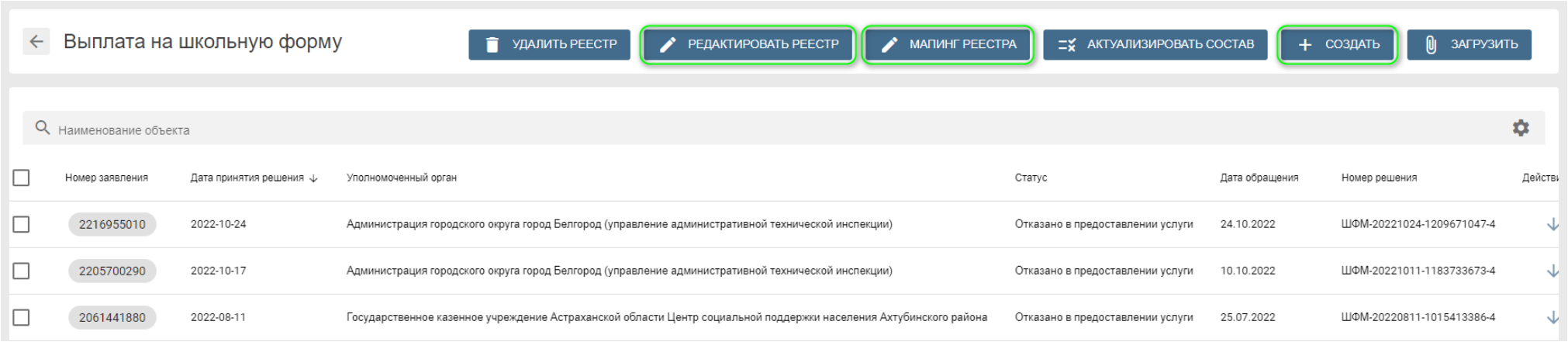


Рисунок 181 — Карточка реестра решений. Список записей реестра и кнопки управления реестром

Записи реестра отображаются в карточке реестра списком. Список записей поделен на столбцы, в которых выводятся данные, содержащиеся в соответствующих полях записей реестра. По умолчанию список записей о решениях по заявлениям о предоставлении любой услуги включает следующие столбцы (рисунок 181):

– «Номер заявления»;

– «Дата принятия решения»;

– «Уполномоченный орган»;

– «Статус»;

– «Дата обращения»;

– «Номер решения»;

– «Действия».

Просмотреть и изменить таблицу соответствия между столбцами списка записей реестра решений и атрибутами в описании (коде) заявления в ПГС на предоставление услуги можно, нажав кнопку «Мапинг реестра». При этом открывается окно, представленное на рисунке 182.

Кнопка «Добавить строку» позволяет добавить новую строку, соответствующую новому столбцу списка записей, в таблицу соответствия; кнопка «Удалить» позволяет удалить соответствующую строку (столбец списка). Посредством снятия отметки в той или иной строке можно исключить из списка записей соответствующий столбец без удаления строки из таблицы сответствия. Посредством редактирования значений в строках можно изменить соответствие между столбцами списка и атрибутами в описании (коде) заявления в ПГС.

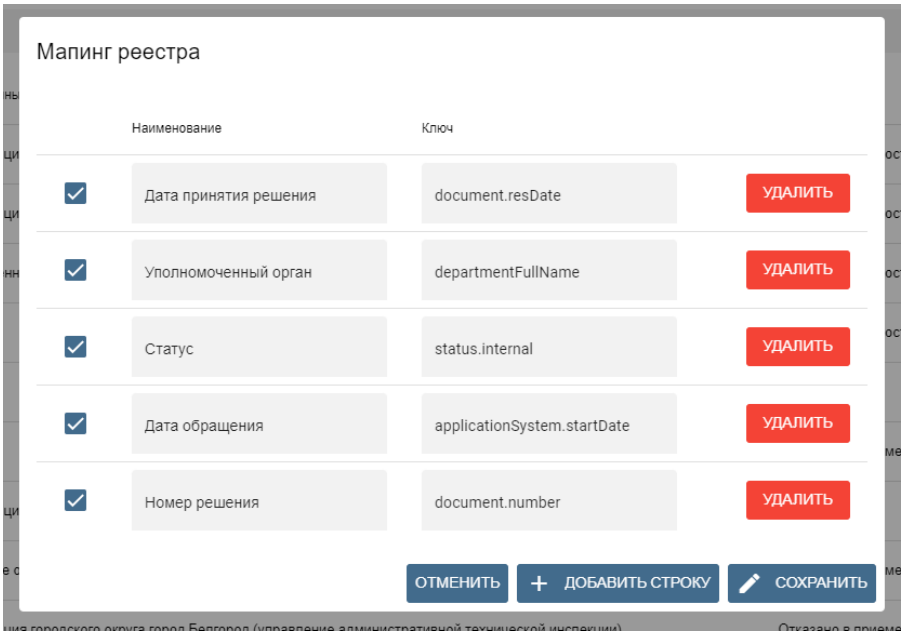


Рисунок 182 — Окно редактирования таблицы соответствия между столбцами списка записей реестра решений и атрибутами в описании (коде) заявления о предоставлении услуги в ПГС

Если вышеописанные действия выполнены корректно, по завершении процесса предоставления услуги вынесенное решение будет автоматически заноситься в реестр, к которому привязана услуга.

Записи реестра могут также создаваться вручную по нажатию на кнопку «Создать» в карточке реестра. Если планируется использование данной функции, то необходимо в коде экранной формы корректным образом настроить поля для ввода данных при ручном создании записи реестра.

3.20. Настройка экранной формы очного приема

Настройка экранной формы очного приема необходима для тех услуг, которые допускают очное обращение заявителей или иных лиц, представляющих интересы заявителей, в ведомство (организацию), предоставляющее услугу, с запросом о предоставлении данной услуги.

Для того чтобы услуга была доступна для выбора при ручном создании заявления о ее предоставлении в ПГС, необходимо, чтобы:

– на вкладке карточки услуги «Основная информация» был установлен признак «Добавить форму заявления в очный прием» (см. подраздел 3.3);

– услуга была включена в список услуг, предоставляемых ведомством (организацией), в котором состоит пользователь (для перехода к списку необходимо перейти в раздел

ПГС «Организации», открыть карточку нужной организации и перейти на вкладку «Услуги» — рисунок 183).

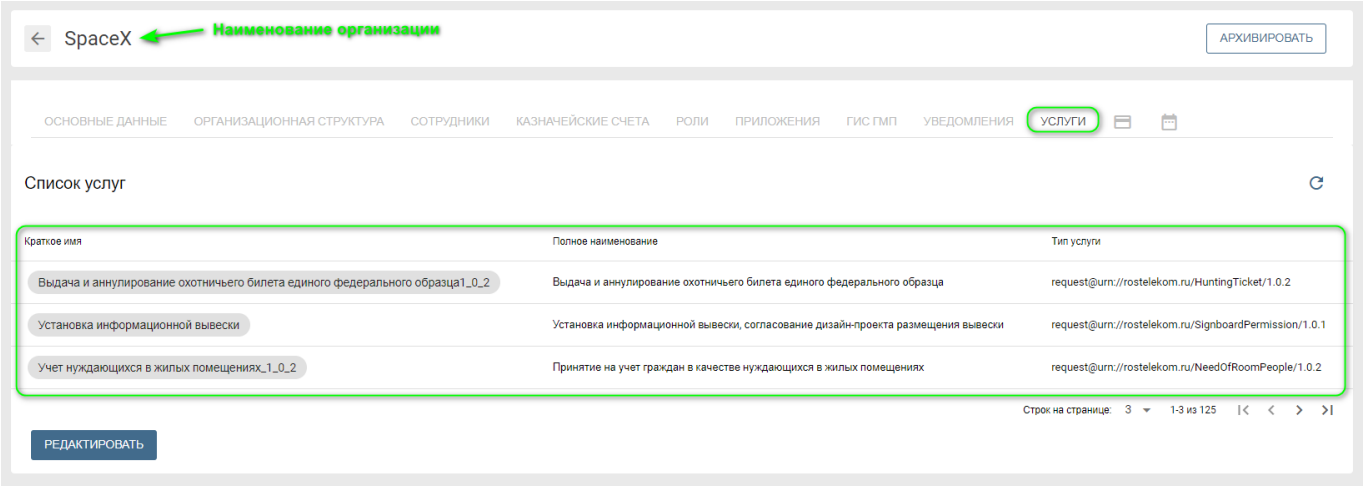


Рисунок 183 — Раздел ПГС «Организации».
Список услуг на вкладке «Услуги» карточки организации

3.20.1. Создание и кодирование экранной формы очного приема

Настройка экранной формы очного приема осуществляется на вкладке карточки услуги «Экранные формы» в группе «Очный прием» (рисунок 184) в соответствии с общими принципами настройки экранных форм, изложенными в подразделе 3.14. Объект «Экранная форма», не содержащий саму экранную форму, создается для экранной формы очного приема автоматически при создании услуги (подраздел 3.2).

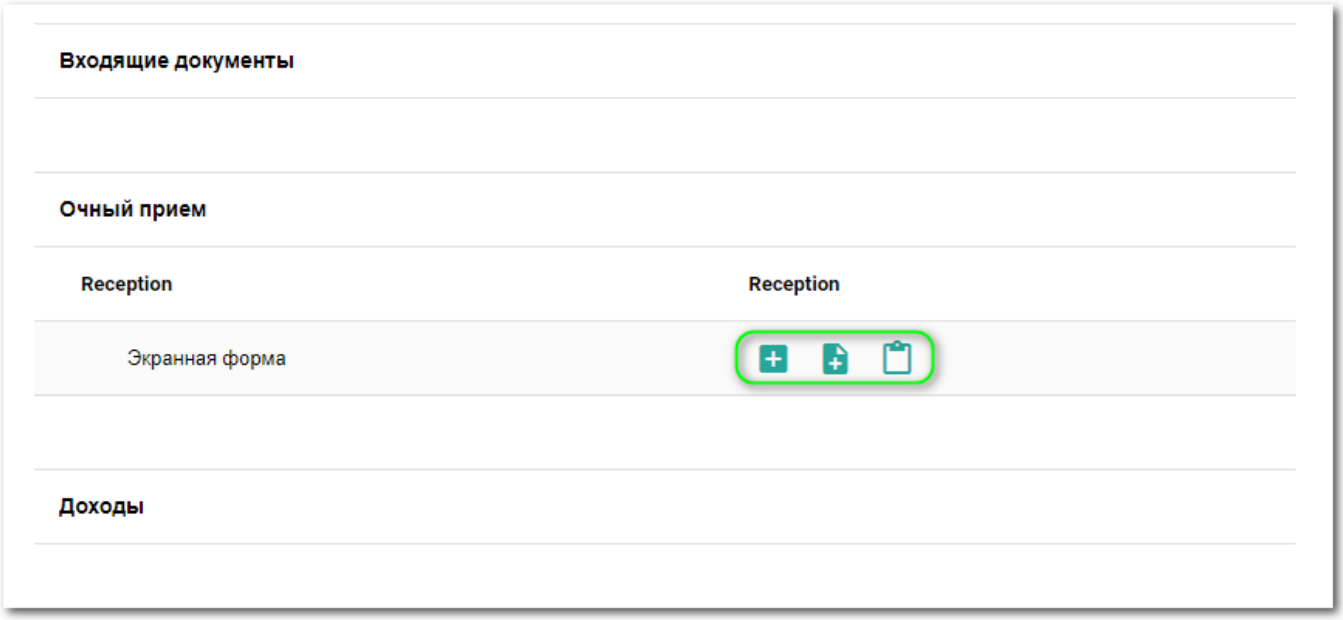


Рисунок 184 — Объект «Экранная форма» для экранной формы очного приема на вкладке карточки услуги «Экранные формы» (сама экранная форма не создана)

Свойство name в коде экранной формы очного приема, задающее наименование экранной формы, обязательно должно иметь значение ReceptionBase, поскольку именно по этому признаку ПГС обнаруживает присутствие данной экранной формы и определяет ее как экранную форму очного приема.

JSON-схема экранной формы очного приема должна основываться на модели (коде) заявления в ПГС, которая получается после импорта.

Наиболее простым способом получить модель заявления является отправка в ПГС какого-либо заявления (желательно, максимально заполненного), имитирующего заявление с ЕПГУ, либо тестового двойника такого заявления. В ПГС следует открыть его через историю его requestJson.

Каждый атрибут в модели (коде) заявления в ПГС должен быть занесен в схему очного приема (если он требуется в заявлении, формируемом при очном приеме) с соблюдением полного пути до атрибута.

Как и в любой другой схеме представления (заявления, МВЗ), в схеме экранной формы очного приема отдельно настраиваются атрибуты (attrList), группы для этих атрибутов (groupList), вложенные схемы (linkedSchema), а также extensions.

3.20.1.1 Описание extensions

Расширенные настройки для элементов схемы, когда отображение атрибута требуется при определенных условиях, задаются при помощи extensions.

В схеме экранной формы extensions задаются внутри блока extensions.settings.

Настройка состоит из:

– actions — описывает необходимые элементы (атрибуты или группы) и необходимые действия с ними (отобразить, скопировать, настроить маску и т. д.);

– conditions — описывает при выполнении каких условий. Условия могут быть как простыми, так и сложными.

Actions могут применяться к списку отдельных атрибутов (attrs) и/или к группе (groups), которые заданы ранее в блоках attrList и groupList.

В параметре type задается тип действия. На текущий момент реализована поддержка следующих действий:

– show — показывает/скрывает поле экранной формы;

– enable — включает режим readonly для поля;

– mandatory — переключает обязательность поля;

– copy — копирует из одного место в другое / очищает значения;

– setProps — устанавливает/удаляет свойства props указанным атрибутам;

– setRules — устанавливает/удаляет правила rules указанным атрибутам.

В Conditions задается тип условия (type), атрибут значение которого будет проверяться (value.attr.codeAttr), блок описывающий операнд и значение (validators). Условия могут быть односложными («type»: «simple») и сложными («type»: «complex»).

