

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ УЧРЕЖДЕНИЕ  
НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ «ВОСХОД»

**Учебное пособие по теме**

**«Интеграция «Портала ГосЭДО» в СЭД»**

2024 год

# Содержание

Перечень принятых сокращений.....	4
Введение.....	6
1 Назначение и функции «Портала ГосЭДО».....	8
2 Возможности интеграции Программы в СЭД.....	10
3 Подготовка Программы к интеграции.....	12
4 Демонстрационный пример.....	16
4.1 Демонстрация интеграции редактора в интерфейс СЭД.....	19
4.2 Пример заполнения структурированных данных и передача контекста в редактор.....	20
4.3 Демонстрация выгрузки данных из формы в XML.....	24
4.4 Демонстрация загрузки данных из файла XML в редактор.....	26
4.5 Демонстрация передачи стилей в редактор.....	28
4.6 Демонстрация получения HTML-визуализации из файла XML.....	29
4.7 Демонстрация создания печатной формы PDF из файла XML.....	31
4.8 Демонстрация регистрации цифрового документа XML во внутреннем модуле координации.....	33
4.9 Демонстрация получения информации о состоянии цифрового процесса.....	35
4.10 Демонстрация вставки события регистрации в файл XML.....	37
4.11 Демонстрация получения данных ГАС.....	39
4.12 Демонстрация получения данных НСИ.....	40
5 Интеграция редактора структурированных данных цифрового документа.....	42
5.1 Метод передачи данных и команд в форму редактирования.....	44
5.2 Подключение обработчика событий и получение данных из формы.....	45
5.3 Получение данных из формы в формате XML.....	46
5.4 Загрузка данных в форму из XML.....	48
5.5 Передача контекста для формы редактирования.....	49
5.6 Применение CSS-стилей к форме.....	55

6	Интеграция функций обработки цифровых документов.....	57
6.1	Метод передачи данных и команд.....	57
6.2	HTML-визуализация документа из файла XML.....	58
6.3	Создание документа PDF из XML.....	59
6.4	Создание документа PDF из HTML-визуализации.....	61
6.5	Конвертация паспорта документа в цифровой документ «Служебное письмо».....	63
6.6	Конвертация контейнера МЭДО в контейнер цифрового документа «Служебное письмо».....	65
6.7	Вставка событий регистрации и ограничения доступа в XML.....	67
6.8	Регистрация цифрового документа во внутреннем модуле координации «Портала ГосЭДО».....	72
6.9	Получение информации о состоянии цифрового процесса.....	74
7	Возможности интеграции сервисов работы с ГАС, НСИ и СППР.....	77
7.1	Запрос данных из ГАС через REST API.....	78
7.2	Запрос справочников НСИ через REST API.....	80

## Перечень принятых сокращений

Сокращение	Описание
ГАС	– Глобальный адресный справочник
ГосЭДО	– Информационная система обеспечения внутриведомственного и межведомственного документооборота и контроля исполнения поручений, в том числе с использованием облачных сервисов гриф «Для служебного пользования»
ДСП	– Система межведомственного электронного документооборота
МЭДО	– Система межведомственного электронного документооборота
НСИ	– Нормативно-справочная информация
СППР	– Сервис автоматизации поддержки принятия решений
СЭД	– Система электронного документооборота
ФИО	– Фамилия, имя и отчество
API	– (англ. Application Programming Interface) - программный интерфейс, описание способов взаимодействия одной компьютерной программы с другими
CSS	– (англ. Cascading Style Sheets) -каскадные таблицы стилей, язык декодирования и описания внешнего вида веб-страницы, написанного с использованием языка разметки HTML
GUID	– (англ. Globally Unique Identifier) - глобально уникальный идентификатор
HTML	– (англ. HyperText Markup Language) - стандартизированный язык разметки документов для создания структуры веб-страниц
ID	– (англ. Identifier) - идентификатор
PDF	– (англ. Portable Document Format) – формат электронной книги с «жесткой, фиксированной версткой», отображающий текст и графику
POST	– метод запроса на веб-странице, предназначен для направления запроса, при котором веб-сервер принимает данные, заключённые в тело сообщения
REST	– (англ. Representational State Transfer) — архитектурный стиль взаимодействия компонентов распределённого приложения в сети

- URL – (англ. Uniform Resource Locator) – адрес ресурса в сети
- XML – (англ. Extended Markup Language) – расширяемый язык разметки

## Введение

Одним из способов обмена документами в электронном виде между государственными организациями является система межведомственного электронного документооборота (МЭДО). Основой подхода к обмену документами в МЭДО является свод единых форматов передачи документов через транспортную шину. Документы пересылаются в едином формате транспортного контейнера МЭДО 2.7.1., который подразумевает передачу основного документа в формате PDF, информация в котором не всегда может быть распознана программными средствами.

Минцифры России по заказу Аппарат Правительства Российской Федерации (РФ) провело разработку нового формата документа - цифрового документа. Основой цифрового документа является файл структурированных данных — файл в формате XML, в котором в отдельные поля внесены сведения документа. Такой файл является машиночитаемым и позволяет на программном уровне обрабатывать информацию из документа.

По распоряжению Аппарата Правительства РФ часть организаций должны перейти на обмен с использованием цифровых документов. Однако на данный момент большинство систем электронного документооборота (СЭД), используемых в государственных организациях, не поддерживают работу с цифровыми документами.

Минцифры России было создано типовое программное решение «Портал ГосЭДО», которое позволяет создавать файл структурированных данных цифрового документа, дополнять им сформированный в СЭД транспортный контейнер документа, а также поддерживает работу с новыми централизованными сервисами справочной и аналитической информации. При разработке программного решения «Портал ГосЭДО» (далее также - Программа) была предусмотрена возможность интеграции его функций в другие системы электронного документооборота, в том числе с помощью API.

Настоящее учебное пособие предназначено для ознакомления с порядком интеграции типового программного решения «Портал ГосЭДО» в СЭД.

Материал будет полезен разработчикам систем электронного документооборота, которые планируют внедрение в своих программных решениях функций работы с цифровыми документами в рамках обмена по МЭДО.

В учебном пособии рассмотрены следующие темы:

- назначение и функции «Портала ГосЭДО»;
- возможности интеграции «Портала ГосЭДО» в СЭД;
- подготовка «Портала ГосЭДО» к интеграции;
- демонстрационный пример;
- интеграция редактора структурированных данных;
- интеграция функций обработки цифровых документов;
- возможности интеграции сервисов работы с ГАС, НСИ и СППР.

# 1 Назначение и функции «Портала ГосЭДО»

Модуль «Портал ГосЭДО» является типовым программным решением, которое обеспечивает работу с сервисами ГосЭДО: глобальный адресный справочник (ГАС), нормативно-справочной информации (НСИ) и сервиса автоматизации поддержки принятия решений (СППР). Также «Портал ГосЭДО» позволяет формировать документы в новом формате цифрового документа, требующем наличия файла структурированных данных (XML-файла) в составе пересылаемого транспортного контейнера. Цифровой документ – это электронный документ, представленный в машиночитаемом виде в цифровых форматах, разработанных и утвержденных регуляторами информационного взаимодействия.

«Портал ГосЭДО» позволяет выполнять следующие функции:

- формирование файла структурированных данных цифрового документа (XML-файла);
- дополнение контейнера документа формата МЭДО 2.7.1 структурированными данными в формате цифрового документа и их отправка получателям;
- перевод входящих цифровых документов в неструктурированный формат МЭДО 2.7.1 и их передача в систему электронного документооборота (СЭД);
- формирование запросов на получение актуальных сведений ГАС и обработка ответов от сервиса ГАС;
- формирование запросов на получение актуальных сведений НСИ и обработка ответов от сервиса НСИ;
- формирование запросов на формирование отчетов СППР и выгрузка полученных отчетов для просмотра;
- обновление информации об организации в ГАС.

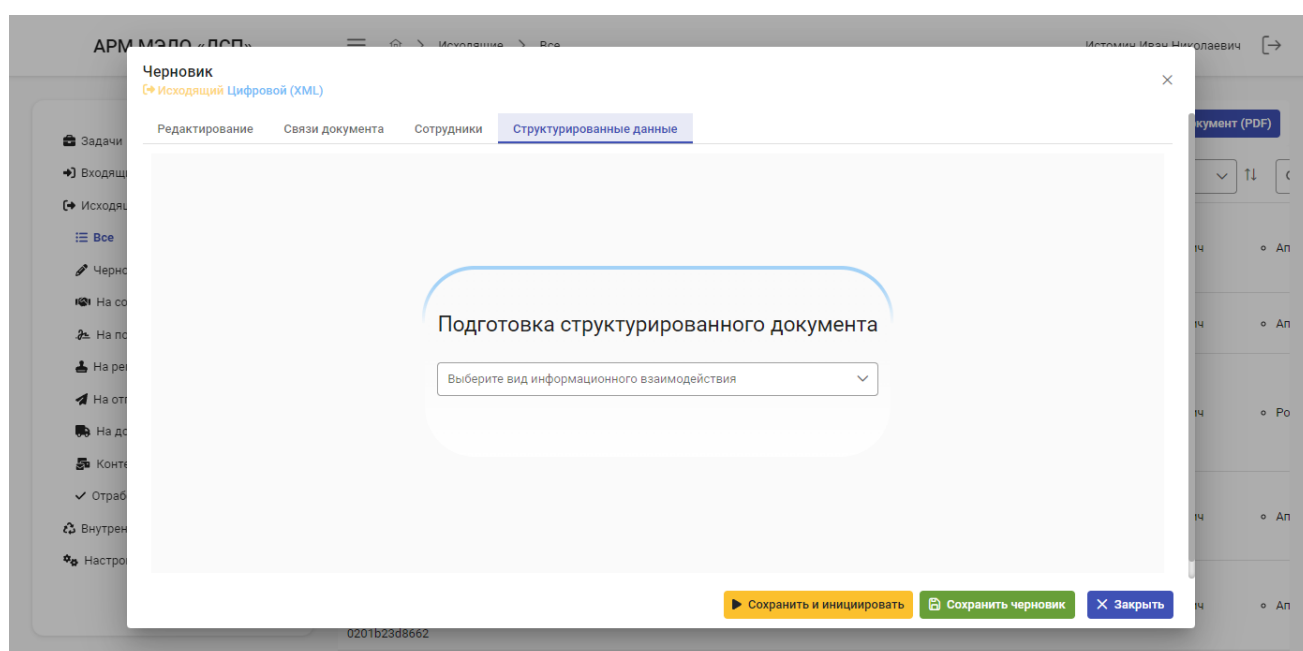


Для ознакомления с полным перечнем поддерживаемых функций и порядком их выполнения необходимо изучить дистанционный курс Центра обучения ГосЭДО «Базовый курс по работе с Порталом ГосЭДО».

## 2 Возможности интеграции Программы в СЭД

«Портал ГосЭДО» поддерживает различные возможности интеграции в СЭД.

Для интеграции «Портала ГосЭДО» с целью подготовки файла структурированных данных цифрового документа (digital.xml) рекомендуется встроить форму редактора структурированных данных в интерфейс карточки СЭД (например, открытие по кнопке или на отдельной вкладке карточки документа). Пример интеграции формы подготовки структурированных данных на отдельной вкладке карточки документа приведён на Рисунок 2.1.



**Рисунок 2.1. Пример реализованной интеграции формы подготовки структурированных данных в карточке документа СЭД**

Это позволит формировать файл структурированных данных цифрового документа (XML) сразу в карточке документа СЭД, изменять этот файл, просматривать его, подготавливать PDF-визуализацию, дополнять файл данными о регистрации документа и других событиях. Сформированный в редакторе XML-файл должен быть помещен средствами СЭД как вложение в транспортный контейнер формата МЭДО. При таком методе интеграции формирование и отправка исходящего транспортного контейнера документа должна быть

выполнена средствами СЭД. Подробнее о реализации интеграции описано в разделах 5 Интеграция редактора структурированных данных цифрового документа и 6 Интеграция функций обработки цифровых документов.

Функции обработки цифровых документов, доступные через взаимодействие с API, также могут быть использованы для обработки входящих цифровых документов.

Данные, формируемые на «Портале ГосЭДО», зависят от справочников ГАС и НСИ, поэтому важно обеспечить своевременное обновление этих справочников. Для поддержания в актуальном состоянии справочников ГАС и НСИ, а также для доступа к работе с отчетами СППР можно интегрировать соответствующие разделы «Портала ГосЭДО» в интерфейс СЭД через iframe. В этом случае сервисы открываются как веб-страница внутри СЭД. Пользователь может отправлять запросы и просматривать ответы на них, работая в интерфейсе «Портала ГосЭДО», интегрированном в СЭД. Некоторые сведения ГАС и НСИ могут быть получены через запросы к REST API. Подробнее о возможностях интеграции сервисов ГАС, НСИ и СППР описано в разделе 7 Возможности интеграции сервисов работы с ГАС, НСИ и СППР.

### 3 Подготовка Программы к интеграции

Для выполнения интеграции «Портала ГосЭДО» в СЭД необходимо развернуть экземпляр Программы на сервере и обеспечить доступ к нему при обращении из исполняемого кода СЭД.

Для получения актуального дистрибутива, системных требований и инструкции по установке «Портала ГосЭДО» необходимо обратиться в Центр поддержки ГосЭДО (+7(800)500-85-92, support@gosedo.ru).

Можно рассмотреть следующие варианты установки Программы:

1. Развернуть «Портал ГосЭДО» без подключения к транспортной шине МЭДО, тогда одной копией экземпляра Программы могут пользоваться более 1 организации. При этом следует учесть следующие условия:
  - входящие и исходящие цифровые документы не регистрируются автоматически в модуле координации экземпляра «Портала ГосЭДО», поэтому следует инициировать регистрацию цифровых документов (входящих и исходящих) в модуль координации через запрос к API (подробнее в разделе 6.8 Регистрация цифрового документа во внутреннем модуле координации «Портала ГосЭДО»);
  - при регистрации документов в модуле координации рекомендуется обеспечить изоляцию данных, используя присвоение идентификатора сессии, что позволит изолировать документы разных организаций друг от друга, как если бы документы с одним идентификатором сессии хранились в отдельной копии Программы;
2. Развернуть «Портал ГосЭДО» на локальном узле каждой организации и выполнить настройку Программы в соответствии с типовыми схемами подключения между СЭД и шлюзом МЭДО (подключить «в разрез» между СЭД и МЭДО). Из настроек СЭД предусмотреть ввод URL-адреса для обращения по API к экземпляру Программы, установленной в организации.

В этом случае регистрация во внутреннем модуле координации «Портала ГосЭДО» всех зарегистрированных входящих и исходящих цифровых документов данной организации будет выполняться Программой автоматически.

Одной из ключевых настроек Программы является заполнение информации об организации согласно ГАС в разделе «**Настройки**» → «**МЭДО**» (Рисунок 3.1), а именно поля:

- «**Наименование организации**»;
- «**GUID адрес организации**».

Портал ГосЭДО

Тема: Светлая Темная Выгрузить информацию для технической поддержки Положить контейнер

- > Служба
- > Общее
- ▼ МЭДО
  - Наименование организации
  - GUID адрес организации
  - Папка входящих сообщений ГосЭДО из шлюза

./Storage/medo\_in

**Рисунок 3.1. Поля для заполнения сведений об организации в разделе «Настройки» → «МЭДО» Программы**

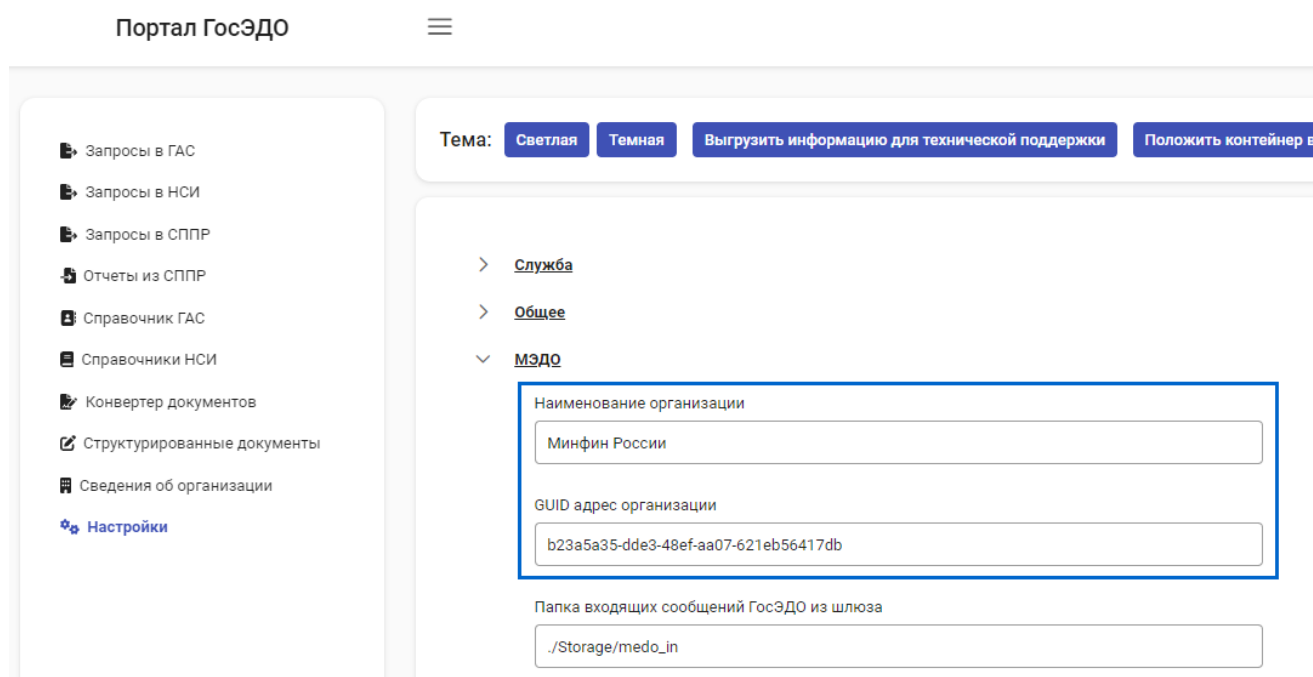
Информация из этих полей используется «Порталом ГосЭДО» при формировании сведений об авторе и отправителе документов, а также влияет на отображение списка доступных процессов при работе с формой структурированных данных.

Возможны следующие варианты настройки информации об организации:

- не заполнять поля «**Наименование организации**» и «**GUID адрес организации**», тогда при обращении к форме подготовки

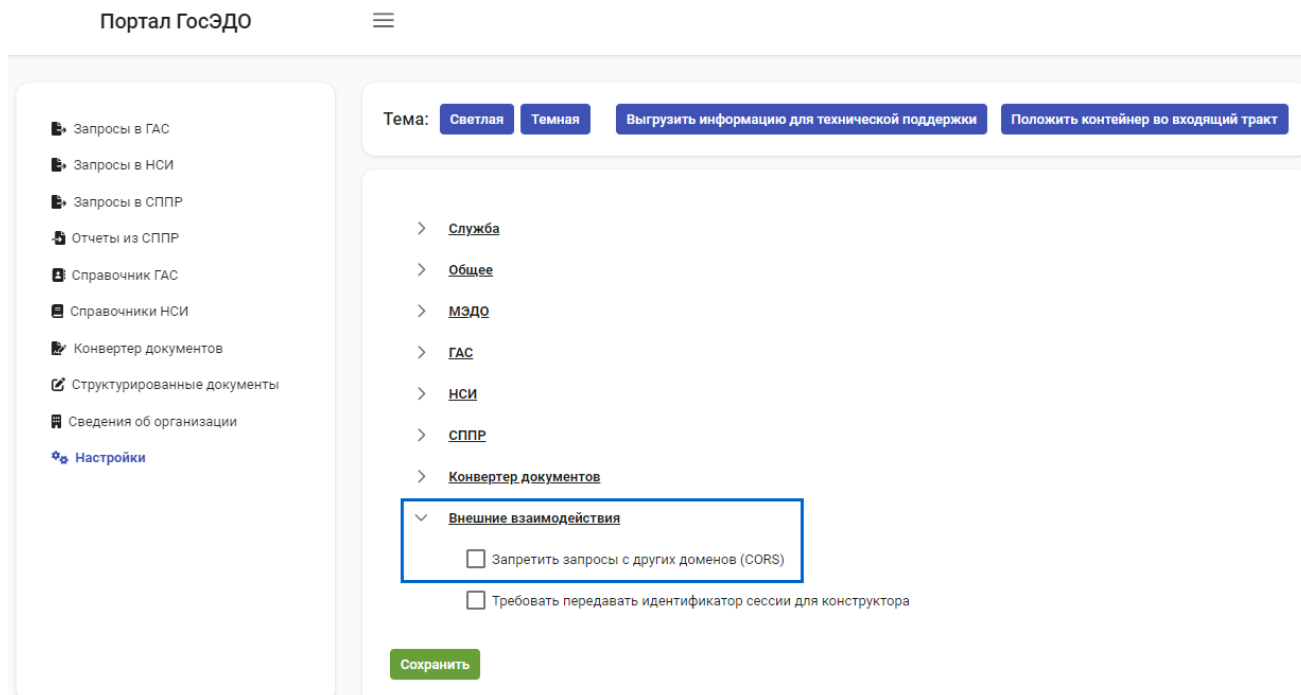
структурированных данных необходимо передавать в форму через API наименование и GUID организации (подробнее см. в разделе 5.5 Передача контекста для формы редактирования);

- заполнить поля «**Наименование организации**» и «**GUID адрес организации**» сведениями одной организации, которая использует этот экземпляр Программы (Рисунок 3.2), тогда при формировании данных Программой эти сведения будут автоматически использованы для заполнения полей об авторе и отправителе документа в структурированных данных.



**Рисунок 3.2. Пример заполненных данных об организации в разделе «Настройки» → «МЭДО» Программы**

Для корректной работы запросов к API необходимо убедиться, что в Программе в разделе «**Настройки**» → «**Внешние взаимодействия**» снят маркер «**Запретить запросы с других доменов (CORS)**» (Рисунок 3.3).



**Рисунок 3.3. Раздел Программы для настройки внешних взаимодействий**

Также независимо от вида установки и настройки, необходимо обеспечить обновление справочников ГАС и НСИ на экземпляре Программы, к которой происходит обращение по API, одним из двух способов:

1. Подключить экземпляр Программы к транспортной шине МЭДО и получать обновленные версии справочников через запрос организатору из Программы.
2. Загружать актуальные справочники вручную через загрузку транспортных контейнеров с актуальными справочниками по кнопке «Положить во входящий тракт» раздела «Настройки» Программы. В этом случае транспортные контейнеры с актуальными версиями справочников (ГАС, НСИ) должны быть предварительно получены в другой программе, совместимой с сервисами НСИ и ГАС, или в Центре поддержки ГосЭДО.

## 4 Демонстрационный пример

Для демонстрации возможностей интеграции «Портала ГосЭДО» в СЭД разработана специальная веб-страница, изображающая интерфейс СЭД. Вы можете загрузить её по ссылке: [ПорталГосЭДО\\_API.html](https://portal.gosedo.ru/API.html).

Серая часть страницы (1) имитирует вид карточки документа в СЭД, центральная часть страницы (2) является блоком iframe, где будет отображена интеграция «Портала ГосЭДО» (Рисунок 4.1).