Далее приведен пример сложного условия.

```
«conditions»: [ // Пример настройки со сложным условием. Присутствуют два условия. Настройка сработает, если одно из условий в массиве conds вернет true. Причем одно из этих условий также сложное и имеет два условия в себе.
        {
          «type»: «complex», //
          «operator»: «||», // Одно из условий должно быть верным
          “conds”: [
            {
              “type”: “simple”,
              “value”: {
                “attr”: {
                  “codeAttr”: “pRelation”
                },
                “validators”: [
                  {
                    “type”: “cond”,
                    «operator»: «==»,
                    «value»: 1
                  }
                ]
              }
            },
            {
              «type»: «complex», // второе сложное условие, но в этот раз должны быть выполнены оба условия внутри
              “operator”: “&&”,
              “conds”: [
                {
                  “type”: “simple”,
                  “value”: {
                    “attr”: {
                      “codeAttr”: “pRelation”
                    },
                    “validators”: [
                      {
                        “type”: “cond”,
```

```
                        "operator": "==",
                        "value": 2
                      }
                    ]
                  }
                },
                {
                  "type": "simple",
                  "value": {
                    "attr": {
                      "codeAttr": "Bithday"
                    },
                    "validators": [
                      {
                        "type": "cond",
                        "operator": "IS NOT NULL"
                      },
                      {
                        "type": "cond",
                        «operator»: «date<=relative», // раньше или равна дате,
вычисленной относительно сегодняшнего дня
                        «value»: [
                          [«subtract», 14, «years»] // в этом случае вернется
true, если сравниваемая дата была ранее чем 14 лет назад
                        ] }]}}]}}
      ]
```

3.20.1.1.1 Примеры реализации extensions

3.20.1.1.1.1 Отображение/скрытие элементов. Тип show

В данном примере атрибуты и группа delegate будут видны, только если у атрибута DelegateInfo будет значение «2». (Поля для заполнения данных о представителе будут отображаться, если в поле «Категория заявителя» будет выбрано значение из справочника «Представитель заявителя».)

3.20.1.1.1.2 На задание определенных свойств. Тип setProps

В данном примере, для двух атрибутов (applicantInfo.identityDocInfo.number и applicantInfo.identityDocInfo.series) задаются индивидуальные свойства (props), если в качестве типа документа, удостоверяющего личность, заявителя (applicantInfo.identityDocInfo.type.code) выбран из справочника код «03».

```
{
  "id": "applicant_soldier",
  "actions": [
    {
      "type": "setProps",
      "attrs": [
        "applicantInfo.identityDocInfo.number"
      ],
      "props": {
        "maxLength": 7,
        "minLength": 6
      }
    },
    {
      "type": "setProps",
      "attrs": [
        "applicantInfo.identityDocInfo.series"
      ],
      "props": {
        "maxLength": 2,
        "minLength": 2
      }
    }
  ],
  "conditions": [
    {
      "type": "simple",
      "value": {
        "attr": {
          "codeAttr": "applicantInfo.identityDocInfo.type.code",
          "title": "TODO attr title"
        },
        "validators": [
          {
            "type": "cond",
            "operator": "==",
            "value": "03"
          }
        ]
      }
    }
  ]
},
```

3.20.1.1.1.3 Копирование значения атрибута. Тип copy

В данном примере происходит копирование значения одного атрибута в другой. Здесь задается параметр «isDirect»: true (действие выполняется вне зависимости от указанных атрибутов, позволяет пропустить выборку по полям из attrs и groups), внутри «copyDirections» задается «from» (из какого поля модели копировать) и «to» (в какое поле модели копировать).

```
{
  "id": "copyname",
  "actions": [
    {
      "type": "copy",
      "isDirect": true,
      "copyDirections": [
        {
          "from": "RecipientPersonalData.DocumentPersonal.typeDoc",
          "to": "RecipientPersonalData.DocumentPersonal.nameDoc"
        }
      ]
    }
  ],
  "conditions": [
    {
      "type": "simple",
      "value": {
        "attr": {
          "codeAttr": "RecipientPersonalData.DocumentPersonal.typeDoc",
          "title": "TODO attr title"
        },
        "validators": [
          {
            "type": "cond",
            "operator": "!=",
            "value": "1"
          }
        ]
      }
    }
  ]
},
```

3.20.1.1.2 Доступные типы операндов для работы с условиями

Доступные типы операндов для работы с условиями приведены ниже.

```
MORE = '>',                                  // БОЛЬШЕ
MORE_OR_EQUAL = '>=',                        // БОЛЬШЕ ИЛИ РАВНО
DATE_MORE = 'date>',                         // ДАТА БОЛЬШЕ
DATE_MORE_OR_EQUAL = 'date>=',               // ДАТА БОЛЬШЕ ИЛИ РАВНО
DATE_MORE_RELATIVE = 'date>relative',        // ДАТА БОЛЬШЕ ОТНОСИТЕЛЬНО ТЕКУЩЕЙ
DATE_MORE_OR_EQUAL_RELATIVE = 'date>=relative',  // ДАТА БОЛЬШЕ ИЛИ РАВНО ОТНОСИТЕЛЬНО ТЕКУЩЕЙ
DATE_LESS = 'date<',                         // ДАТА МЕНЬШЕ
DATE_LESS_OR_EQUAL = 'date<=',               // ДАТА МЕНЬШЕ ИЛИ РАВНО
DATE_LESS_RELATIVE = 'date<relative',        // ДАТА МЕНЬШЕ ОТНОСИТЕЛЬНО ТЕКУЩЕЙ
DATE_LESS_OR_EQUAL_RELATIVE = 'date<=relative',  // ДАТА МЕНЬШЕ ИЛИ РАВНО ОТНОСИТЕЛЬНО ТЕКУЩЕЙ
LESS = '<',                                  // МЕНЬШЕ
LESS_OR_EQUAL = '<=',                        // МЕНЬШЕ ИЛИ РАВНО
NOT_EQUAL = '!=',                            // НЕ РАВНО
IS_NOT_NULL = 'IS NOT NULL',                 // НЕ РАВНО NULL
NOT_LIKE = 'NOT LIKE',                       // НЕ СОДЕРЖИТ
EQUAL = '==',                                // РАВНО
IS_NULL = 'IS NULL',                         // РАВНО NULL,
LIKE = 'LIKE'                                // СОДЕРЖИТ,
}
```

Например, для календаря можно настроить ограничение и зависимость дат:

– на ненулевое значение («operator»: «IS NOT NULL»);

– на дату меньше которой не даст выбрать от текущей («operator»: «date<=relative»);

– на дату, больше которой не даст выбрать от текущей («operator»: «date>=relative») и др.

Ниже приведен пример ограничения дат в диапазоне 8–17 лет относительно текущей не пустой даты. Даты, сравниваемые с текущей, должны указываться в формате ГГГГ-ММ-ДД.

3.20.1.2 Настройка отдельных элементов схемы

3.20.1.2.1 Настройка вложенных схем

Для работы с массивами используются вложенные схемы — linkedSchema.

Атрибут, который представляет собой массив, должен содержать ссылку на linkedSchema (reference) и ее название (reference.entity) — рисунок 185.

Рисунок 185 — Массив, содержащий ссылку на linkedSchema,
в коде экранной формы очного приема

Внутри linkedSchema.linked указывается ее название, заданное ранее в reference.entity.

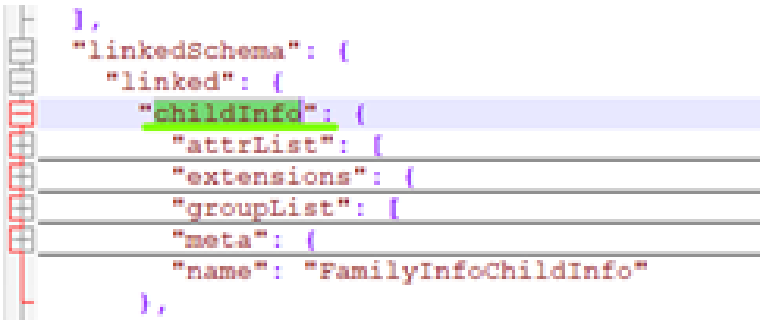


Рисунок 186 — linkedSchema.linked в коде экранной формы очного приема

Внутри linkedSchema действуют такие же правила, как и внутри обычной схемы (задаются атрибуты, группы, extensions), но нет необходимости указывать полные пути до элементов — нужно указывать только пути внутри массива.

Элементы массива в код экранной формы очного приема можно добавлять, используя кнопку «Добавить». Она будет доступна при указании в атрибуте шаблона «template»: «tableFill», а также в случае настроенной linkedSchema (рисунок 187).

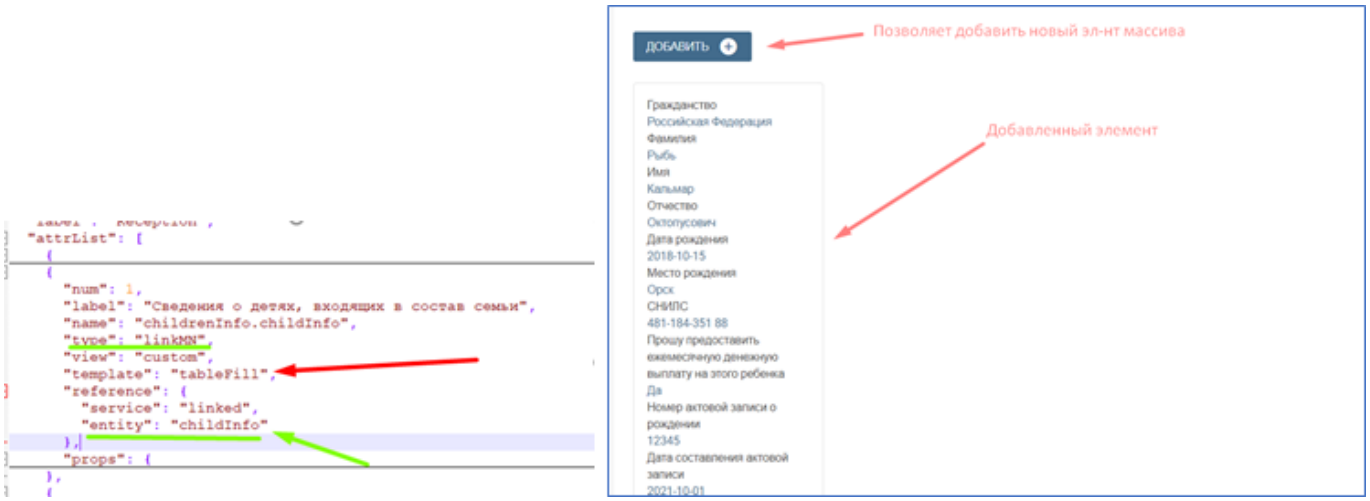


Рисунок 187 — Добавление элементов массива в код экранной формы очного приема с использованием кнопки «Добавить». Шаблон "template": "tableFill" внутри атрибута в коде экранной формы, при наличии которого доступна кнопка «Добавить»

3.20.1.2.2 Настройка таблицы соответствия для справочника

3.20.1.2.2.1 Пример для атрибута «Тип документа»

В модели (коде) заявления в ПГС атрибут «Тип документа» выглядит, как показано на рисунке 188 (поле type с полями code и content внутри).

Рисунок 188 — Атрибут «Тип документа» в модели (коде) заявления в ПГС

В форме очного для него должно присутствовать одно поле, в котором специалист, осуществляющий прием заявителя, в выпадающем списке выбирает значение по названию (рисунок 189).

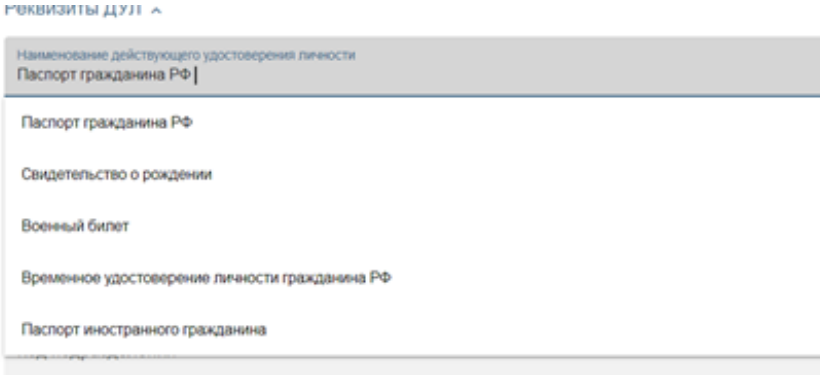


Рисунок 189 — Выпадающий список в поле экранной формы создания заявления при очном приеме заявителя

Для этого в атрибуте «Тип документа» задается шаблон "template": "MappedDictionary". В props задаются код справочника (props.dictionaryId) и соответствие между полями справочника и полями в модели заявления (props.mapFields.code в коде экранной формы соответствует полю applicantInfo.identityDocInfo.type.code в модели заявления, и props.mapFields.value в коде экранной формы соответствует полю applicantInfo.identityDocInfo.type.content в модели заявления) — рисунок 190.

```
},
{
  "name": "applicantInfo.identityDocInfo.type",
  "label": "Тип документа",
  "num": 0,
  "type": "string",
  "view": "custom",
  "template": "MappedDictionary",
  "props": {
    "dictionaryId": "familychildsub_manual_doc_type",
    "mapFields": {
      "code": "code",
      "value": "content"
    }
  },
  "createRequired": false,
  "editRequired": false
},
{
```

Рисунок 190 — Настройка атрибута «Тип документа» коде экранной формы очного приема

3.20.1.2.2.2 Пример для адреса ФИАС

В данном примере таблица соответствия настраивается аналогично примеру выше, за исключением шаблона "template": "fiasAddress", а также того, что в названии атрибута должен присутствовать идентификатор (Id) адреса ФИАС, несмотря на то что в модели (коде) заявления в ПГС он отсутствует (ср. рисунки 191 и 192).