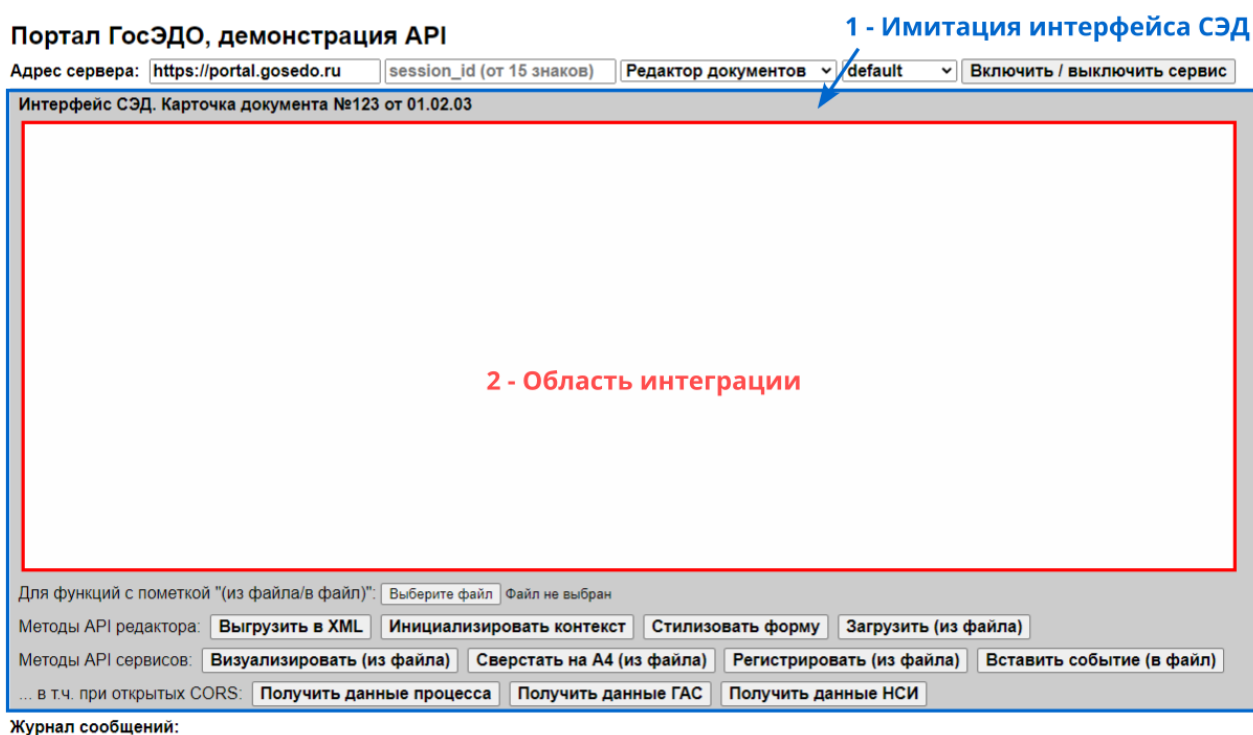
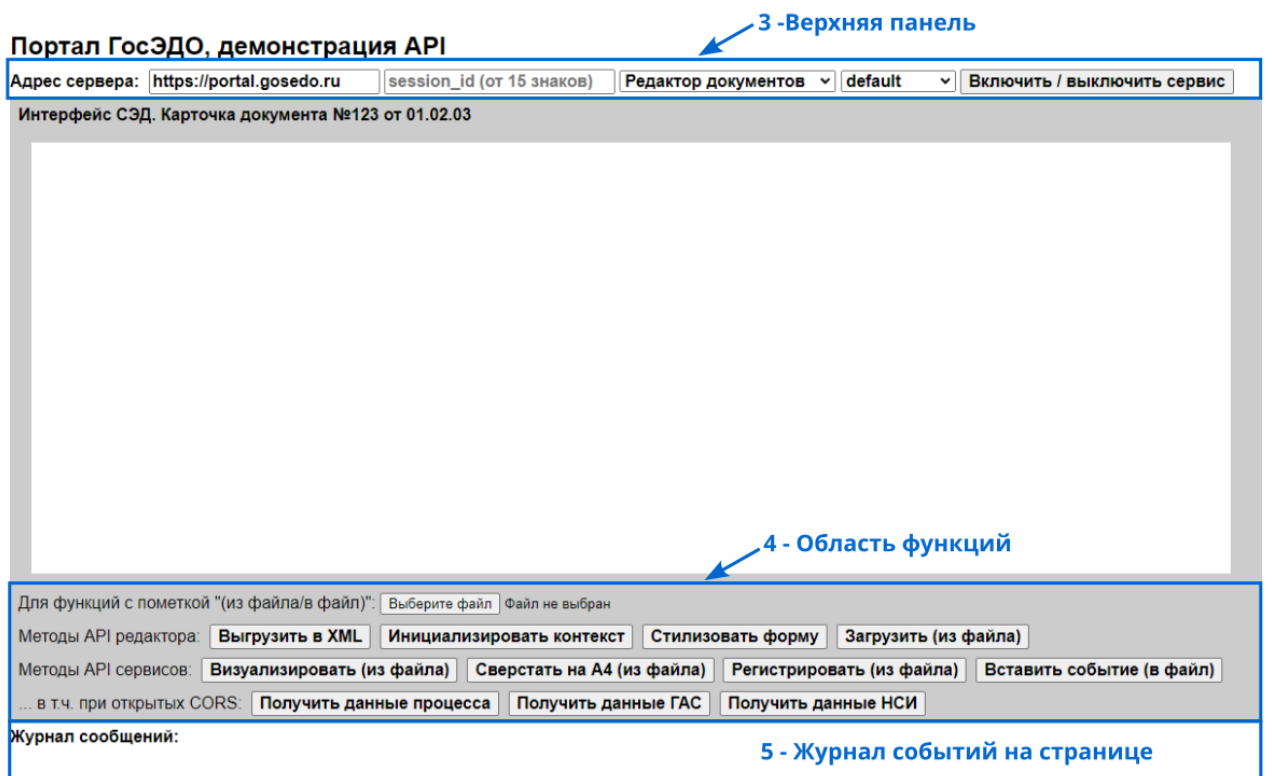


Рисунок 4.1. Страница демонстрации интеграции «Портала ГосЭДО» в СЭД

Другие функциональные области страницы обозначены на Рисунок 4.2. Верхняя панель (3) позволяет задать параметры подключения. Под областью интеграции расположена область функций (4) и область журналирования событий на странице (5).





**Рисунок 4.2. Функциональные области страницы демонстрации**

На верхней панели доступны следующие параметры подключения:

- **«Адрес сервера»** - адрес до экземпляра «Портала ГосЭДО», к которому будет выполняться обращение в рамках текущей демонстрации. В целях демонстрации может быть использован «Портал ГосЭДО», доступный по адресу [«https://portal.gosedo.ru»](https://portal.gosedo.ru) или можно указать адрес до локального экземпляра «Портала ГосЭДО» (например [«http://10.77.142.22:84»](http://10.77.142.22:84)).
- **«session\_id»** - идентификатор сессии, в рамках которой организована изоляция данных (необязательно). Позволяет изолировать данные документов, с которыми выполняется работа в рамках одного идентификатора, как если бы пользователь работал в отдельном экземпляре «Портала ГосЭДО». Разграничение на сессии позволяет, например, изолировать данные пользователей из разных организаций друг от друга на одном экземпляре «Портала ГосЭДО».

- **«Редактор документов»** - выбор раздела «Портала ГосЭДО», который будет выведен в iframe. Доступные значения:
  - «Редактор документов» - если выбрано это значение, после запуска сервиса в iframe будет выведена форма подготовки и редактирования структурированных данных;
  - «Конвертер документов» - если выбрано это значение, после запуска сервиса в iframe будет выведен раздел «Конвертер документов» «Портала ГосЭДО»;
  - «Запросы в ГАС» - если выбрано это значение, после запуска сервиса в iframe будет выведен раздел «Запросы в ГАС» «Портала ГосЭДО»;
  - «Запросы в НСИ» - если выбрано это значение, после запуска сервиса в iframe будет выведен раздел «Запросы в НСИ» «Портала ГосЭДО»;
  - «Запросы в СППР» - если выбрано это значение, после запуска сервиса в iframe будет выведен раздел «Запросы в СППР» «Портала ГосЭДО».
- **«default»** - выбор цветовой темы оформления, которая будет применена при выводе интерфейса «Портала ГосЭДО»: «dark» - темная тема оформления, «light» - светлая тема оформления.
- **«Включить/выключить сервис»** - кнопка запуска подключения и вывода в iframe выбранного раздела «Портала ГосЭДО». Повторное нажатие выполнит отключение от сервиса.

В разделах далее вы можете ознакомиться со следующей информацией:

- демонстрация интеграции редактора в интерфейс СЭД;
- пример заполнения данных и передача контекста в редактор;
- демонстрация выгрузки данных из формы в XML;
- демонстрация загрузки данных из файла XML в редактор;
- демонстрация передачи стилей в редактор;
- демонстрация получения HTML-визуализации из файла XML;
- демонстрация создания печатной формы PDF из файла XML;

- демонстрация регистрации цифрового документа XML во внутреннем модуле координации;
- демонстрация получения информации о состоянии цифрового процесса;
- демонстрация вставки события регистрации в файл XML;
- демонстрация получения данных ГАС;
- демонстрация получения данных НСИ.

#### **4.1 Демонстрация интеграции редактора в интерфейс СЭД**

Для запуска демонстрации интеграции редактора цифровых документов необходимо на демонстрационной веб-странице (Рисунок 4.3):

1. В поле **«Адрес сервера»** ввести веб-адрес экземпляра «Портала ГосЭДО», к которому будет выполняться обращение (например **«<https://portal.gosedo.ru/>»**, или **«http://10.77.142.22:84/»**).
2. Убедиться, что для вывода выбран раздел **«Редактор документов»**.
3. Выбрать тему оформления в следующем поле, например для светлой темы оформления - **«light»**.
4. Нажать кнопку **«Включить/выключить сервис»**.

В центральной части окна отобразится форма подготовки структурированных данных.

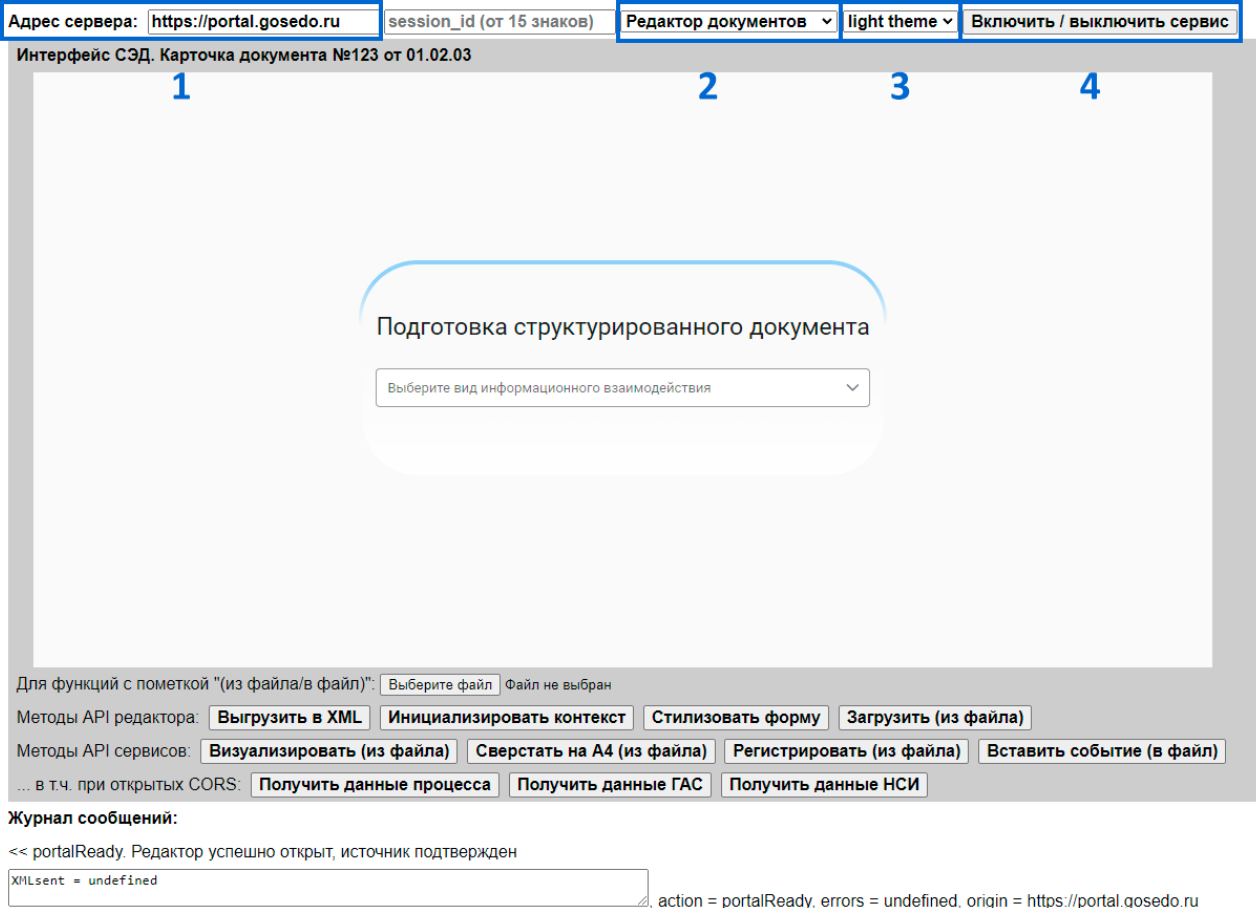


Рисунок 4.3. Процесс запуска демонстрации интеграции редактора в интерфейс СЭД

## 4.2 Пример заполнения структурированных данных и передача контекста в редактор

Запустите сервис в режиме «Редактор документов» на демонстрационной веб-странице, как описано в разделе 4.1 Демонстрация интеграции редактора в интерфейс СЭД.