Рисунок 191 — Шаблон "template": " fiasAddress " и идентификатор Id внутри атрибута, отвечающего за адрес ФИАС, в коде экранной формы очного приема

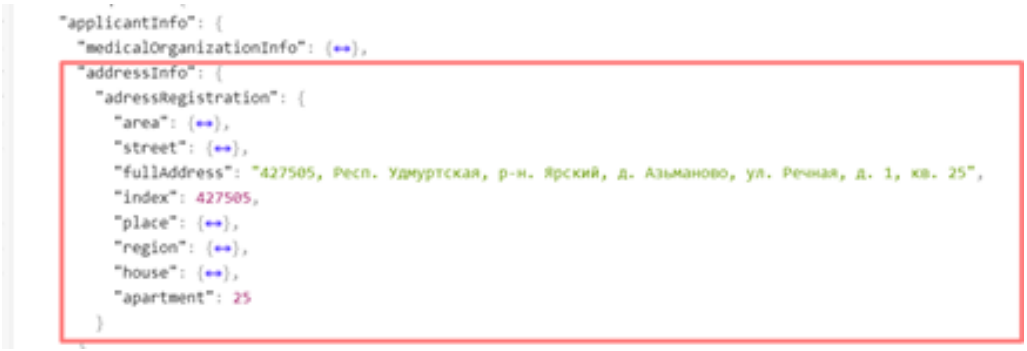


Рисунок 192 — Атрибут, отвечающий за адрес ФИАС, в модели (коде) заявления в ПГС

3.20.2. Документы, прикрепляемые к заявлению, формируемому при очном приеме

В заявлениях, формируемых при очном приеме, всегда присутствует, содержащий документы, прикрепленные к услуге (на вкладке карточки услуги «Документы»), категории «Входящие документы» (см. подраздел 3.8).

Блок добавляется в заявления автоматически; никаких действий от специалиста, настраивающего экранную форму очного приема, при этом не требуется. Возможность отключения присутствия данного блока в заявлениях не предусмотрена.

3.21. Настройка криптопровайдера КриптоПро

Для проверки работоспособности настроенных процессов предоставления услуг и для работы с функциональностью подписания с помощью ЭП-СП файлов — результатов (решений) необходимо на АРМ специалиста установить и настроить криптопровайдер КриптоПро, Cades-плагин для веб-браузера и сертификат ЭП.

Установка криптопровайдера КриптоПро CSP версии 4.0 или выше должна производиться в соответствии с инструкцией, расположенной на общедоступном ресурсе в сети Интернет по адресу https://www.cryptopro.ru/sites/default/files/docs/instruction_manual_csp_r3.pdf.

После установки КриптоПро CSP необходимо установить CAdES-плагин для веб-браузера (https://www.cryptopro.ru/products/cades/plugin — рисунок 193) в соответствии с инструкцией по установке, расположенной по адресу https://docs.cryptopro.ru/cades/plugin/plugin-installation-windows.

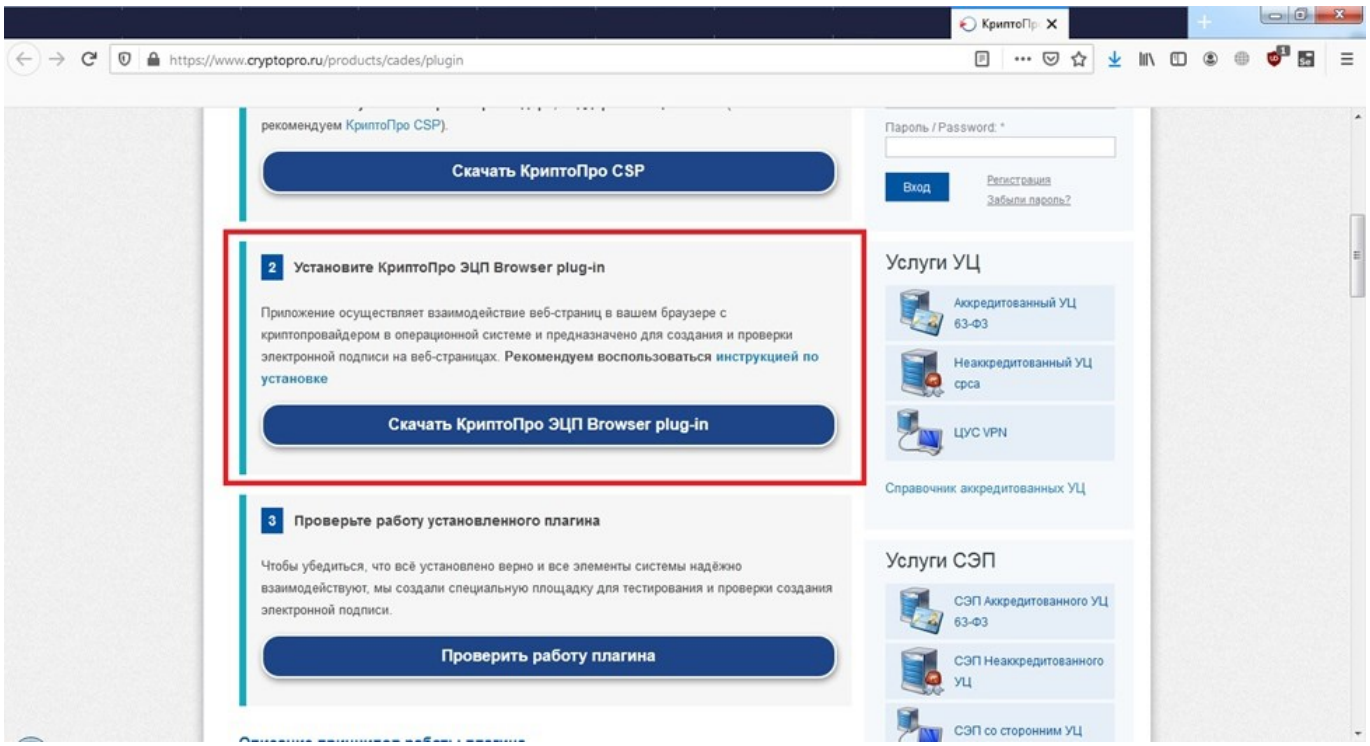


Рисунок 193 — Установка CAdES-плагина для веб-браузера

Для выпуска тестового[1] сертификата ЭП необходимо перейти на сайт тестового удостоверяющего центра КриптоПро по адресу https://www.cryptopro.ru/certsrv , далее перейти по ссылке «Сформировать ключи и отправить запрос на сертификат» (рисунок 194).

[1] В инструкции приведен пример формирования и настройки тестового сертификата.

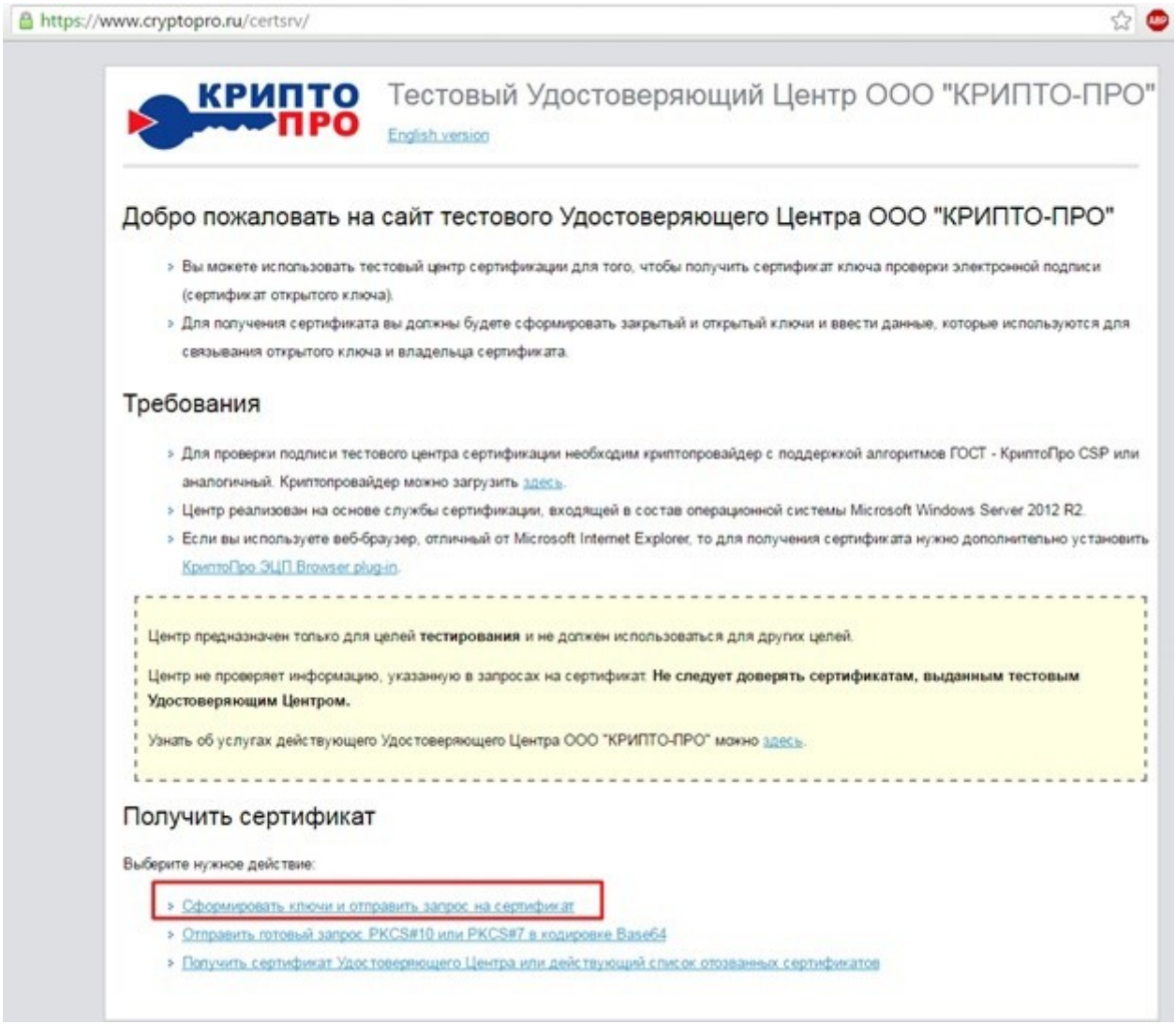


Рисунок 194 — Выпуск тестового сертификата ЭП

Форму запроса сертификата следует заполнить тестовыми данными (можно выбрать опцию «Пометить ключ как экспортируемый») и нажать кнопку «Выдать» (рисунок 195).

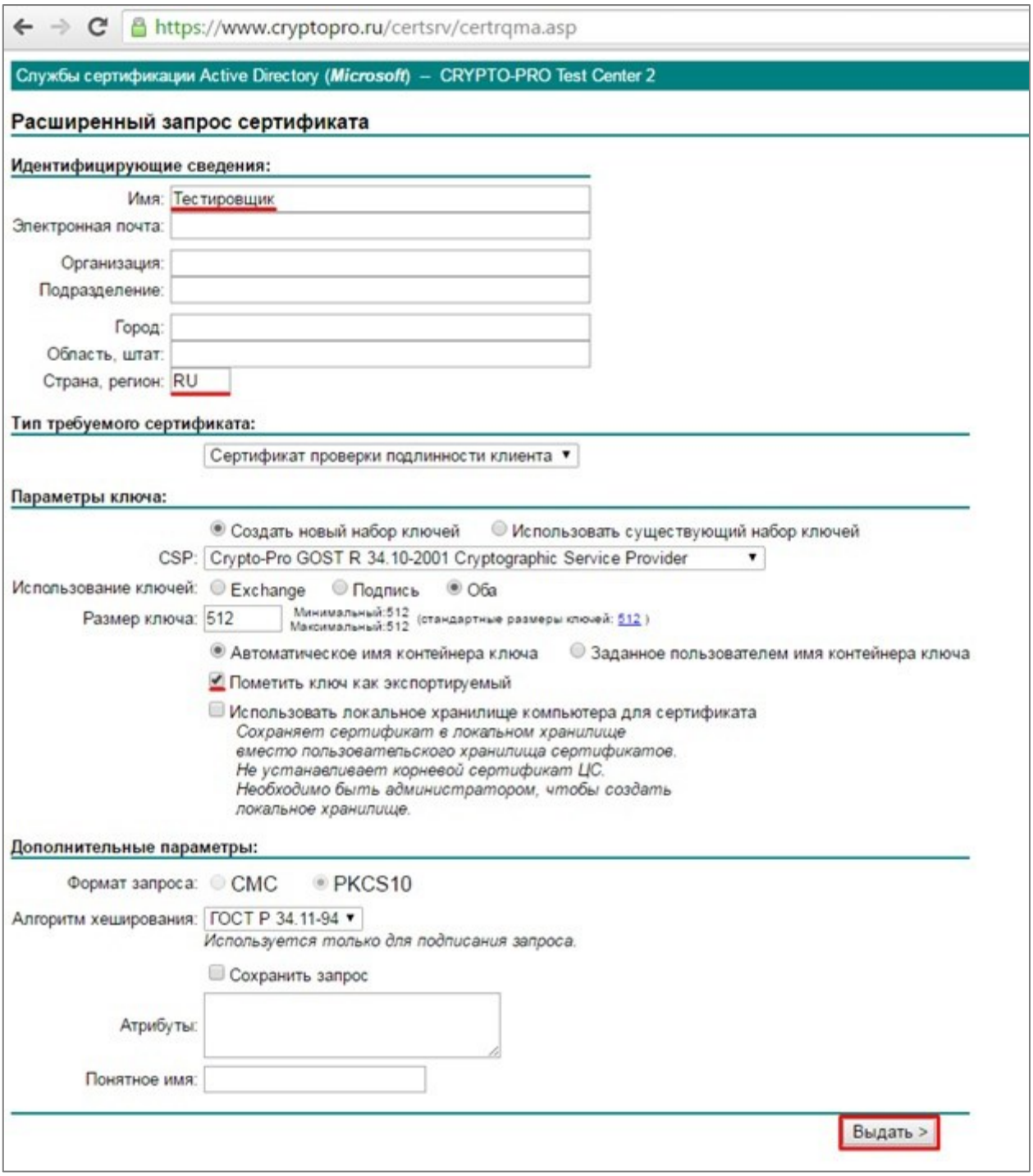


Рисунок 195 — Формирование сертификата ЭП

Далее следует в открывшемся окне, в списке устройств выбрать «Реестр» как носитель для хранения контейнера закрытого ключа и нажать кнопку «ОК» (рисунок 196).