В качестве примера будет рассмотрена подготовка структурированных данных документа вида «Служебное письмо». Данные по другим видам документов формируются по аналогичному принципу.

В форме «Подготовка структурированного документа» выберите:

- вид информационного взаимодействия «Служебная переписка»;
- вид события «Отправка письма»;
- вид документа «Служебное письмо» (Рисунок 4.4).

## Портал ГосЭДО, демонстрация API

Адрес сервера:   Редактор документов  Включить / выключить сервис

Интерфейс СЭД. Карточка документа №123 от 01.02.03

### Подготовка структурированного документа

Для функций с пометкой "(из файла/в файл)":

Методы API редактора:

Методы API сервисов:

... в т.ч. при открытых CORS:

**Журнал сообщений:**

```
<< portalReady. Редактор успешно открыт, источник подтвержден  
XMLsent = undefined
```

action = portalReady, errors = undefined, origin = https://portal.gosedo.ru

### Рисунок 4.4. Выбор вида документа в форме подготовки структурированных данных

Нажмите кнопку «Заполнить данные». Откроется форма заполнения структурированных данных по выбранному виду документа (Рисунок 4.5).

## Портал ГосЭДО, демонстрация API

Адрес сервера:   Редактор документов

Интерфейс СЭД. Карточка документа №123 от 01.02.03

Для функций с пометкой "(из файла/в файл)":

Методы API редактора:

Методы API сервисов:

... в т.ч. при открытых CORS:

Журнал сообщений:

```
<< portalReady. Редактор успешно открыт, источник подтвержден  
XMLsent = undefined
```

action = portalReady, errors = undefined, origin = https://portal.gosedo.ru

**Рисунок 4.5. Форма заполнения структурированных данных по виду документа «Служебное письмо»**

Заполните поле **«Получатель»**, нажав кнопку **«Выбрать участника»** и выбрав в открывшемся списке **«Аппарат Правительства РФ»**.

Заполните поле **«Текст сообщения»** введя текст документа, например: **«Сообщаем о выполнении заявленных работ в срок до конца 4 квартала текущего года»**.

## Портал ГосЭДО, демонстрация API

Адрес сервера:   Редактор документов

Интерфейс СЭД. Карточка документа №123 от 01.02.03

Сохранить данные Сбросить выбор Простой вид Полный вид

АгентыСобытия

АгентыПолучатели

Получатель

Аппарат Правительства РФ

+Получатель

ПредметыСобытия

ИсходноеСообщение

ТекстСообщения

Сообщаем о выполнении заявленных работ в срок до конца 4 квартала текущего года.

+СвязанныеДокументы

+ПриложенияДокумента

Для функций с пометкой "(из файла/в файл)":

Методы API редактора:

Методы API сервисов:

... в т.ч. при открытых CORS:

Журнал сообщений:

```
<< portalReady. Редактор успешно открыт, источник подтвержден
XMLsent = undefined
action = portalReady, errors = undefined, origin = https://portal.gosedo.ru
```

Рисунок 4.6. Пример заполнения данных документа вида «Служебное письмо»

Нажмите кнопку «Инициализировать контекст» (Рисунок 4.7). В журнале событий будет отражено, что в форму переданы GUID организации-автора документа, наименование организации, список приложений и список сотрудников.

Для функций с пометкой "(из файла/в файл)":

Методы API редактора:

Методы API сервисов:

... в т.ч. при открытых CORS:

Журнал сообщений:

```
>> initData() {"session_id":"","organization_guid":"b23a5a35-dde3-48ef-aa07-621eb56417db","organization_title":"Минфин России","attachments":
[{"id":"d0725f36","name":"проект.docx","description":"проект договора"},{"id":"f09daf08","name":"схема.pdf","description":"схема проезда"}],"users":[{"id":1,"fio":"Иванов Иван
Иванович"}, {"id":2,"fio":"Петров Петр Петрович"}, {"id":3,"fio":"Сидоров Сидор Сидорович"}]}
```

Рисунок 4.7. Журнал сообщений с информацией о передаче контекста в редактор

Нажмите «+Приложения документа», раскройте значения в поле «СсылкаИД». Убедитесь, что среди доступных для выбора значений появился

список приложений (проект.docx, схема.pdf), переданных **при инициации** контекста. Выберите, для примера, «проект.docx» (Рисунок 4.8).

ТекстСообщения

Сообщаем о выполнении заявленных работ в срок до конца 4 квартала текущего года.

+СвязанныеДокументы

ПриложенияДокумента - Удалить блок

ПриложениеДокумента - Удалить блок

СсылкаИД

Значение не выбрано

Значение не выбрано

+Приложение

проект.docx

схема.pdf

**Рисунок 4.8.** Доступные значения поля «ПриложениеДокумента» → «СсылкаИД» после передачи контекста в редактор

### 4.3 Демонстрация выгрузки данных из формы в XML

Запустите сервис в режиме «**Редактор документов**» на демонстрационной веб-странице, заполните данные по виду документа «**Служебное письмо**» и передайте контекст документа как указано в разделе 4.2 Пример заполнения структурированных данных и передача контекста в редактор.

Для выгрузки данных из формы редактирования в виде файла структурированных данных XML нажмите «**Выгрузить в XML**» (Рисунок 4.9).



## Портал ГосЭДО, демонстрация API

Адрес сервера: <https://portal.gosedo.ru> session\_id (от 15 знаков) Редактор документов light theme Включить / выключить сервис

Интерфейс СЭД. Карточка документа №123 от 01.02.03

Для функций с пометкой "(из файла/в файл)": Выберите файл Файл не выбран

Методы API редактора: **Выгрузить в XML** Инициализировать контекст Стилизовать форму Загрузить (из файла)

Методы API сервисов: Визуализировать (из файла) Сверстать на А4 (из файла) Регистрировать (из файла) Вставить событие (в файл)

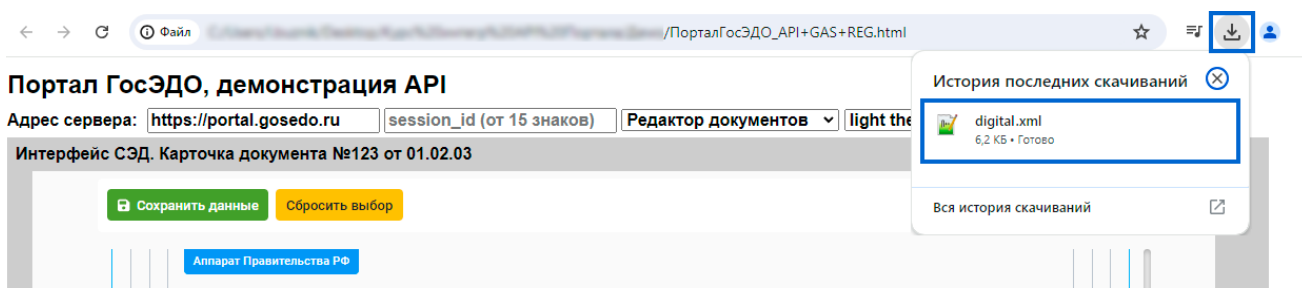
... в т.ч. при открытых CORS: Получить данные процесса Получить данные ГАС Получить данные НСИ

### Журнал сообщений:

```
>> initData() {"session_id":"","organization_guid":"b23a5a35-dde3-48ef-aa07-621eb56417db","organization_title":"Минфин России","attachments": [{"id":"d0725f36","name":"проект.docx","description":"проект договора"}, {"id":"f09daf08","name":"схема.pdf","description":"схема проезда"}], "users": [{"id":1,"fio":"Иванов Иван"}]}
```

### Рисунок 4.9. Кнопка выгрузки структурированных данных в файл XML

Начнется загрузка файла **digital.xml** (Рисунок 4.10), содержащего структурированные данные цифрового документа, заполненные в форме.



### Рисунок 4.10. Результат выгрузки данных формы в файл XML

Файл **digital.xml** можно открыть на просмотр с помощью текстового редактора (Рисунок 4.11).



```
digital.xml – Блокнот
Файл Правка Формат Вид Справка
<?xml version="1.0" encoding="UTF-8"?>
<doc:Документ xmlns:doc="urn:Doc01-004-00001:DocumentMeta:v0.1.0" xmlns:cdm="urn:Adm01-002:CoreDomainMeta:v0.1.0" xmlns:exc="urn:Exc01-004-00003:ExchangeMeta:v0.1.0" xmlns:adm="urn:Adm01-001:AnyDomainMeta:v0.1.0" xmlns:mdm="urn:Adm01-003:MasterDomainMeta:v0.1.0" xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:exc1="urn:Exc01-004-00001:ExchangeMeta:v0.1.0" xmlns:sdm004="urn:Adm01-004:SubjectDomainMeta:v0.1.0" xmlns:exc2="urn:Exc01-004-00002:ExchangeMeta:v0.1.0" xmlns:idp230="urn:Doc01-004-00002:DocumentMeta:v0.1.0">
  <doc:ДанныеДокумента xml:id="ДанныеДокумента">
    <doc:ЗаголовокДокумента doc:ВидИД="Doc01-004-00001" doc:ВидНазвание="Служебное письмо">
      <cdm:СсылкаДокумента cdm:ДокументУИД="37c9ffa9-2026-4996-8371-b8504f046957" cdm:Представление="Служебное письмо от 2024-06-24 #37c9ffa9-2026-4996-8371-b8504f046957"/>
      <cdm:СсылкаСоздателя cdm:АгентУИД="b23a5a35-dde3-48ef-aa07-621eb56417db" cdm:Представление="Минфин России"/>
      <cdm:ВремяСоздания>2024-06-24T13:33:31+03:00</cdm:ВремяСоздания>
      <cdm:СодержитСведенияДСП>Нет</cdm:СодержитСведенияДСП>
      <doc:ТребуетсяПодписания>Да</doc:ТребуетсяПодписания>
      <doc:ТребуетсяРегистрация>Да</doc:ТребуетсяРегистрация>
    </doc:ЗаголовокДокумента>
    <doc:СообщенияДокумента>
      <exc:ОтправкаПисьма>
        <exc:ИнформацияОПроцессе exc:ВидИД="Exc01-004-00003" exc:ВидНазвание="Служебная переписка">
          <cdm:СсылкаПроцесса cdm:ПроцессУИД="9069dc96-113f-4969-9b63-ace516214b58" cdm:Представление="Служебная переписка от 2024-06-24 #9069dc96-113f-4969-9b63-ace516214b58"/>
          <cdm:СсылкаСоздателя cdm:АгентУИД="b23a5a35-dde3-48ef-aa07-621eb56417db" cdm:Представление="Минфин России"/>
          <cdm:ВремяСоздания>2024-06-24T13:33:31+03:00</cdm:ВремяСоздания>
        </exc:ИнформацияОПроцессе>
        <exc:ИнформацияОСобытии exc:ВидИД="Exc01-004-00003-Event0001" exc:ВидНазвание="Отправка письма">
          <cdm:СсылкаСобытия cdm:СобытиеУИД="2a88912d-4e91-43e6-a1fd-3088e78d7504" cdm:Представление="Отправка письма от 2024-06-24 #2a88912d-4e91-43e6-a1fd-3088e78d7504"/>
          <cdm:ВремяСобытия>2024-06-24T13:33:31+03:00</cdm:ВремяСобытия>
          <exc:СтатусСобытия>Инициация</exc:СтатусСобытия>
        </exc:ИнформацияОСобытии>
      </exc:СообщенияДокумента>
    </doc:СообщенияДокумента>
  </doc:ДанныеДокумента>
</doc:Документ>
```

**Рисунок 4.11. Просмотр файла структурированных данных XML, полученный в результате сохранения данных из формы**

С примером файла, полученного в результате выполнения указанных действий, можно ознакомиться по ссылке: [digital.xml](#).

## 4.4 Демонстрация загрузки данных из файла XML в редактор

Запустите сервис в режиме **«Редактор документов»** на демонстрационной веб-странице, как описано в разделе 4.1 Демонстрация интеграции редактора в интерфейс СЭД.

Если сервис уже запущен и заполнены данные в редакторе, рекомендуется очистить форму нажав кнопку **«Сбросить выбор»**, чтобы в полной мере увидеть демонстрацию загрузки данных в редактор.

Нажмите кнопку **«Выберите файл»** под формой и на локальном компьютере выберите файл **digital.xml**, который был получен ранее в результате выгрузки данных из формы, как было описано в разделе 4.3 Демонстрация выгрузки данных из формы в XML.