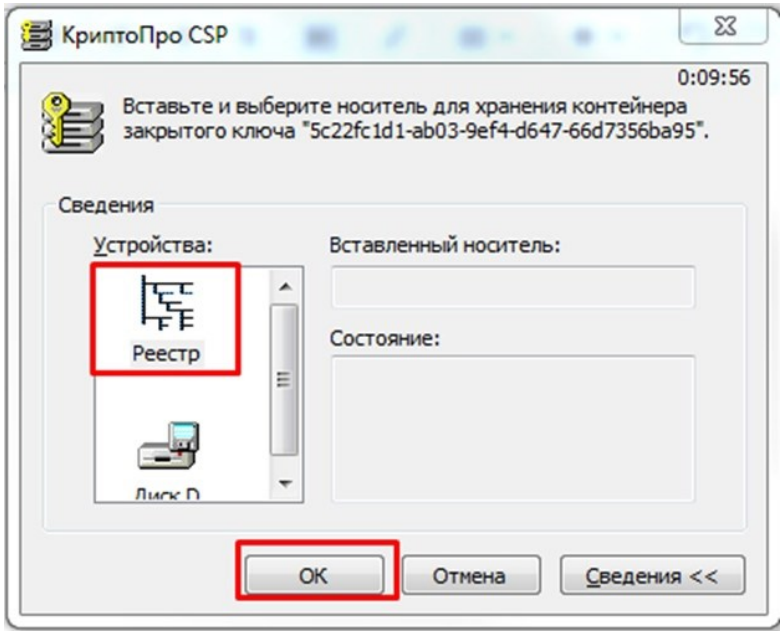


Рисунок 196 — Установка сертификата ЭП

Для генерации случайной последовательности необходимо перемещать курсор мыши по экрану и нажимать на различные кнопки до тех пор, пока процесс генерации не завершится (рисунок 197).

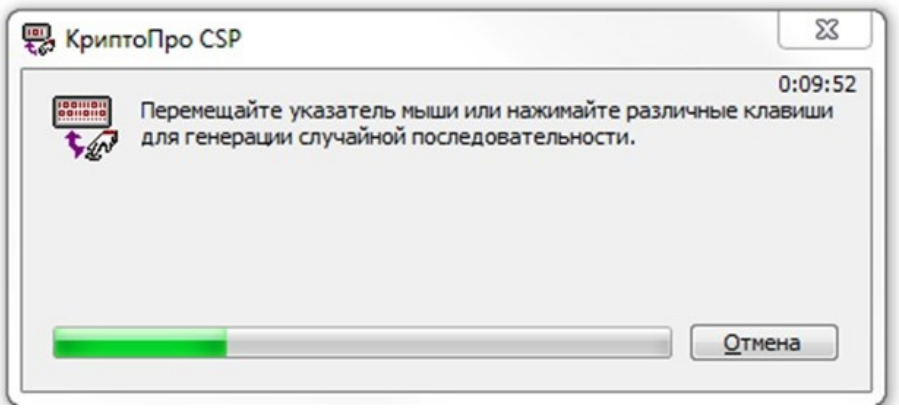


Рисунок 197 — Генерация случайной последовательности

Для закрытого ключа сертификата ЭП необходимо определить пароль (рисунок 198).

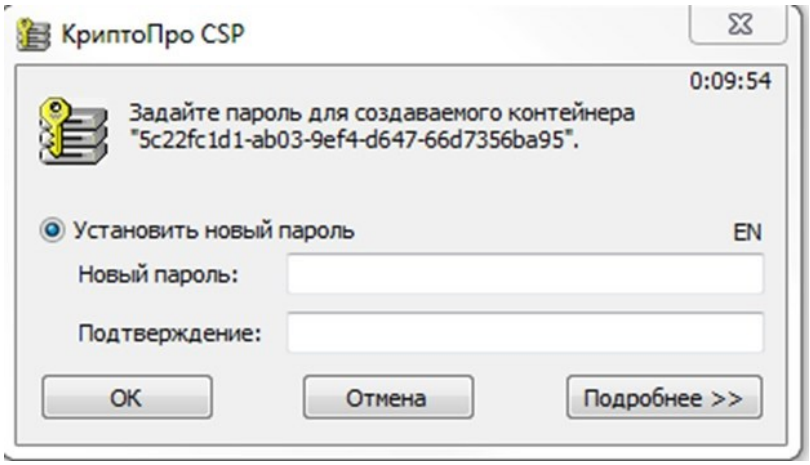


Рисунок 198 — Ввод пароля

После ввода пароля пользователя сервис открывает на страницу выдачи сертификата, на которой необходимо нажать на ссылку «Установить этот сертификат» (рисунок 199).

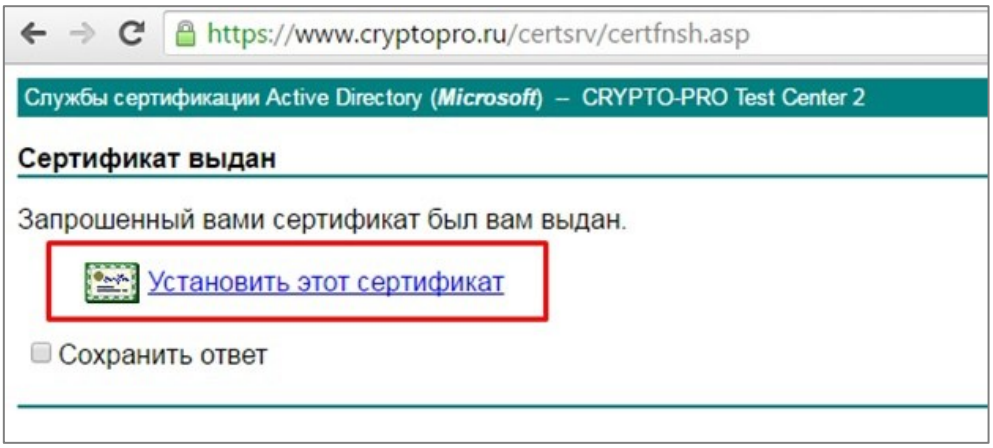


Рисунок 199 — Установка сертификата ЭП

В открывшемся окне следует ввести определенный ранее пароль для закрытого ключа сертификата ЭП (рисунок 200).

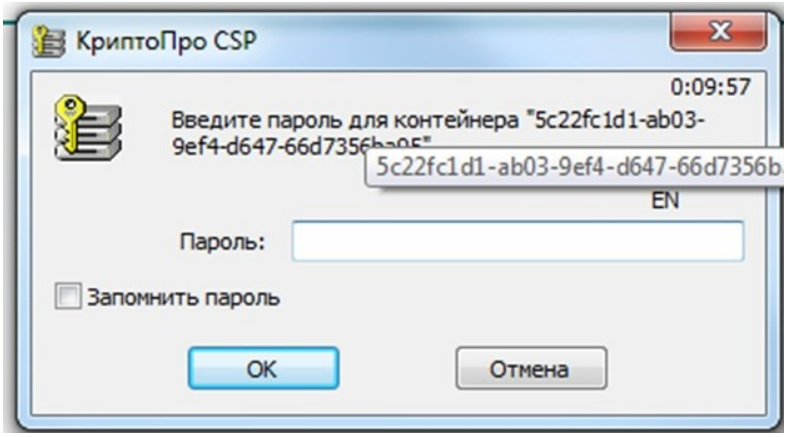


Рисунок 200 — Ввод пароля

В случае успешного завершения процесса будет отображено соответствующее сообщение (рисунок 201).

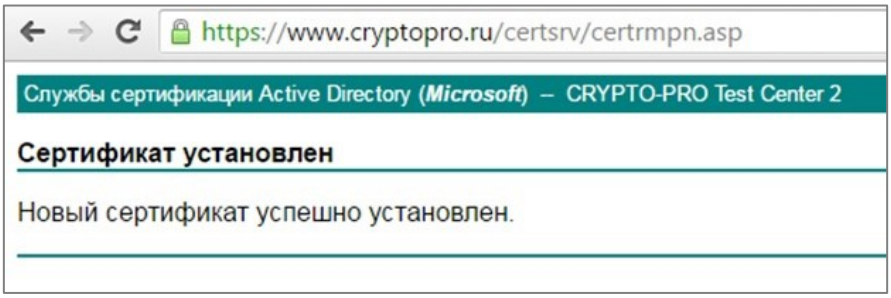


Рисунок 201 — Сообщение об успешной установке сертификата ЭП

Для проверки корректности установленного сертификата необходимо перейти на страницу https://www.cryptopro.ru/sites/default/files/products/cades/demopage/cades_bes_sample.html , где убедиться, что плагин корректно установлен в браузере. Об этом можно узнать по сообщению, выведенному на экран (рисунок 202).

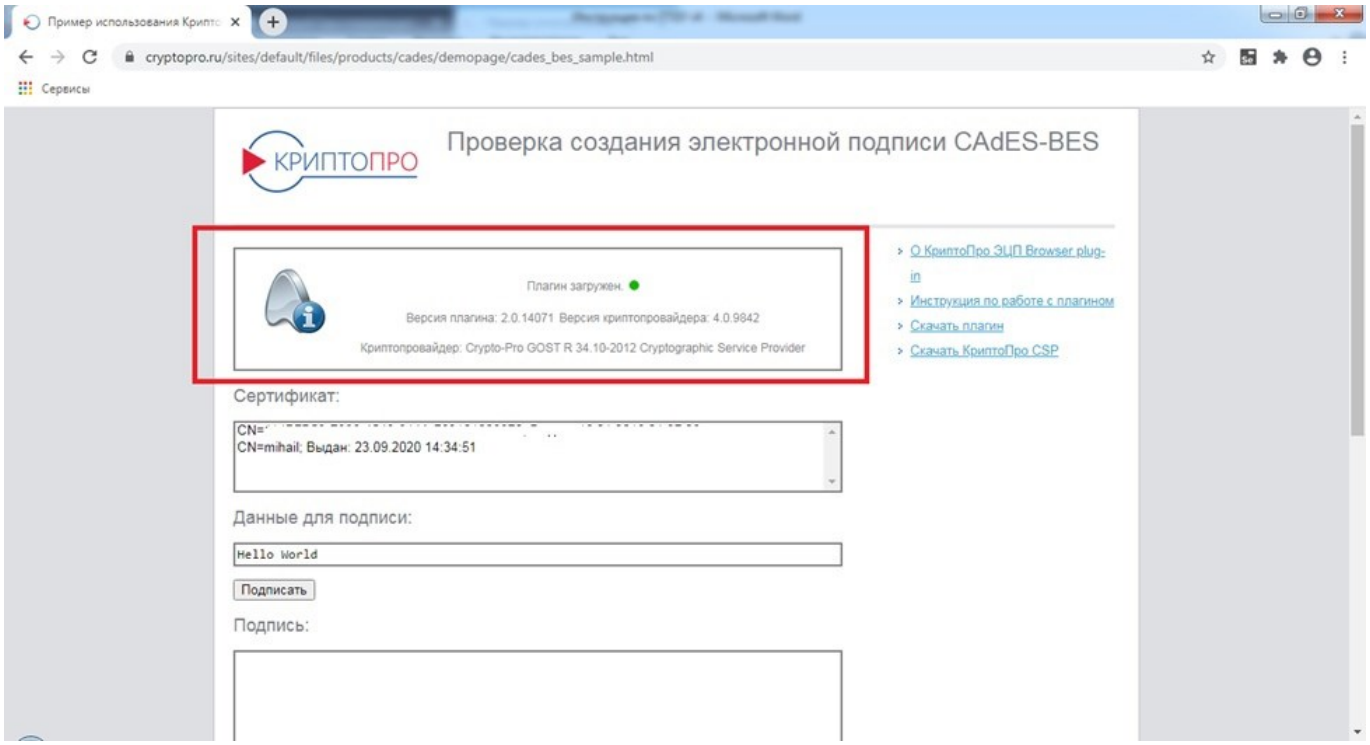


Рисунок 202 — Проверка плагина

Далее необходимо выбрать в поле «Сертификат» ранее установленный сертификат, после чего откроется форма с подробной информацией о сертификате (рисунок 203).

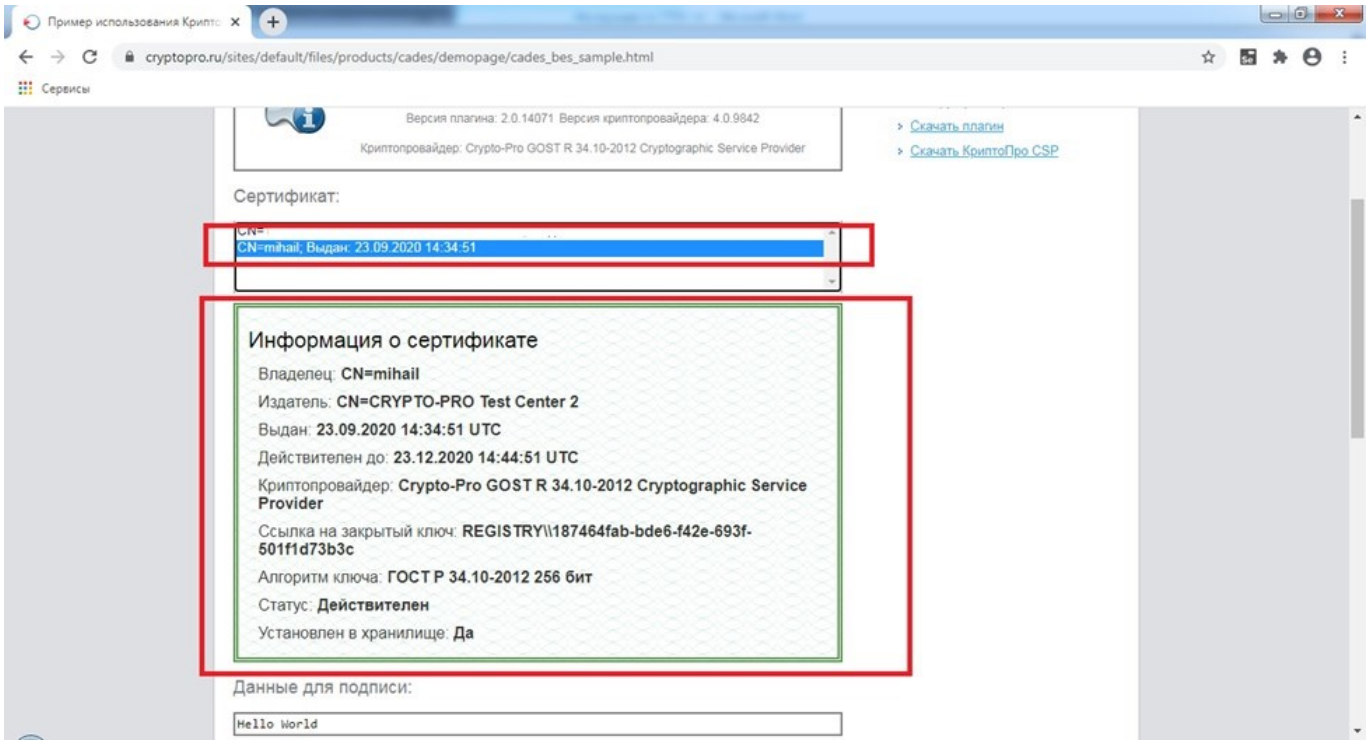


Рисунок 203 — Выбор сертификата

После выбора сертификата необходимо нажать кнопку «Подписать» и в появившейся форме ввести пароль. В случае успешной установки сертификата будет сформирована ЭП (рисунок 204).

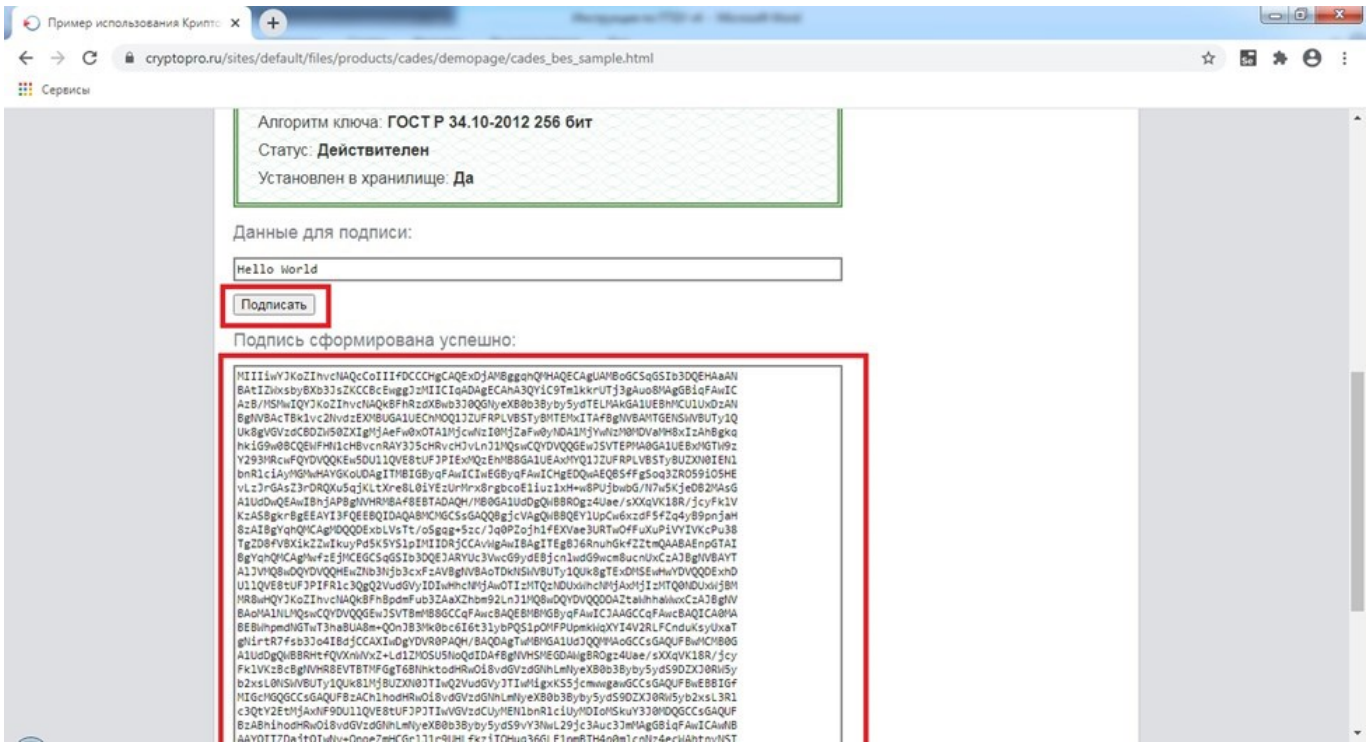


Рисунок 204 — Сформированная ЭП

На этом процедура установки и настройки криптопровайдера КриптоПро, CAdES-плагина для веб-браузера и сертификата ЭП будет завершена.

3.22. Эмуляции процесса обработки XML-заявления в Конструкторе процессов

3.22.1. Эмуляция процесса импорта заявления

Эмуляция процесса обработки XML-заявления доступна из пункта меню «Конструктор услуг» ПГС (рисунок 205).

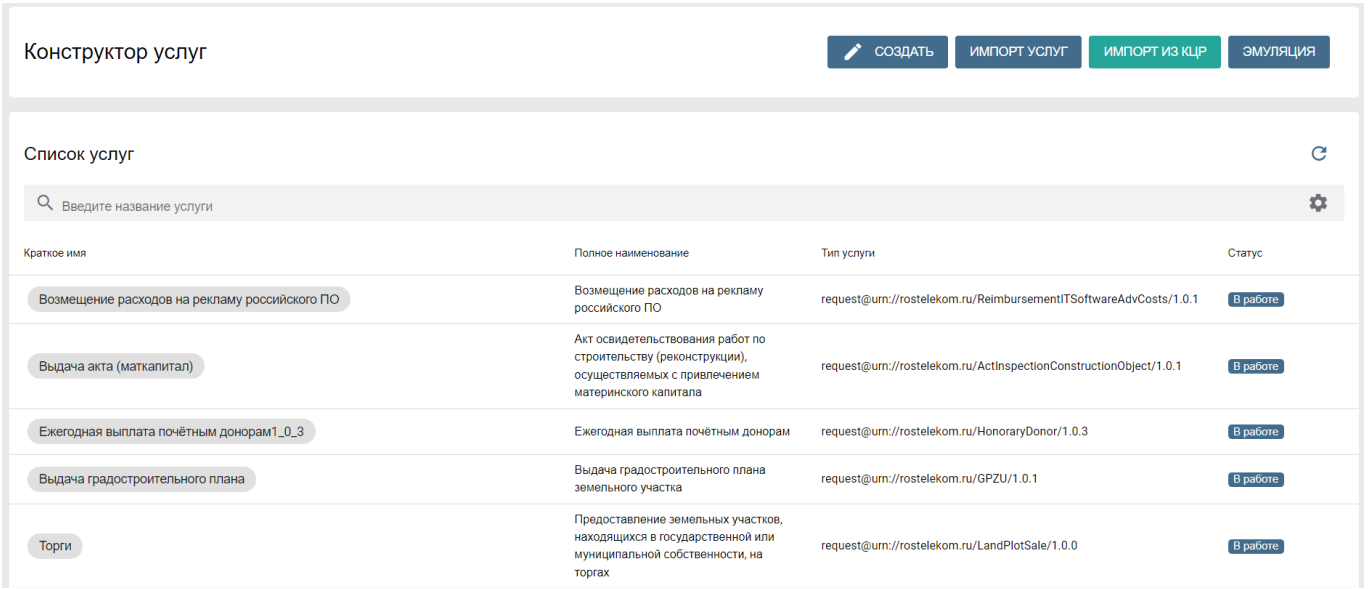


Рисунок 205 — Раздел «Конструктор услуг»

В разделе «Конструктор услуг» пользователю доступен список конфигураций всех услуг. Необходимо нажать кнопку «Эмуляция». Откроется экран эмуляции услуги (рисунок 206).

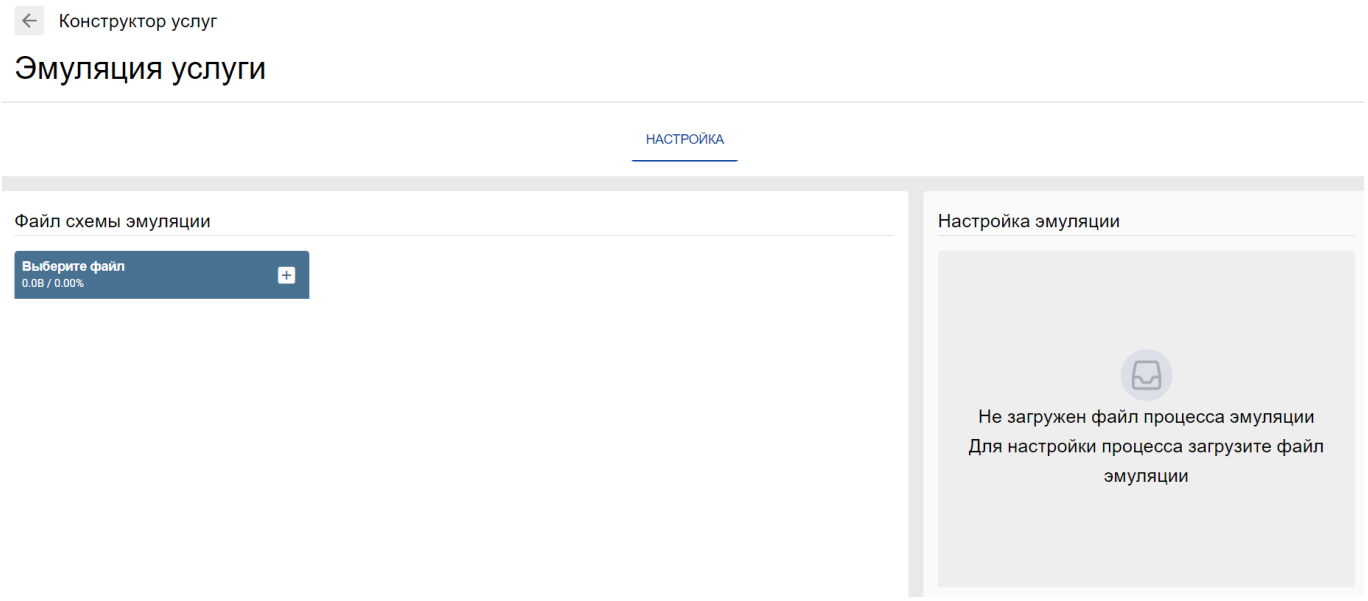


Рисунок 206 — Раздел «Конструктор услуг» – «Эмуляция услуги»

Далее необходимо выбрать и загрузить с локального компьютера файл XML-заявления. Содержимое заявления полностью отображается в окне (рисунок 207).

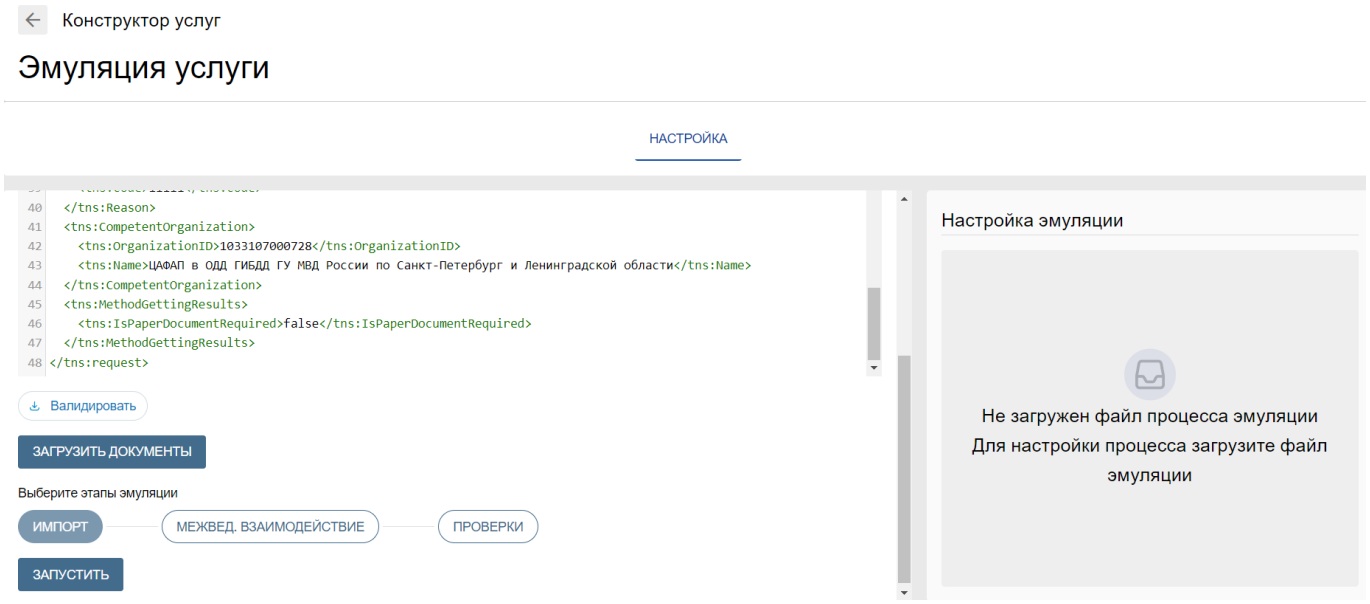


Рисунок 207 — Эмуляция услуги. Загрузка XML-заявления

Необходимо нажать кнопку «Валидировать», выбрать и загрузить с локального компьютера XSD-схему услуги. При необходимости загрузить прилагаемые к заявлению файлы. Выбрать этапы эмуляции, которые необходимо пройти (рисунок 208).