Нажмите кнопку **«Загрузить в форму (из файла)»** (Рисунок 4.12).

## Портал ГосЭДО, демонстрация API

Адрес сервера:   Редактор документов

Интерфейс СЭД. Карточка документа №123 от 01.02.03

Подготовка структурированного документа

12

Для функций с пометкой "(из файла/в файл)":

Методы API редактора:

Методы API сервисов:

... в т.ч. при открытых CORS:

### Журнал сообщений:

<< XMLSend. Нажата кнопка сохранения данных, скачивается XML

```
XMLsent = <?xml version="1.0" encoding="UTF-8"?>
<doc:Документ xmlns:doc="urn:Doc01-004-00001:DocumentMeta:v0.1.0"
action = XMLSend, errors = , origin = https://portal.gosedo.ru
```

**Рисунок 4.12. Порядок действий при демонстрации загрузки данных из файла XML в редактор**

В редакторе откроется форма по переданному в функцию виду документа, а поля будут заполнены сведениями из файла **digital.xml** (Рисунок 4.13).

## Портал ГосЭДО, демонстрация API

Адрес сервера: <https://portal.gosedo.ru> session\_id (от 15 знаков) Редактор документов light theme Включить / выключить сервис

Интерфейс СЭД. Карточка документа №123 от 01.02.03

Для функций с пометкой "(из файла/в файл)": Выберите файл digital.xml

Методы API редактора: **Выгрузить в XML** Инициализировать контекст Стилизовать форму Загрузить (из файла)

Методы API сервисов: **Визуализировать (из файла)** Сверстать на A4 (из файла) Регистрировать (из файла) Вставить событие (в файл)

... в т.ч. при открытых CORS: **Получить данные процесса** Получить данные ГАС Получить данные НСИ

Журнал сообщений:

>> setIframesContent() - Передача файла в форму ввода...

Рисунок 4.13. Результат загрузки данных из файла XML в редактор

## 4.5 Демонстрация передачи стилей в редактор

Запустите сервис в режиме «Редактор документов» на демонстрационной веб-странице, откройте форму редактирования данных как описано в разделе 4.2. Пример заполнения структурированных данных и передача контекста в редактор.

Нажмите кнопку «**Стилизовать форму**».

Убедитесь что в журнале появилась запись «`addStyleTag() .container{width: 800pt !important;}`», а ширина формы визуально соответствует переданному стилю в 800 пикселей.

Нажмите кнопку «**Стилизовать форму**» еще раз (Рисунок 4.14).

Убедитесь что в журнале появилась запись «`addStyleTag() .container{width: 500pt !important;}`», а ширина формы визуально стала соответствовать переданному стилю в 500 пикселей.

## Портал ГосЭДО, демонстрация API

Адрес сервера:   Редактор документов

Интерфейс СЭД. Карточка документа №123 от 01.02.03

АгентыПолучатели

Получатель

ПредметыСобытия

ИсходноеСообщение

ТекстСообщения

Сообщаем о выполнении заявленных работ в срок до конца 4 квартала текущего года.

ПриложенияДокумента

ПриложениеДокумента

1

Для функций с пометкой "(из файла/в файл)":  digital.xml

Методы API редактора:

Методы API сервисов:

... в т.ч. при открытых CORS:

Журнал сообщений:

```
>> addStyleTag() .container{width: 500pt !important;}
>> addStyleTag() .container{width: 800pt !important;}
```

2

Рисунок 4.14. Порядок действий при демонстрации передачи стилей в редактор

## 4.6 Демонстрация получения HTML-визуализации из файла XML

Убедитесь, что на демонстрационной веб-странице на верхней панели введён корректный адрес «Портала ГосЭДО» для обращения к нему по API, например: «<https://portal.gosedo.ru/>».

В нижней части формы нажмите «**Выберите файл**» и выберите на локальном компьютере файл **digital.xml**, который был получен ранее в результате выгрузки данных из формы, как было описано в разделе 4.3 Демонстрация выгрузки данных из формы в XML.

Нажмите кнопку «**Визуализировать (из файла)**».

Начнется скачивание файла **document.html**, который является HTML-визуализацией документа на основе данных входного файла digital.xml (Рисунок 4.15).

Портал ГосЭДО, демонстрация API

Адрес сервера: <https://portal.gosedo.ru> session\_id (от 15 знаков) Редактор документов light the

Интерфейс СЭД. Карточка документа №123 от 01.02.03

Сохранить данные Сбросить выбор

АгентыПолучатели

Получатель

Аппарат Правительства РФ

+Получатель

ПредметыСобытия

ИсходноеСообщение

ТекстСообщения

Сообщаем о выполнении заявленных работ в срок до конца 4 квартала текущего года.

+СвязанныеДокументы

ПриложенияДокумента - Удалить блок

ПриложениеДокумента

Для функций с пометкой "(из файла/в файл)": Выберите файл digital.xml

Методы API редактора: Выгрузить в XML Инициализировать контекст Стилизовать форму Загрузить (из файла)

Методы API сервисов: Визуализировать (из файла) Сверстать на А4 (из файла) Регистрировать (из файла) Вставить событие (в файл)

... в т.ч. при открытых CORS: Получить данные процесса Получить данные ГАС Получить данные НСИ

Журнал сообщений: 2

<< response == 200 OK saving... document.html

{"xml": {}, "action": "convertXmlToHtml"}

История последних скачиваний

document.html  
2 540 Б • 2 минуты назад

Вся история скачиваний 3

**Рисунок 4.15. Порядок действий для демонстрации получения HTML-визуализации из файла XML**

Откройте на просмотр **document.html** с помощью браузера (Рисунок 4.16).

## Служебное письмо

Автор документа: Минфин России

УИД документа: 37c9ffa9-2026-4996-8371-b8504f046957

Время создания: 2024-06-24 в 13:33:31+03:00

### Отправка письма

Агенты-получатели:

Получатель: Аппарат Правительства РФ

Исходное сообщение >> Отправлено

ТекстСообщения: Сообщаем о выполнении заявленных работ в срок до конца 4 квартала текущего года.

ПриложенияДокумента:

ПриложениеДокумента:

СсылкаИД: enclosure\_d0725f36.docx

Событие: Отправка письма

УИД события: 2a88912d-4e91-43e6-a1fd-3088e78d7504

Время события: 2024-06-24 в 13:33:31+03:00

Процесс: Служебная переписка

УИД процесса: 9069dc96-113f-4969-9b63-ace516214b58

Время создания: 2024-06-24, Минфин России

### Приложения:

проект.docx (проект договора)

схема.pdf (схема проезда)

#### Рисунок 4.16. Просмотр HTML-визуализации в браузере

С примером файла, полученного в результате выполнения указанных действий, можно ознакомиться по ссылке: [document.html](#).

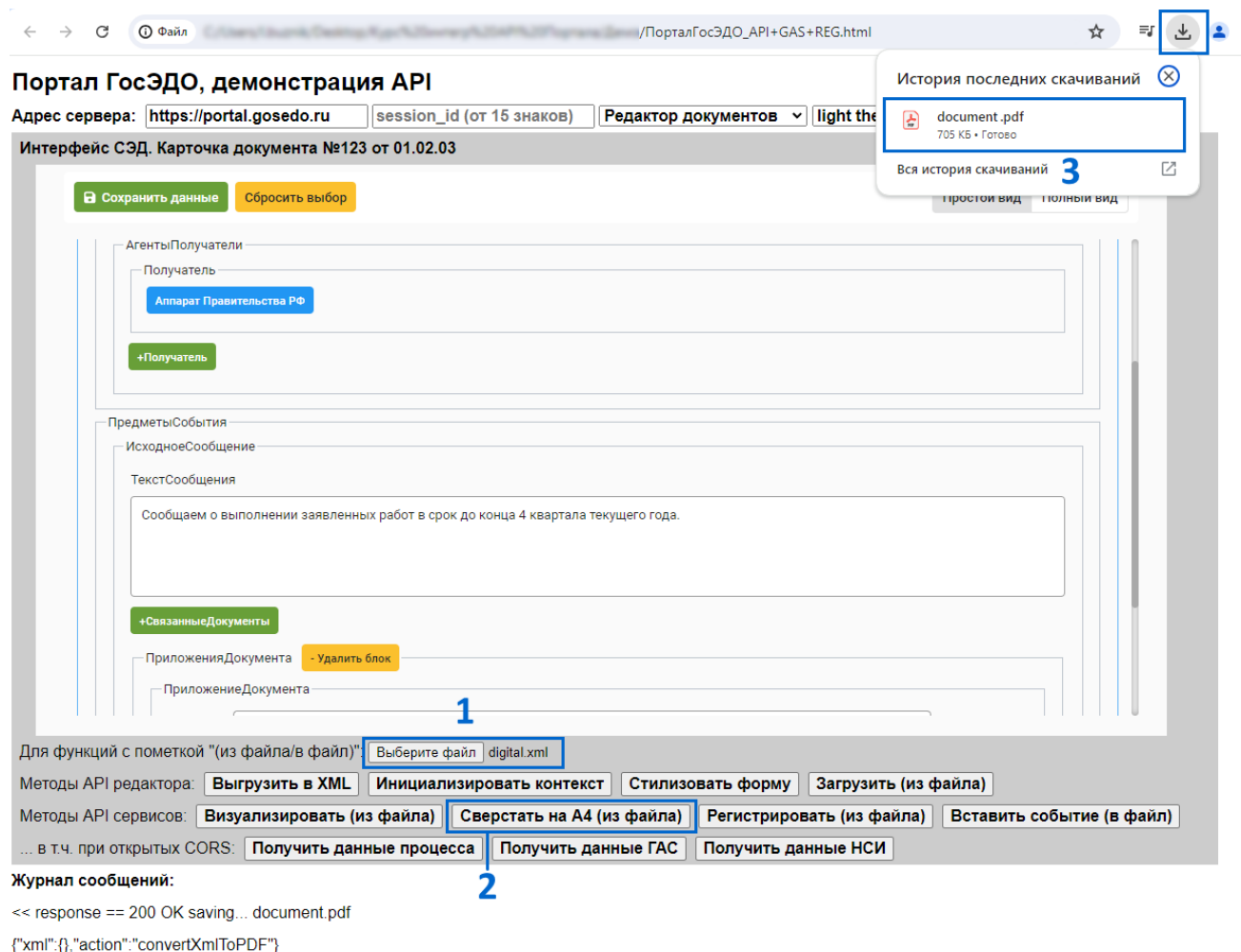
## 4.7 Демонстрация создания печатной формы PDF из файла XML

Убедитесь, что на демонстрационной веб-странице на верхней панели введён корректный адрес «Портала ГосЭДО» для обращения к нему по API, например: «<https://portal.gosedo.ru/>».

В нижней части формы нажмите «**Выберите файл**» и выберите на локальном компьютере файл **digital.xml**, который был получен ранее в результате выгрузки данных из формы, как было описано в разделе 4.3 Демонстрация выгрузки данных из формы в XML.

Нажмите кнопку «Сверстать на А4 (из файла)».

Начнется загрузка на компьютер файла **document.pdf**, который является печатной формой документа PDF, сформированного на основе данных файла digital.xml (Рисунок 4.17).



**Рисунок 4.17. Порядок действий для демонстрации создания печатной формы PDF из файла XML**

Откройте на просмотр файл **document.pdf** с помощью браузера или другой программы для просмотра PDF (Рисунок 4.18).