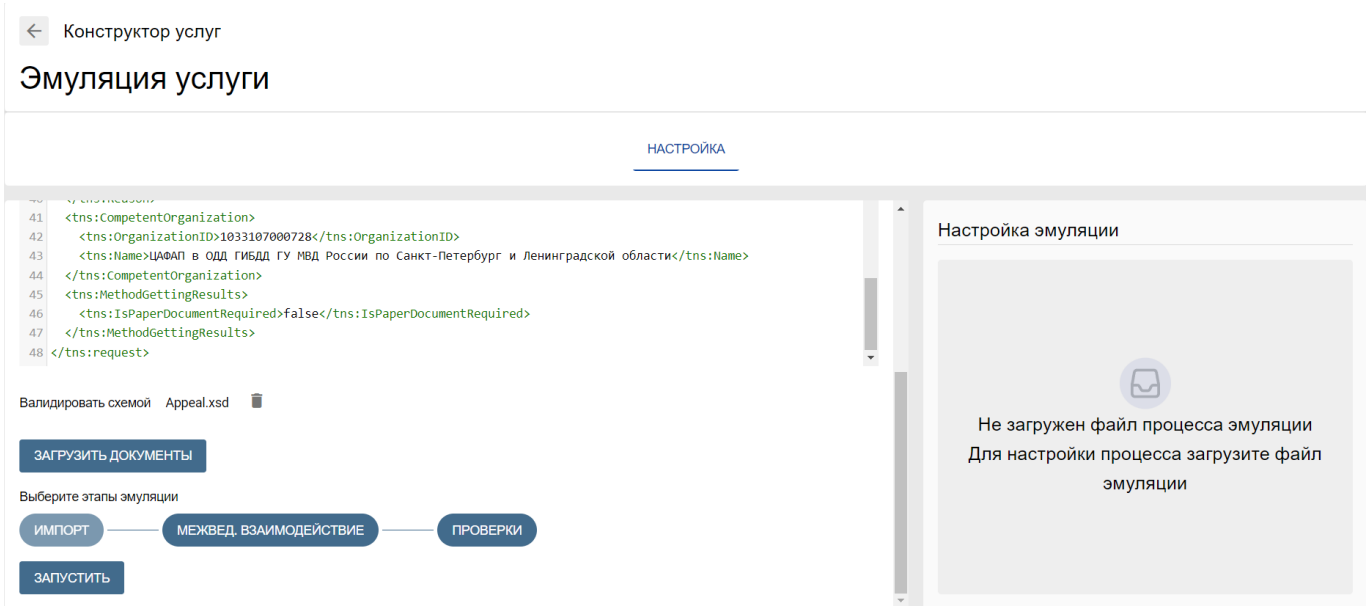


Рисунок 208 — Эмуляция услуги. Загрузка XSD-схемы и выбор этапов эмуляции

Запуск эмуляции обработки XML-заявления осуществляется нажатием кнопки «Запустить» (рисунок 209).

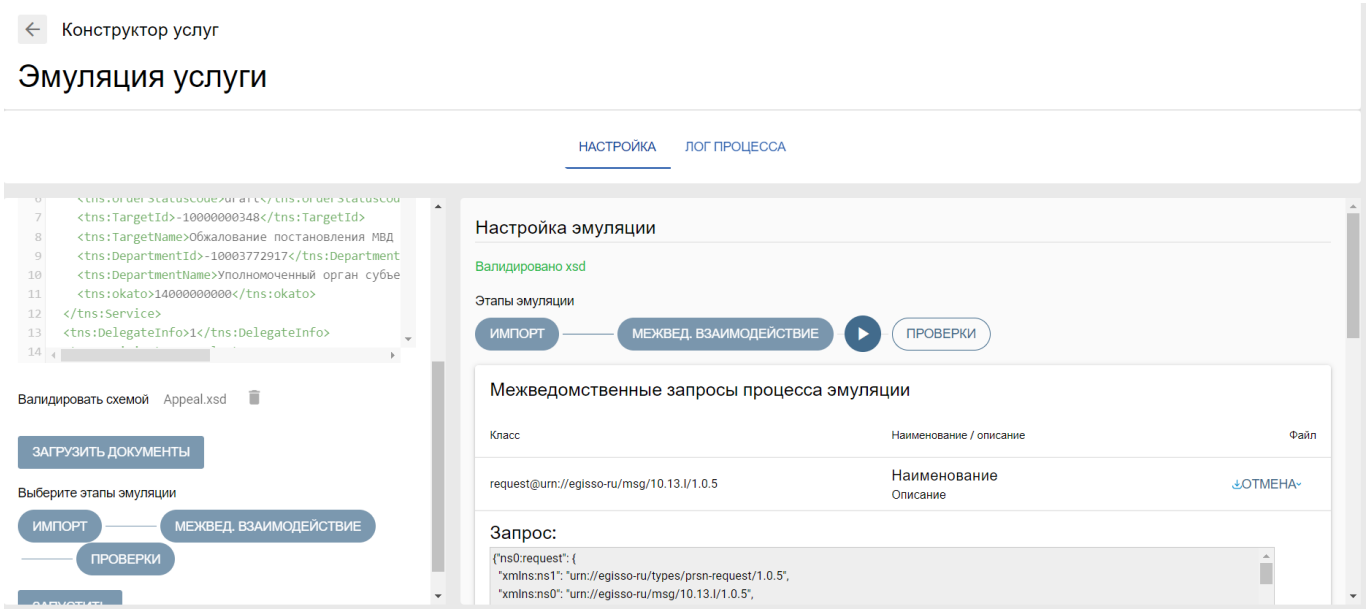


Рисунок 209 — Запуск эмуляции

Процесс работы эмулятора можно просмотреть на вкладке «Лог процесса» отдельно по этапам эмуляции (рисунок 210).

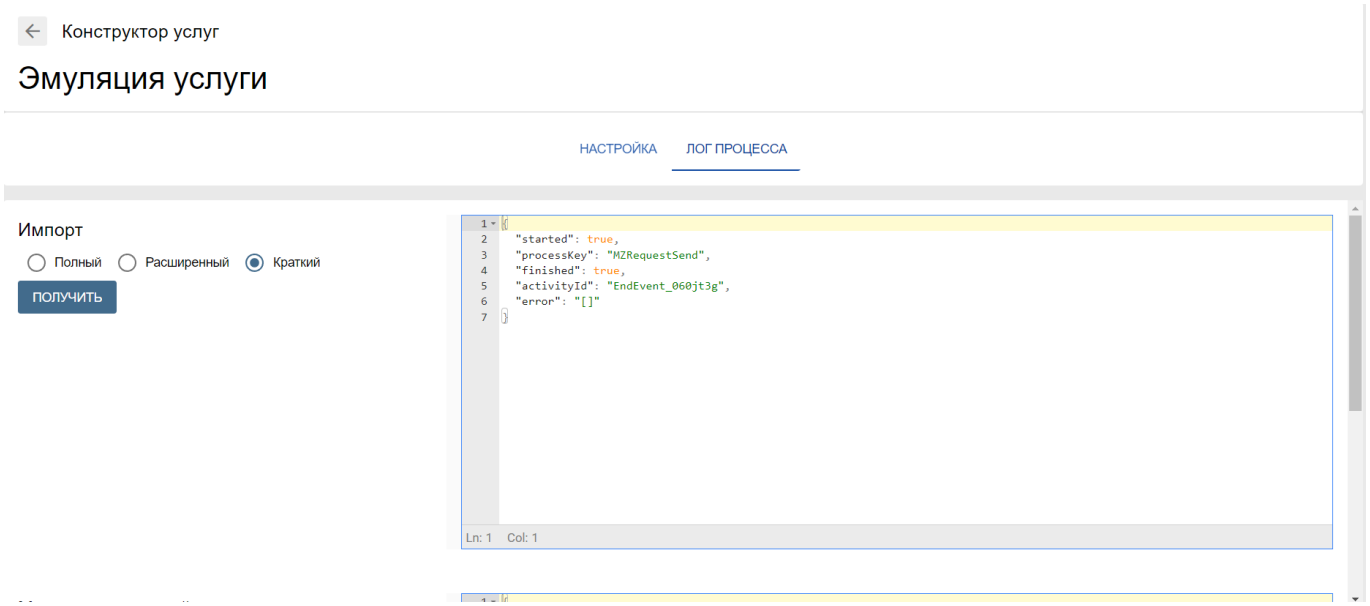


Рисунок 210 — Просмотр лога эмуляции

3.22.2. Эмуляция процесса выполнения межведомственных запросов

Эмуляция подпроцесса выполнения межведомственных запросов доступна после эмуляции подпроцесса импорта (рисунок 211).

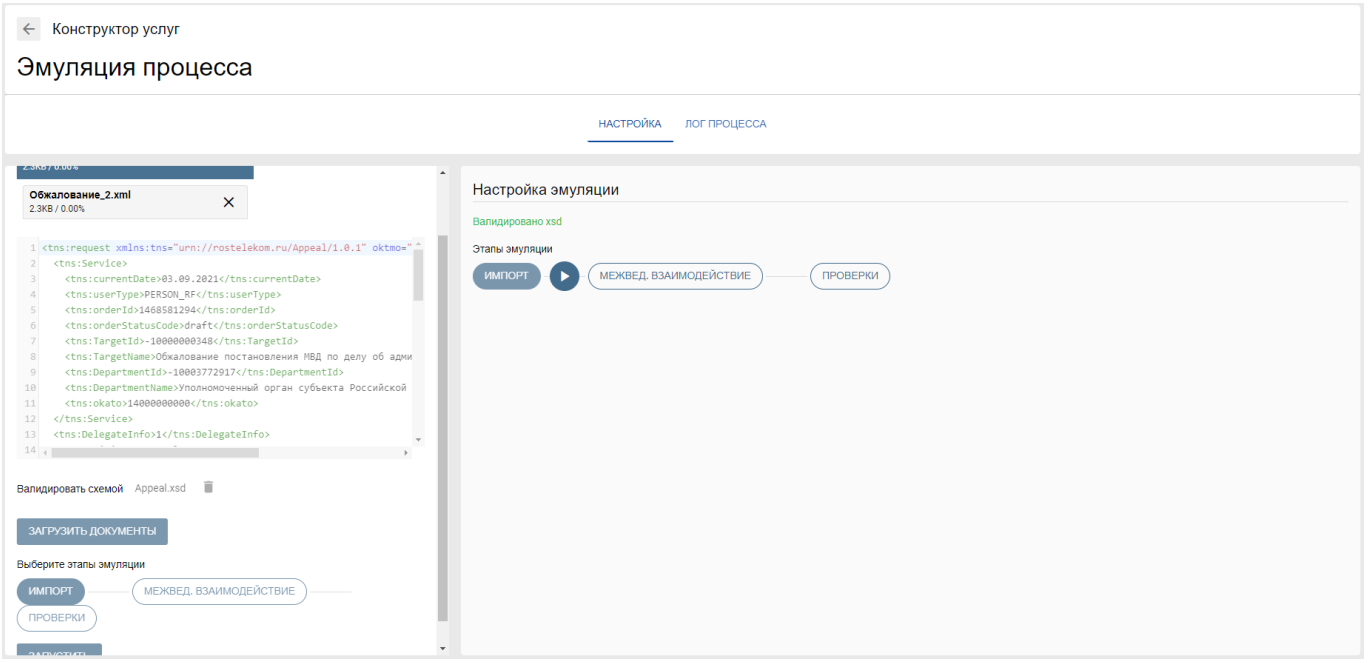


Рисунок 211 — Запуск эмуляции подпроцесса межведомственных запросов

Запуск подпроцесса осуществляется нажатием кнопки . Эмулятор из коллекции описания услуги получает список используемых межведомственных запросов (рисунок 212).

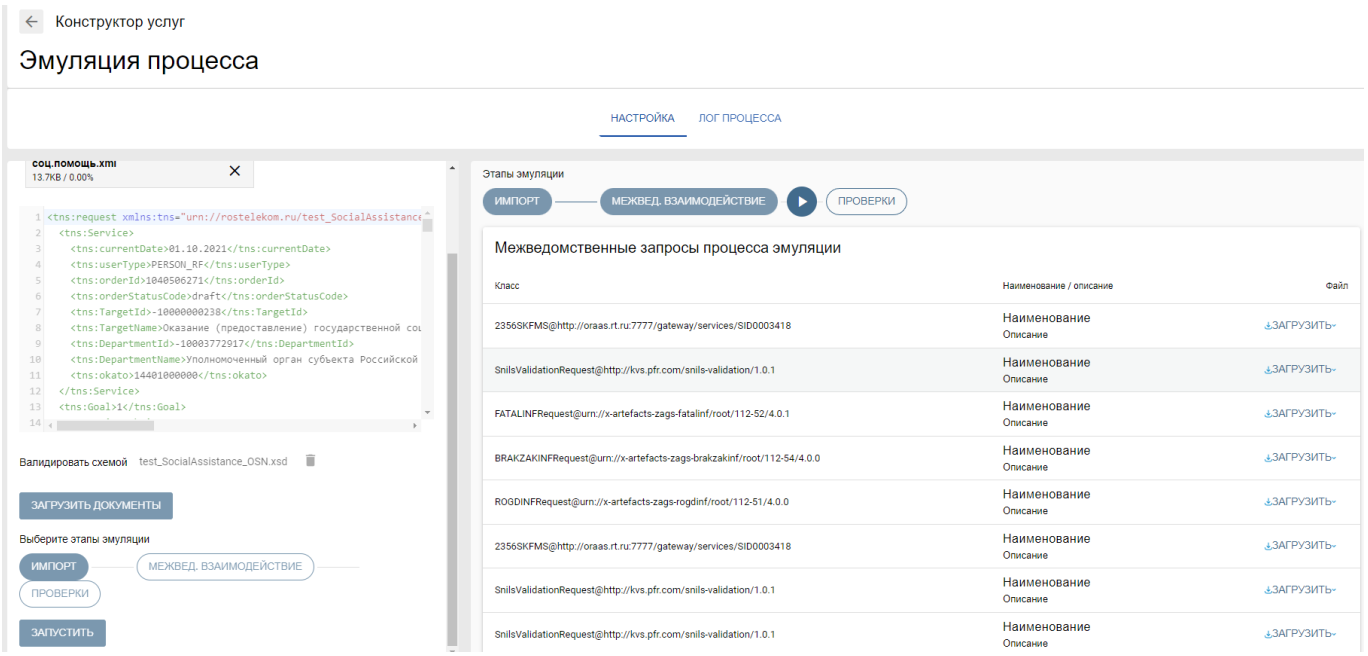


Рисунок 212 — Получение списка межведомственных запросов по услуге

Нажав кнопку «Загрузить» в строке выбранного межведомственного запроса, можно настроить ответ на запросы (рисунок 213).

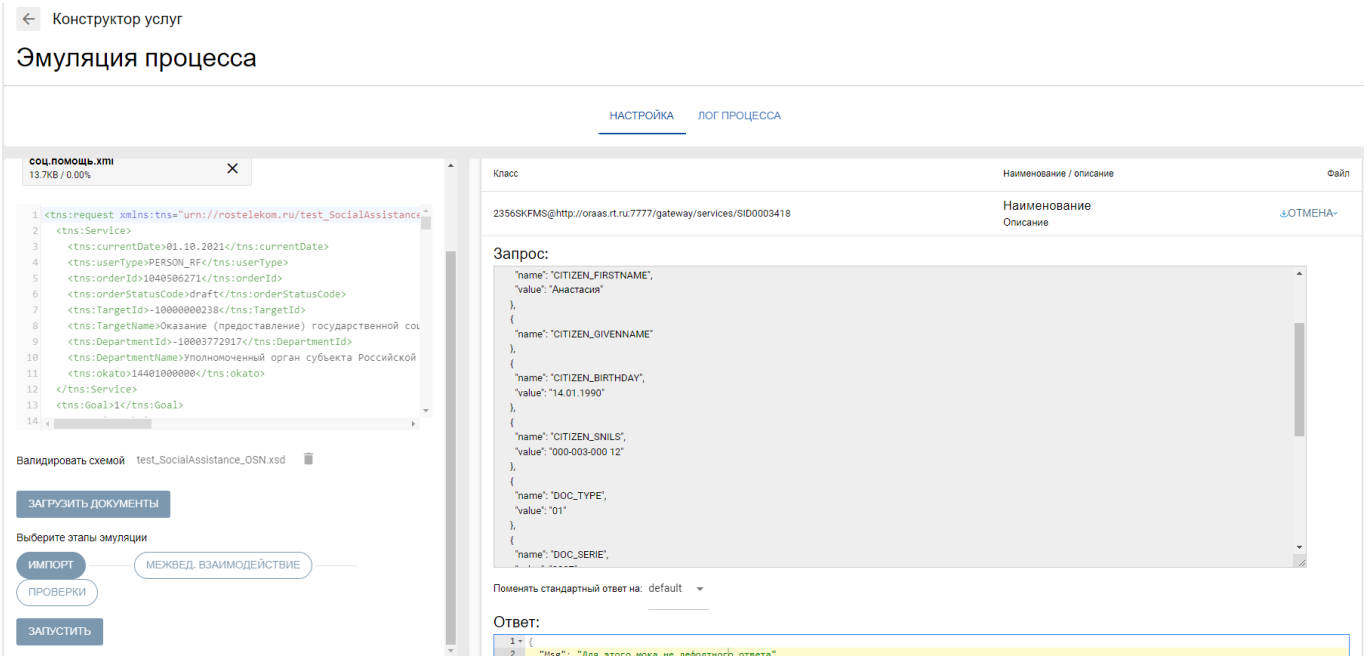


Рисунок 213 — Загрузка межведомственного запроса

В отдельном окне представлен запрос в JSON-формате. Ниже расположено окно настройки ответа на запрос (рисунок 214).

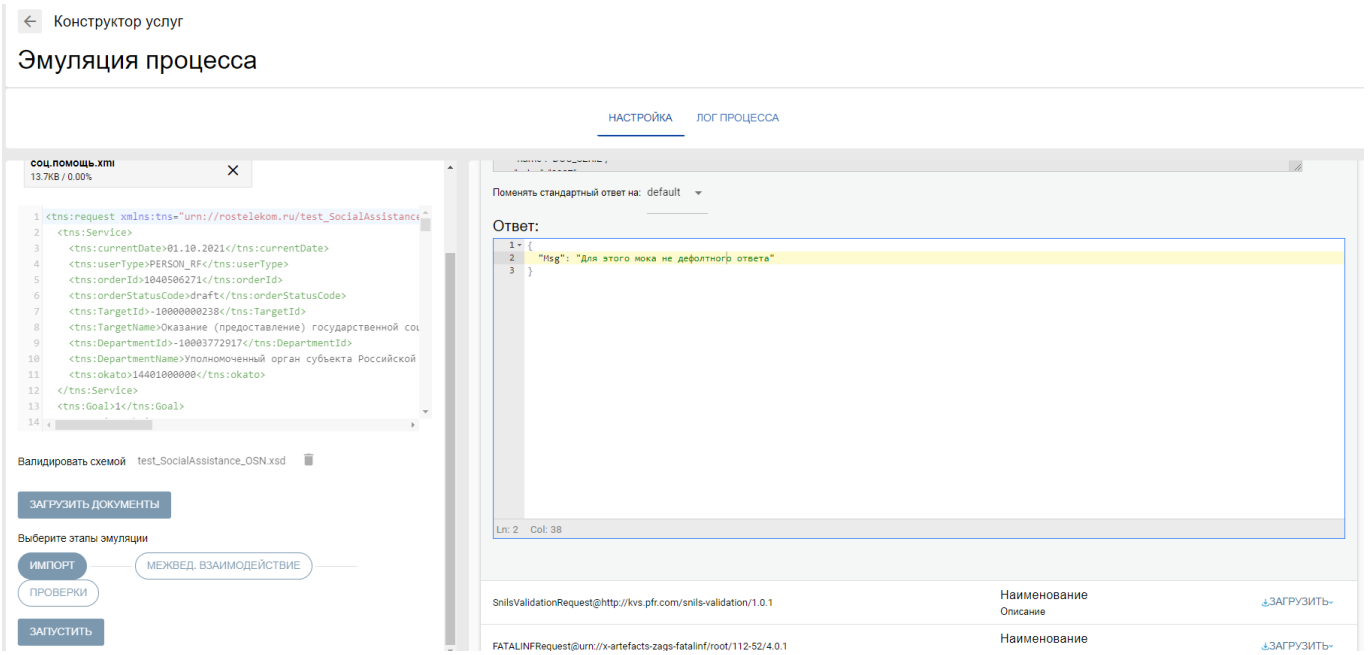


Рисунок 214 — Настройка ответа на межведомственный запрос

Настройка осуществляется путем выбора типа ответа custom. В этом режиме возможны различные варианты:

– создание ответа «с нуля»;

– загрузка готового ответа в JSON-формате, нажав на кнопку «Выберите файл»;

– дополнительное редактирование загруженного ответа.

Необходимые изменения возможно сохранить, нажав на кнопку «Сохранить» (рисунок 215).

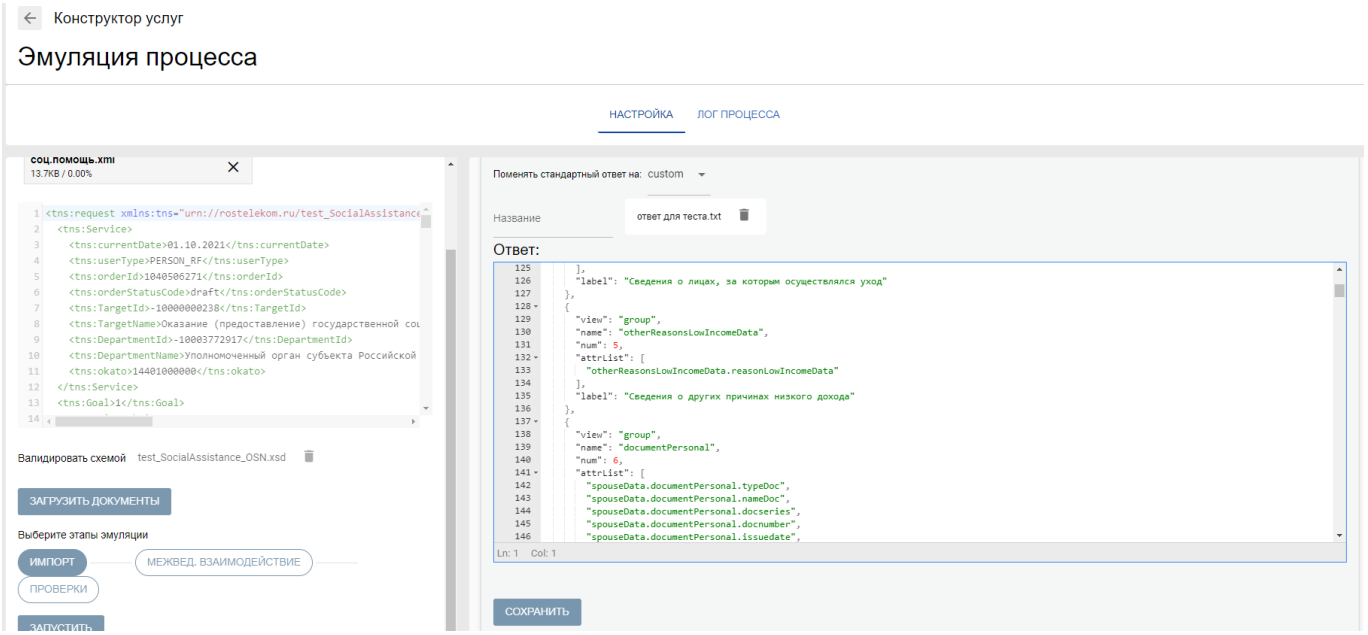


Рисунок 215 — Загрузка готового ответа на межведомственный запрос

После настройки ответов межведомственных запросов можно продолжить эмуляцию процесса предоставления услуги. Возможно возвращение к настройке межведомственных запросов с более поздних этапов эмуляции процесса предоставления услуги.

Процесс работы эмулятора подпроцесса выполнения межведомственных запросов можно просмотреть на вкладке «Лог процесса», в блоке «Межвед. взаимодействие» (рисунок 216).

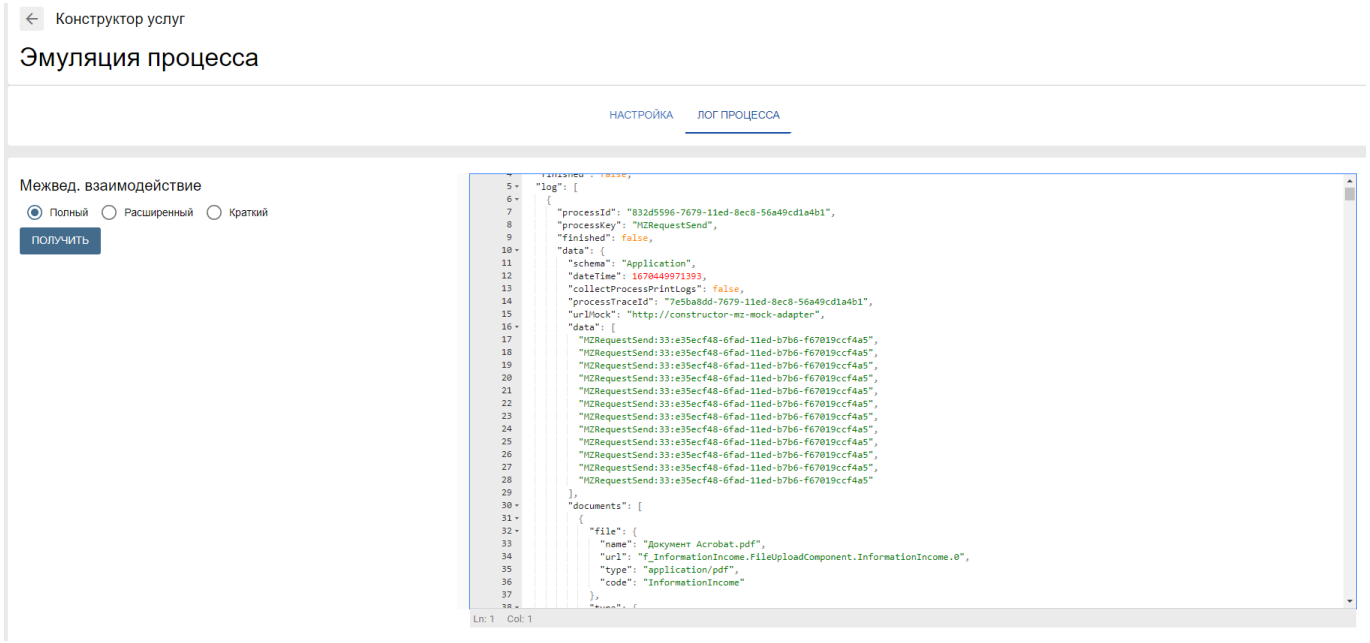


Рисунок 216 — Просмотр лога эмуляции межведомственного взаимодействия

Возможно получить логи работы эмулятора в трех вариантах: полном, расширенном или кратком. Для этого выбрать необходимый вариант и нажать кнопку «Получить».

3.22.3. Эмуляция процесса выполнения проверок

Эмуляция подпроцесса выполнения проверки соответствия критериям принятия решения доступна после эмуляции подпроцесса межведомственного взаимодействия (рисунок 217).