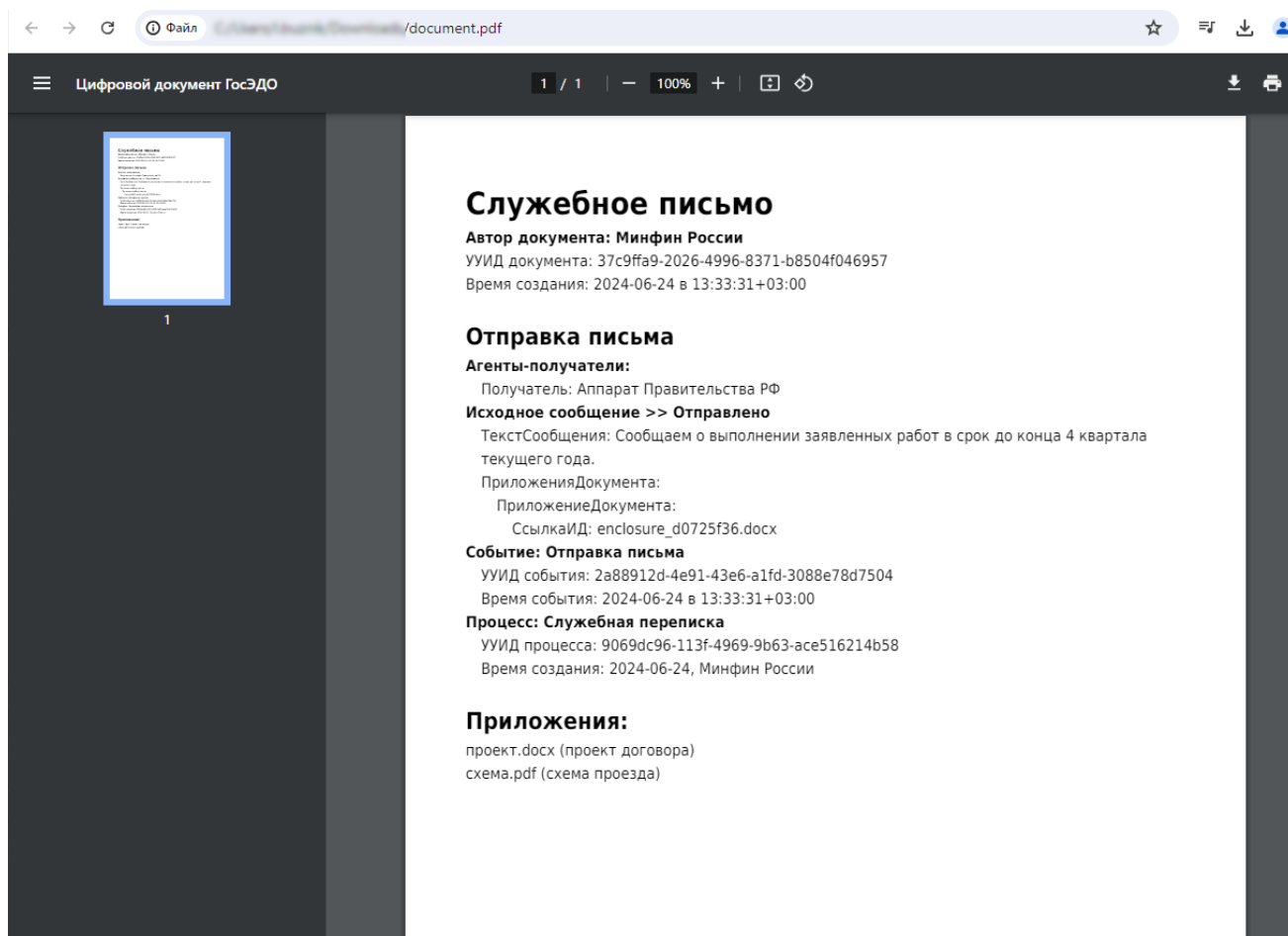


Рисунок 4.18. Просмотр файла PDF, сформированного на основании входного файла digital.xml

С примером файла, полученного в результате выполнения указанных действий, можно ознакомиться по ссылке: document.pdf.

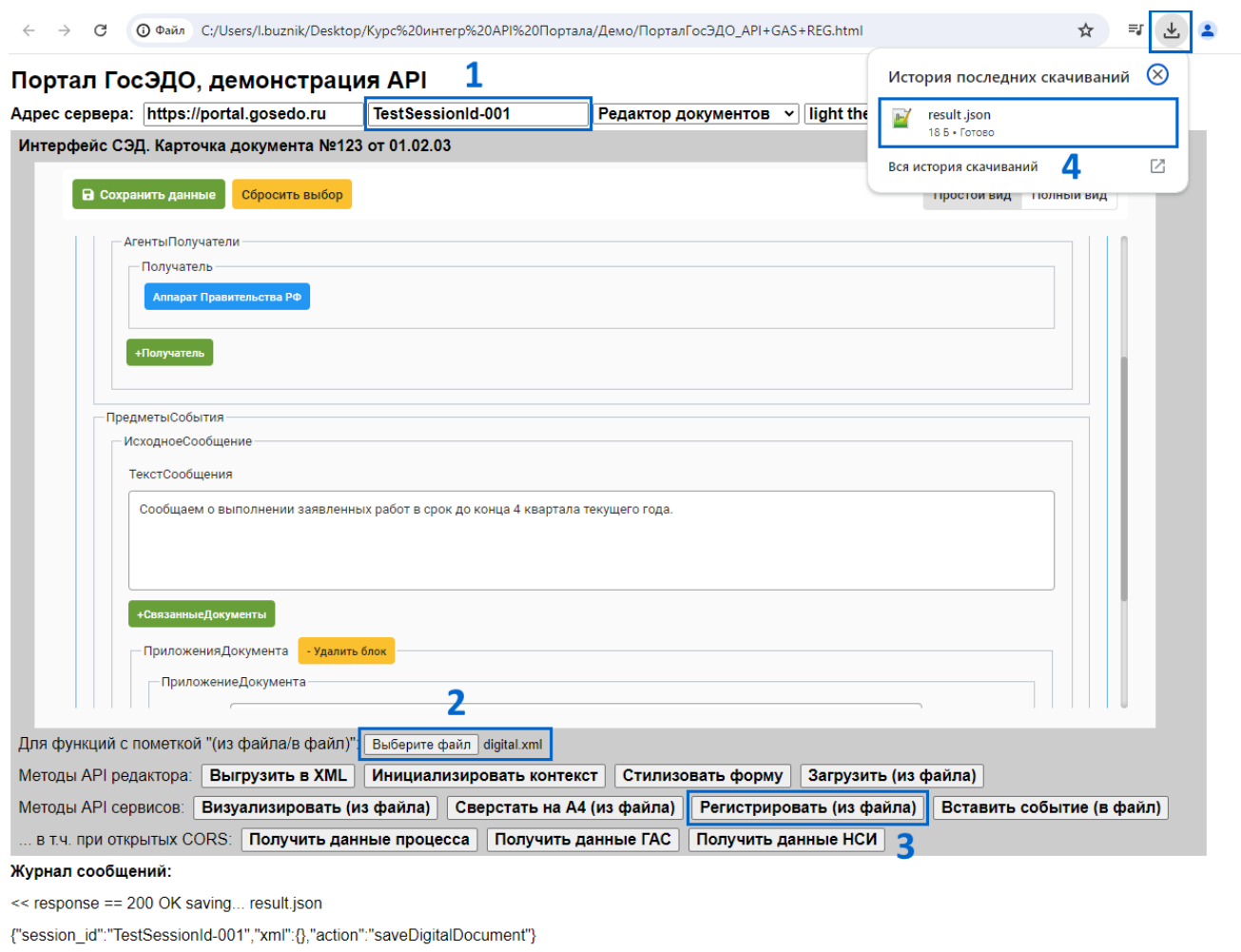
## 4.8 Демонстрация регистрации цифрового документа XML во внутреннем модуле координации

Убедитесь, что на демонстрационной веб-странице на верхней панели введён корректный адрес «Портала ГосЭДО» для обращения к нему по API, например: «<https://portal.gosedo.ru/>» и в поле «session\_id» введён идентификатор сессии не менее 16 символов, например: «TestSessionId-001».

В нижней части формы нажмите «**Выберите файл**» и выберите на локальном компьютере файл **digital.xml**, который был получен ранее в результате выгрузки данных из формы, как было описано в разделе 4.3 Демонстрация выгрузки данных из формы в XML.

Нажмите кнопку «**Регистрировать (из файла)**».

Начнется загрузка на компьютер файла **result.json**, который содержит ответ функции в формате JSON (Рисунок 4.19).



**Рисунок 4.19. Порядок действий для демонстрации регистрации цифрового документа XML во внутреннем модуле координации**

Откройте на просмотр файл **result.json** с помощью текстового редактора и убедитесь, что файл содержит данные об успешной регистрации цифрового документа во внутреннем модуле координации «{"success": "true"}» (Рисунок 4.20).

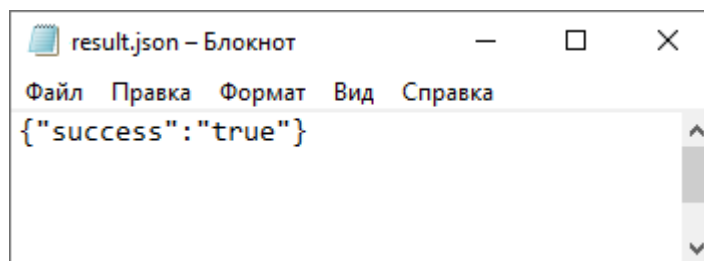


Рисунок 4.20. Содержимое файла **result.json**

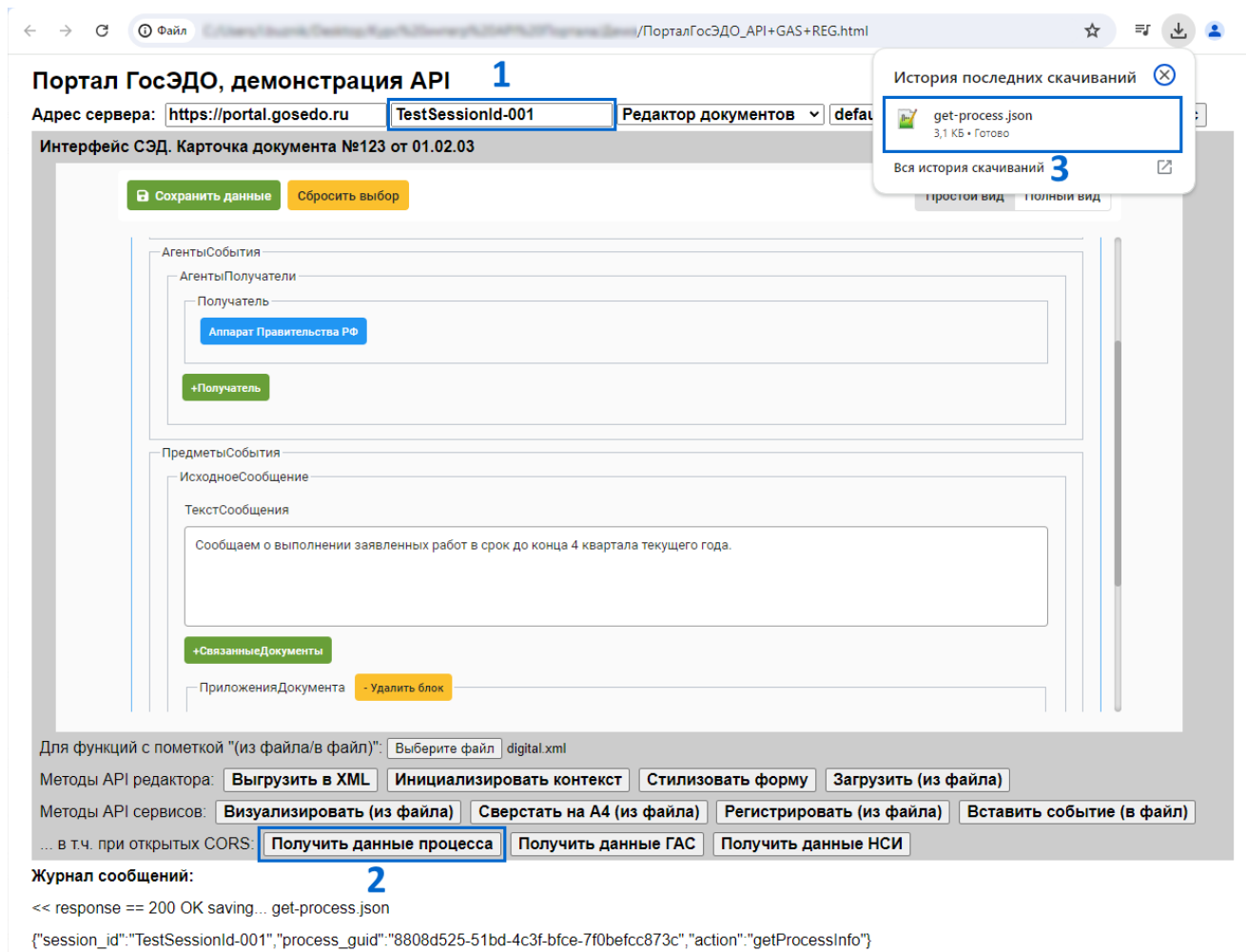
С примером файла, полученного в результате выполнения указанных действий, можно ознакомиться по ссылке: [result.json](#).

## 4.9 Демонстрация получения информации о состоянии цифрового процесса

Убедитесь, что на демонстрационной веб-странице на верхней панели введён корректный адрес «Портала ГосЭДО» для обращения к нему по API, например: «<https://portal.gosedo.ru/>» и в поле «session\_id» введён идентификатор сессии не менее 16 символов, например: «TestSessionId-001».