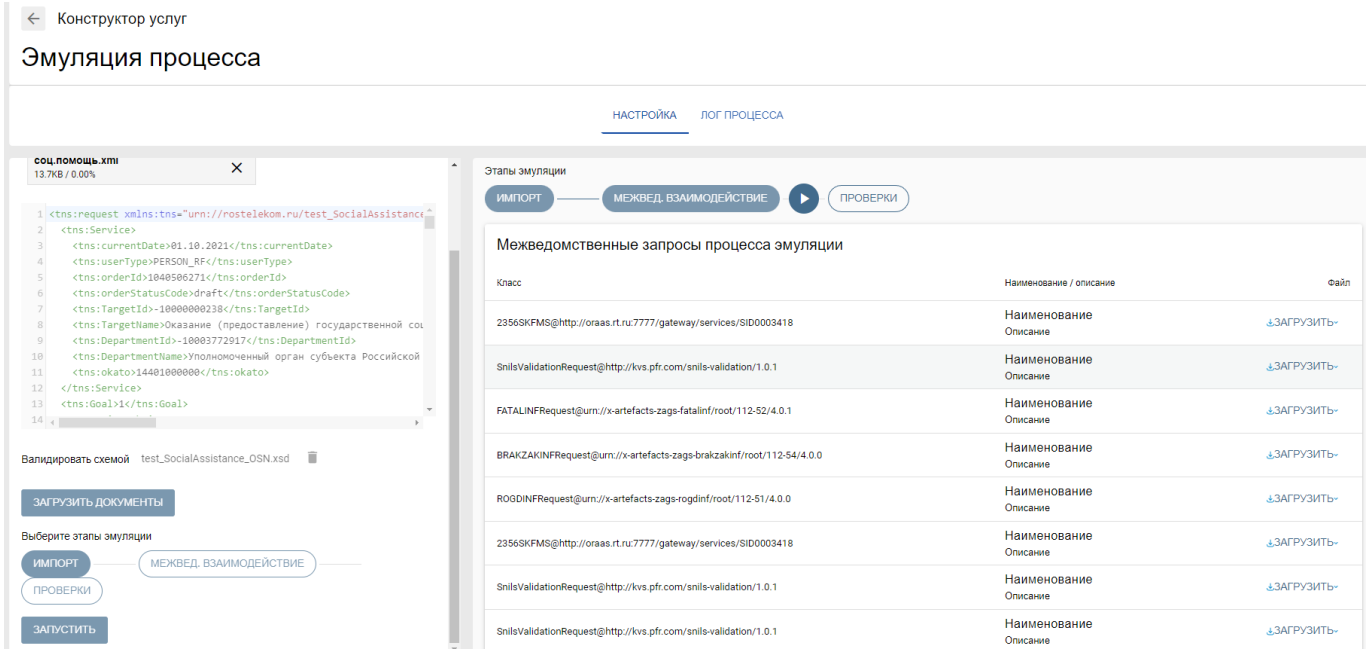


Рисунок 217 — Запуск эмуляции подпроцесса соответствия критериев принятия решения

Запуск подпроцесса осуществляется нажатием кнопки . Эмулятор из описания услуги получает список используемых межведомственных запросов (рисунок 218).

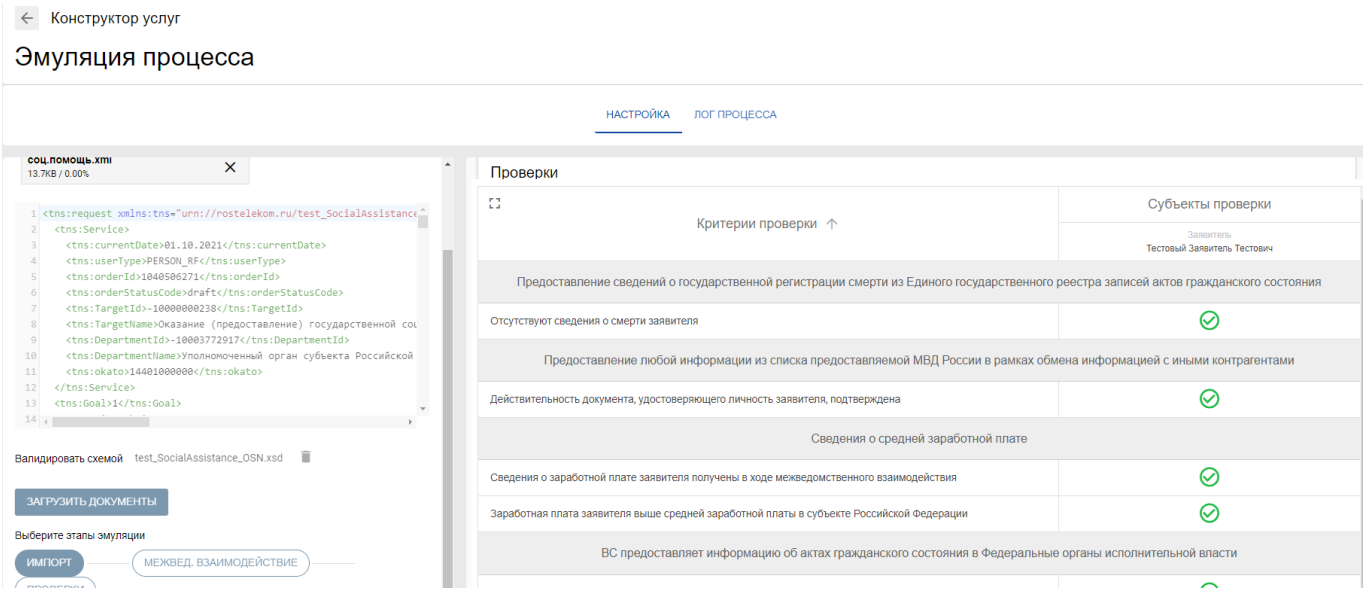


Рисунок 218 — Получение списка проверок

Переопределение входных параметров производится путем возврата на предыдущие шаги эмуляции процесса предоставления услуги («Импорт» и/или «Межвед. взаимодействие»): корректируются входные параметры заявления или ответ на межведомственный запрос. Далее снова запускается подпроцесс выполнения проверки соответствия критериям для получения матрицы принятия решения.

Процесс работы эмулятора подпроцесса можно просмотреть на вкладке «Лог процесса» в блоке «Проверки» (рисунок 219).

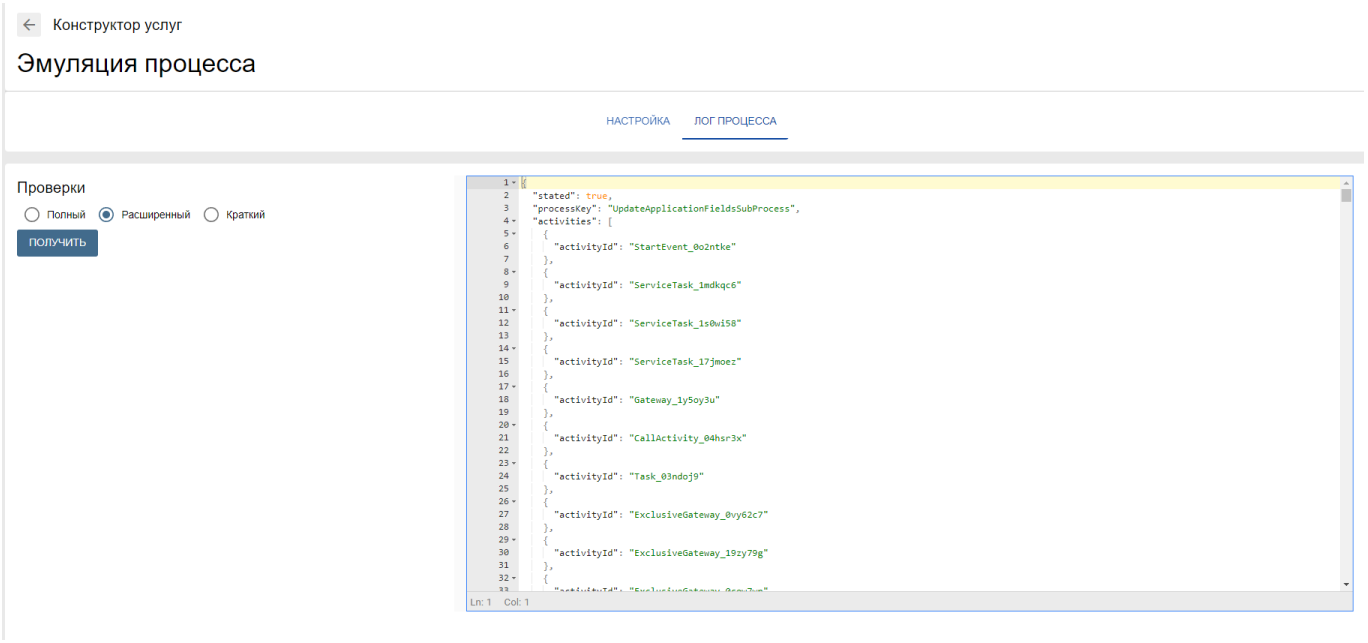


Рисунок 219 — Просмотр лога эмуляции проверок соответствия критериям

Возможно получить логи работы эмулятора в трех вариантах: полном, расширенном или кратком. Для этого требуется выбрать необходимый вариант и нажать кнопку «Получить».

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Изменение		Измененные листы			И. О. Фамилия и подпись внесшего изменение
Номер	Дата	Общее кол.	Номер листа	Краткое описание	
2	23.03.2023	все	все	1. Раздел 1.3 «Уровень подготовки специалистов». Удалена строка «- использования функциональности аутентификации ЕСИА;». 2. Раздел 2 «Условия применения Конструктора процессов предоставления услуг». Удалено упоминание ПО Postman. Изменен абзац «Кроме того, для настройки процессов предоставления услуг сотруднику необходимо: - иметь подтвержденную учетную запись в ЕСИА с предоставленными правами Администратора субъекта Российской Федерации, Администратора ведомства в ПГС, Администратора услуг ПГС;». Изменен абзац «Для создания новой услуги необходимо на стартовой странице раздела «Конструктор услуг» нажать кнопку «Создать» (см. подраздел 3.1), в открывшемся модальном окне задать краткое имя услуги (рисунок 7) и нажать кнопку «Создать»». Изменен текст «Кроме того, для настройки процессов предоставления услуг сотруднику необходимо иметь: - учетную запись с предоставленными правами Администратора услуг ПГС;». 3. Раздел 3.1 ««Авторизация и переход к конструктору процессов» - удалена информации о возможности авторизации через ЕСИА 4. Раздел 3.1 «Авторизация и переход к конструктору процессов. Список услуг». Изменен текст «а) на открывшейся странице (рисунок 1) нажать кнопку «Вход через ЕСИА», либо ввести данные локальной учётной записи (при наличии) в поле «Имя пользователя» \ «Пароль» и нажать кнопку «Вход»». 5. Раздел 3.2. «Создание услуги» Добавлены рисунок 9 и текст «Так же можно задать список пользователей, которые смогут редактировать и настраивать услугу. Выбор производится путем выбора чек-бокса напротив соответствующего пользователя. Возможен выбор всех пользователей выбором пункта «Все» (Рисунок 9)». 6. Раздел 3.3 «Внесение основной информации об услуге». Добавлен текст «При наличии xml и xsd схемы этапы, описанные в разделах 3.3–3.4, можно пропустить. Большинство данных будут сформированы при использовании функции «Импорт с ЕПГУ» (рисунок 21)». Изменены рисунки 12-15. 7. Раздел 3.12 Добавлена формулировка о необходимости перегенерации DNS схем 8. Раздел 3.14.1 Добавлена информации о необходимости доработки экранной формы после импорта	

Изменение		Измененные листы			И. О. Фамилия и подпись внесшего изменение
Номер	Дата	Общее кол.	Номер листа	Краткое описание	
				9. Раздел 3.5 «Получение процесса импорта из XSD-схемы услуги». Изменен абзац «Для того чтобы автоматически получить основную информацию по услуге (см. подраздел 3.3) и процесс импорта (см. подраздел 3.4) из XSD-схемы услуги, необходимо в карточке услуги нажать кнопку («Импортировать из ЕПГУ») справа от заголовка карточки услуги (рисунок 21). После этого откроется страница, представленная на рисунке 22». Удалены и добавлены рисунки. Добавлен текст «Раздел «Настройки» содержит автоматически сформированные сущности, полученные из ЕПГУ: основную информацию по услуге, цели услуги, а также список входных документов, прилагаемых заявителем, которые можно посмотреть, раскрыв списки (Рисунок 26)». Добавлено «Проставив соответсвующий чек-бокс, можно перейти к выполнению базового импорта (подраздел 3.6.2)». 10. Раздел 3.6.1 «Настраиваемый процесс импорта». Удален рисунок 47 «Переменная input и ее значение на вкладке «Ввод/Вывод» панели параметров элемента процесса импорта «Собираем бизнес-данные»». 11. Раздел 3.8 «Входные и выходные документы. Добавление видов входных документов для предоставления услуги». Изменен абзац «Виды документов в услуге задаются в несколько этапов. Первыми добавляются виды входных документов, в том числе и автоматически при импорте из ЕПГУ (см. подраздел 3.5)». 12. Добавлен раздел 3.10, объединивший разделы 3.10-3.12. Новым разделам назначены номера 3.10.1-3.10.3. 13. Раздел 3.20.1 «Создание и кодирование экранной формы очного приема». Изменен абзац «Наиболее простым способом получить модель заявления является отправка в ПГС какого-либо заявления (желательно, максимально заполненного), имитирующего заявление с ЕПГУ, либо мока такого заявления, и открыть через историю его requestJson». 14. Добавден раздел 3.22 «Эмуляции пр оцесса обработки XML-заявления в Конструкторе услуг»	
				1. Из перечня терминов и сокращений удалено сокращение ЭЦП. 2. В подразделе 1.1 слово «федеральной» записано со строчной буквы. 3. В подразделе 1.3 внесены корректирующие исправления в описание требований к квалификации специалистов. 4. В разделе 2 слово «адмнистратора» записано со строчной буквы.	

Изменение		Измененные листы			И. О. Фамилия и подпись внесшего изменение
Номер	Дата	Общее кол.	Номер листа	Краткое описание	
				5. В подразделе 3.1 в п. 2 удалены слово «локальной» и словосочетание «(при наличии)». 6. В подразделе 3.2 внесены корректирующие и уточняющие исправления в последний абзац. 7. В подразделе 3.3 XML и XSD записаны заглавными буквами. 8. В подразделе 3.5 произведена перекомпоновка текста двух абзацев. 9. В п. 3.20.1: – термин «мок» исправлен на «тестовый двойник»; – внесены корректирующие исправления в текст 4-го абзаца. 10. В заголовке подраздела 3.22 слово «услуг» исправлено на «процессов», удалена точка в конце заголовка. 11. В тексте подраздела 3.22: – XML, XSD иJSON записаны заглавными буквами; – слово «надо» исправлено на «необходимо»; – удалены точки в конце подписей к рисункам; – добавлены ссылки на рисунки. 12. По всему тексту документа термин «маппинг» заменен на «таблица соответствия», «соответствие», «соответствия» или «установление соответствия» в зависимости от контекста	