Нажмите кнопку **«Получить данные процесса»**.

Начнется скачивание файла **get-process.json**, который содержит ответ функции в формате JSON (Рисунок 4.21).



**Рисунок 4.21. Порядок действий для демонстрации функции получения данных о состоянии процесса**

Откройте на просмотр файл **get-process.json** с помощью текстового редактора и убедитесь, что файл содержит данные в формате JSON о процессе (Рисунок 4.22).



```
get-process.json - Блокнот
Файл Правка Формат Вид Справка
{"data":{"info":false,"events":[{"guid":"1d05dfef-7a8a-4355-ad1d-0fb68ab3a905","process_guid":"8808d525-51bd-4c3f-bfce-7f0befcc873c","event_type_id":"Exc01-004-00001-Event0001","event_type_name":"Направление исходящего","time":1717414292,"status":"Инициация","sender_guid":"1d04ca3e-df1a-0cb4-c325-6ef4003ca2ab","sender_title":"Минцифры России","receivers":[{"guid":"6d8c1ef5-a5ea-4dd9-a97d-5ee80f0663b1","name":"Аппарат Правительства РФ"}],"data_for_coordination":{"Исходящий":{"\РеквизитыДокумента\@ДокументУИД":{"time":1717414292,"value":"00000000-0000-0000-0000-000000000000"},"\/РеквизитыДокумента\@Представление":{"time":1717414292,"value":"-"},"\/РеквизитыДокумента\РегистрацияДокумента\ДатаРегистрации":{"time":1717414292,"value":"2024-06-03"},"\/РеквизитыДокумента\РегистрацияДокумента\НомерРегистрации":{"time":1717414292,"value":"Черновик"},"\/РеквизитыДокумента\РегистрацияДокумента\ОрганРегистрации\@АгентУИД":{"time":1717414292,"value":"00000000-0000-0000-0000-000000000000"},"\/РеквизитыДокумента\РегистрацияДокумента\ОрганРегистрации\@Представление":{"time":1717414292,"value":"-"},"document_guid":"0ffabc0e-dc60-4bbc-9117-07e02d793dc2"},"things":[{"proc_guid":"8808d525-51bd-4c3f-bfce-7f0befcc873c","thing_type":"Exc01-004-00001-Thing0001","owner_type":"Exc01-004-00001-Agent0001","thing_id":"1d04ca3e-df1a-0cb4-c325-6ef4003ca2ab"},"proc_guid":"8808d525-51bd-4c3f-bfce-7f0befcc873c","thing_type":"Exc01-004-00001-Thing0002","owner_type":"Exc01-004-00001-Agent0002","thing_id":"6d8c1ef5-a5ea-4dd9-a97d-5ee80f0663b1"},"things_data":[{"proc_guid":"8808d525-51bd-4c3f-bfce-7f0befcc873c","thing_type":"Exc01-004-00001-Thing0001","owner_type":"Exc01-004-00001-Agent0001","thing_id":"1d04ca3e-df1a-0cb4-c325-6ef4003ca2ab","prop_name":"РеквизитыДокумента","time":1717414292,"value":{"РеквизитыДокумента":{"@attributes":{"ДокументУИД":"00000000-0000-0000-0000-000000000000"},"Представление":"-"},"РегистрацияДокумента":{"ДатаРегистрации":"2024-06-03"},"НомерРегистрации":"Черновик"},"ОрганРегистрации":{"@attributes":{"АгентУИД":"00000000-0000-0000-0000-000000000000"},"Представление":"-"},"value_simple":null}},"things_statuses":[{"proc_guid":"8808d525-51bd-4c3f-bfce-7f0befcc873c","thing_type":"Exc01-004-00001-Thing0001","owner_type":"Exc01-004-00001-Agent0001","thing_id":"1d04ca3e-df1a-0cb4-c325-6ef4003ca2ab","time":1717414292,"value":"Документ отправлен"},"proc_guid":"8808d525-51bd-4c3f-bfce-7f0befcc873c","thing_type":"Exc01-004-00001-Thing0002","owner_type":"Exc01-004-00001-Agent0002","thing_id":"6d8c1ef5-a5ea-4dd9-a97d-5ee80f0663b1","time":1717414292,"value":"Запрошена регистрация"}]},"success":"true"}
```

Рисунок 4.22. Пример содержимого файла **get-process.json**

С примером файла, полученного в результате выполнения указанных действий, можно ознакомиться по ссылке: [get-process.json](#).

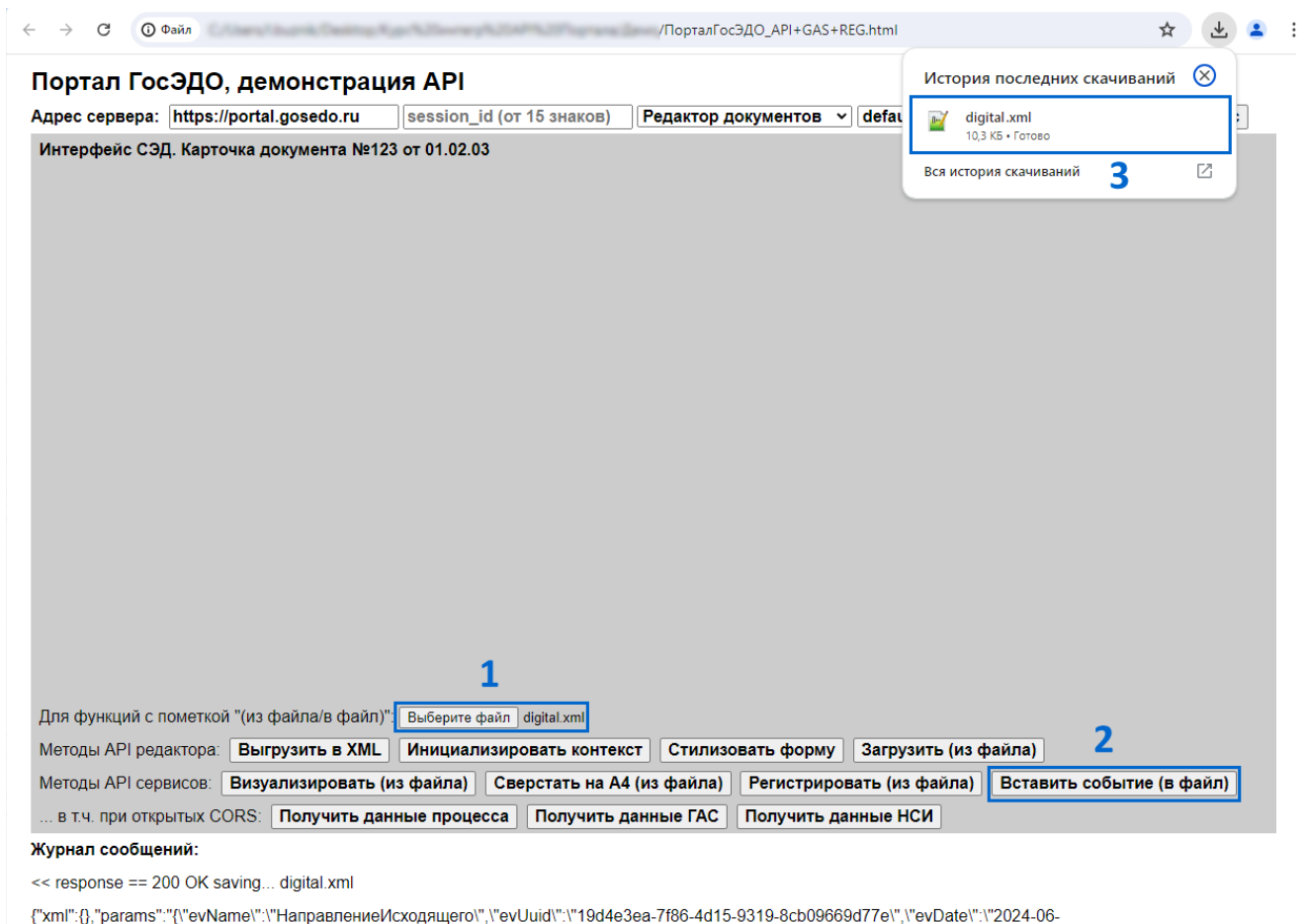
## 4.10 Демонстрация вставки события регистрации в файл XML

Убедитесь, что на демонстрационной веб-странице на верхней панели введён корректный адрес «Портала ГосЭДО» для обращения к нему по API, например: [«https://portal.gosedo.ru/»](https://portal.gosedo.ru/).

В нижней части формы нажмите **«Выберите файл»** и выберите на локальном компьютере файл **digital.xml**, который был получен ранее в результате выгрузки данных из формы, как было описано в разделе 4.3 Демонстрация выгрузки данных из формы в XML.

Нажмите кнопку **«Вставить событие (в файл)»**.

Начнется скачивание файла **digital.xml**, который является входящим файлом **digital.xml**, дополненным информацией о событии регистрации (Рисунок 4.23).



**Рисунок 4.23. Порядок действий для демонстрации вставки события регистрации в файл XML**

Откройте на просмотр файл **digital.xml** с помощью текстового редактора и убедитесь, что файл содержит данные о регистрации (Рисунок 4.24).

```

digital.xml – Блокнот
Файл  Правка  Формат  Вид  Справка
</doc:ПриложенияДокумента>
</doc:ДанныеДокумента>
<РегистрацияДокумента xmlns="urn:Doc01-004-00001:DocumentMeta:v0.1.0">
  <exc401:НаправлениеИсходящего xmlns:exc401="urn:Exc01-004-00001:ExchangeMeta:v0.1.0">
    <exc401:ИнформацияОПроцессе exc401:ВидИД="Exc01-004-00001" exc401:ВидНазвание="Официальная регистрация документа"><cdm:СсылкаПроцесса
cdm:ПроцессУИД="37c9ffa9-2026-4996-8371-b8504f046957" cdm:Представление="Регистрация от 2024-06-05 #12"/><cdm:СсылкаСоздателя
cdm:АгентУИД="1d04ca3e-df1a-0cb4-c325-6ef4003ca2ab" cdm:Представление="Минцифры России"/><cdm:ВремяСоздания>2024-06-
05T12:02:49+03:00</cdm:ВремяСоздания></exc401:ИнформацияОПроцессе>
    <exc401:ИнформацияОСобытии exc401:ВидИД="Exc01-004-00001-Event0001" exc401:ВидНазвание="Направление исходящего"><cdm:СсылкаСобытия
cdm:СобытиеУИД="19d4e3ea-7f86-4d15-9319-8cb09669d77e" cdm:Представление="Направление исходящего"/><cdm:ВремяСобытия>2024-06-
05T12:02:49+03:00</cdm:ВремяСобытия><exc401:СтатусСобытия>Инициация</exc401:СтатусСобытия></exc401:ИнформацияОСобытии>
    <exc401:АгентыСобытия>
      <exc401:АгентОтправитель><exc401:Отправитель exc401:ВидИД="Exc01-004-00001-Agent0001"
exc401:ВидНазвание="Отправитель"><cdm:СсылкаАгента cdm:АгентУИД="1d04ca3e-df1a-0cb4-c325-6ef4003ca2ab" cdm:Представление="Минцифры
России"/></exc401:Отправитель></exc401:АгентОтправитель>
      <exc401:АгентыПолучатели>
        <exc401:Получатель exc401:ВидИД="Exc01-004-00001-Agent0002" exc401:ВидНазвание="Получатель"><cdm:СсылкаАгента
cdm:АгентУИД="6d8c1ef5-a5ea-4dd9-a97d-5ee80f0663b1" cdm:Представление="Аппарат Правительства РФ"/></exc401:Получатель>
      </exc401:АгентыПолучатели>
    </exc401:АгентыСобытия>
    <exc401:ПредметыСобытия>
      <exc401:Исходящий exc401:ВидИД="Exc01-004-00001-Thing0001" exc401:ВидНазвание="Исходящий"><exc401:ВидВладельца exc401:ВидИД="Exc01-
004-00001-Agent0001" exc401:ВидНазвание="Отправитель"/><exc401:СостояниеПредмета>Документ
отправлен</exc401:СостояниеПредмета><exc401:ДанныеДляКоординации><exc401:РеквизитыДокумента cdm:ДокументУИД="19d4e3ea-7f86-4d15-9319-
8cb09669d77e" cdm:Представление="Регистрация от 2024-06-05 #12"><sdm004:РегистрацияДокумента><sdm004:ОрганРегистрации
cdm:АгентУИД="1d04ca3e-df1a-0cb4-c325-6ef4003ca2ab" cdm:Представление="Минцифры
России"/><sdm004:НомерРегистрации>12</sdm004:НомерРегистрации><sdm004:ДатаРегистрации>2024-06-
05</sdm004:ДатаРегистрации></sdm004:РегистрацияДокумента></exc401:РеквизитыДокумента></exc401:ДанныеДляКоординации></exc401:Исходящий>
      <exc401:Входящий exc401:ВидИД="Exc01-004-00001-Thing0002" exc401:ВидНазвание="Входящий"><exc401:ВидВладельца exc401:ВидИД="Exc01-
004-00001-Agent0002" exc401:ВидНазвание="Получатель"/><exc401:СостояниеПредмета>Запрещена
регистрация</exc401:СостояниеПредмета></exc401:Входящий>
    </exc401:ПредметыСобытия>
  </exc401:НаправлениеИсходящего>
</РегистрацияДокумента>
</doc:Документ>

```

Рисунок 4.24. Пример содержимого файла digital.xml после вставки события регистрации

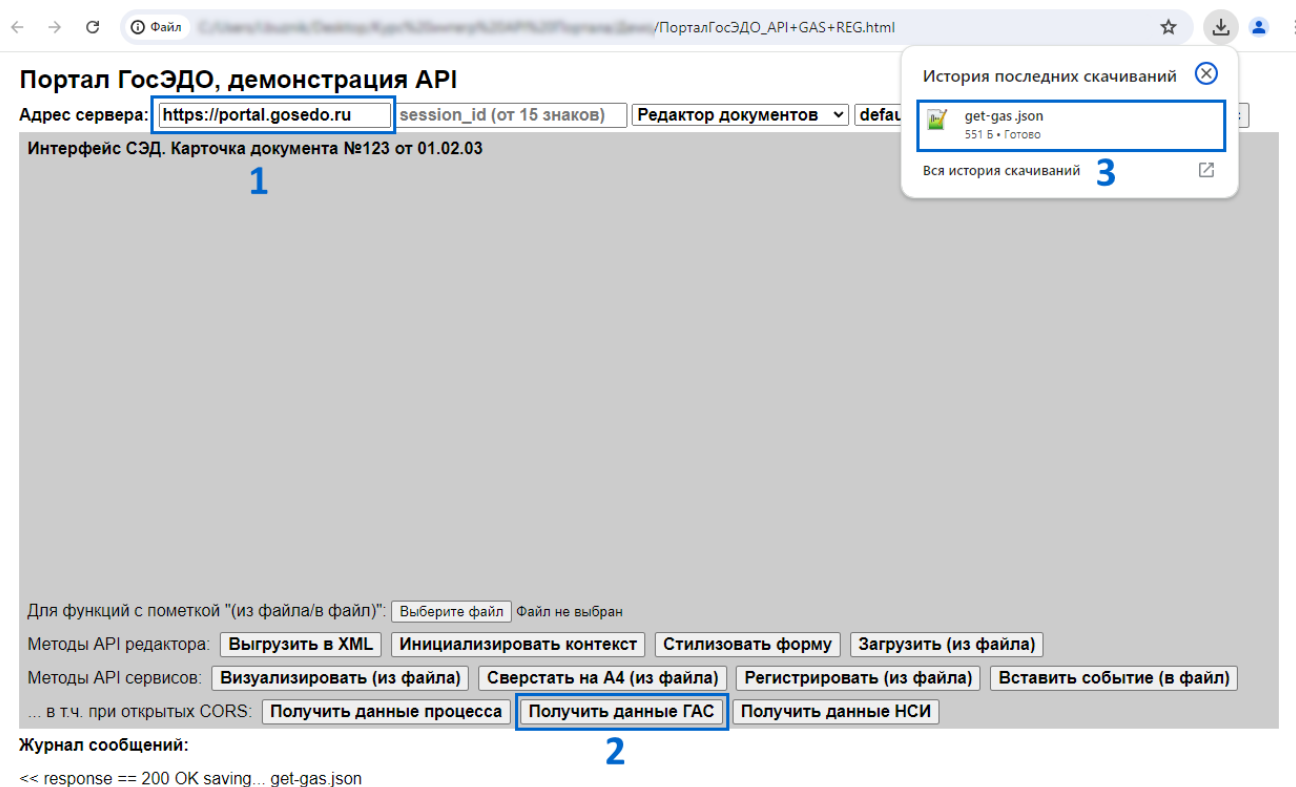
С примером файла, полученного в результате выполнения указанных действий, можно ознакомиться по ссылке: digital.xml.

## 4.11 Демонстрация получения данных ГАС

Убедитесь, что на демонстрационной веб-странице на верхней панели введён корректный адрес «Портала ГосЭДО» для обращения к нему по API, например: «<https://portal.gosedo.ru/>».

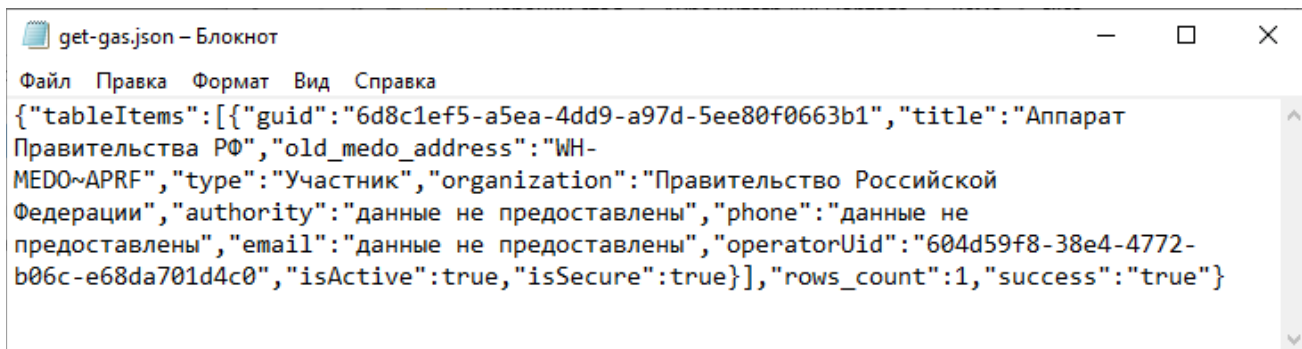
Нажмите кнопку «**Получить данные ГАС**».

Начнется скачивание файла **get-gas.json**, который содержит запрошенные данные ГАС (Рисунок 4.25).



**Рисунок 4.25. Порядок действий для демонстрации получения данных ГАС**

Откройте на просмотр файл **get-gas.json** с помощью текстового редактора и убедитесь, что файл содержит данные ГАС об организации согласно переданным параметрам (`filter: {title: 'Аппарат', }`) (Рисунок 4.26).



**Рисунок 4.26. Пример содержимого файла get-gas.json**

С примером файла, полученного в результате выполнения указанных действий, можно ознакомиться по ссылке: [get-gas.json](#).

## 4.12 Демонстрация получения данных НСИ

Убедитесь, что на демонстрационной веб-странице на верхней панели введён корректный адрес «Портала ГосЭДО» для обращения к нему по API, например: «<https://portal.gosedo.ru/>».



Нажмите кнопку «Получить данные НСИ».

Начнется скачивание файла **get-nsi.json**, который содержит запрошенные данные НСИ согласно переданным параметрам (filter: {id: 'Cat01-001-00001', }) (Рисунок 4.27).

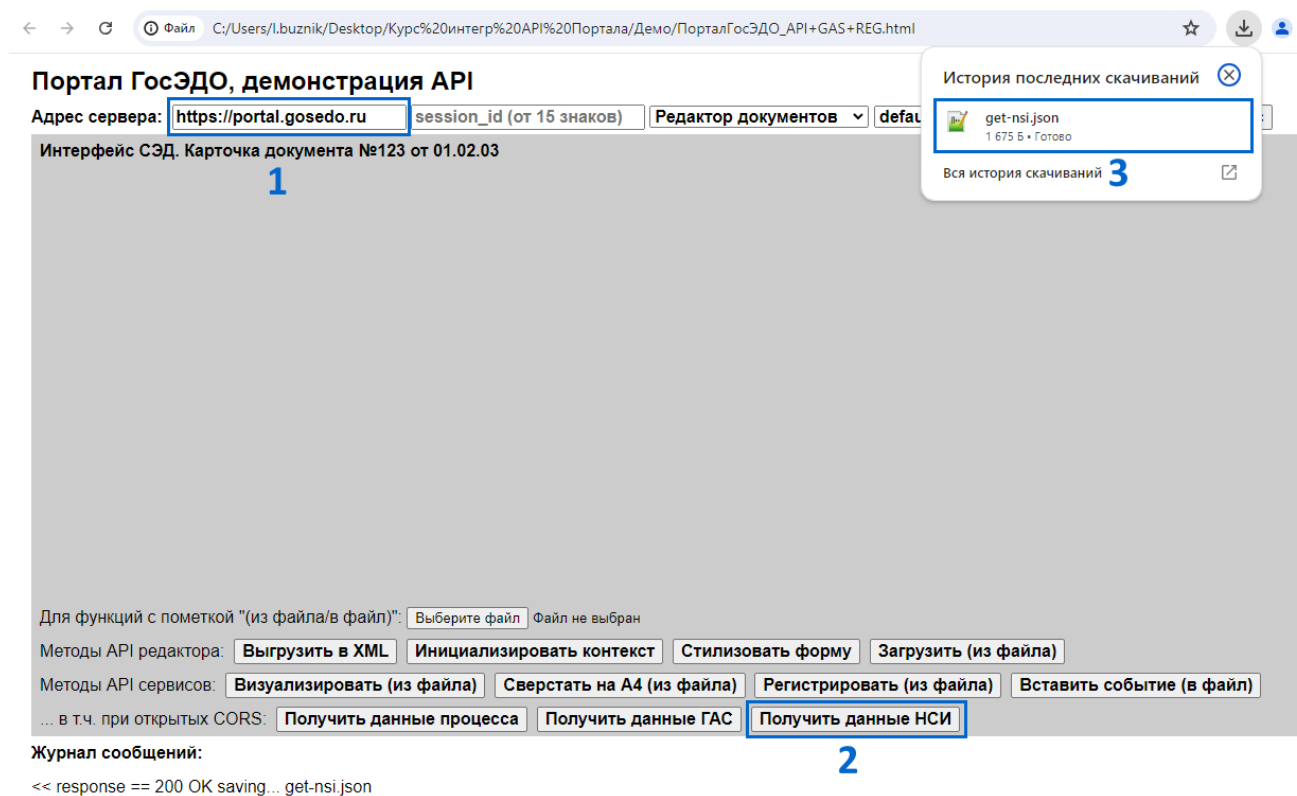


Рисунок 4.27. Порядок действий для демонстрации получения данных НСИ

Откройте на просмотр файл **get-nsi.json** с помощью текстового редактора и убедитесь, что файл содержит запрошенные данные НСИ (Рисунок 4.28).

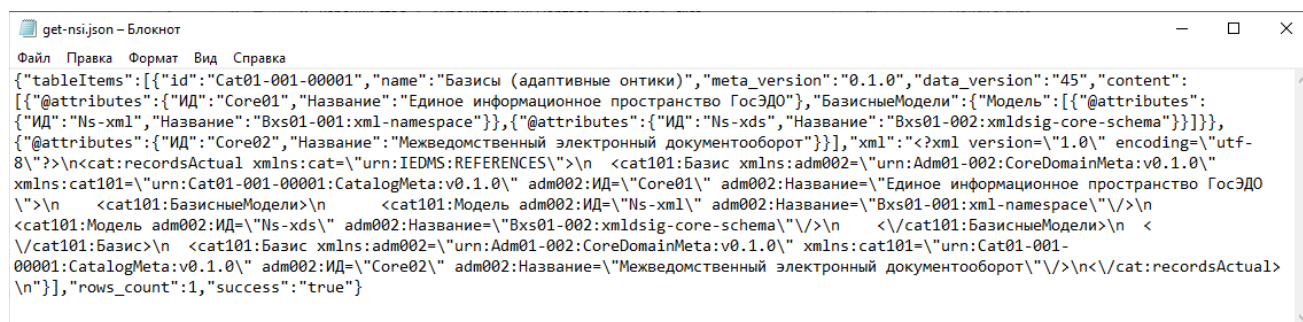


Рисунок 4.28. Пример содержимого файла get-nsi.json

С примером файла, полученного в результате выполнения указанных действий, можно ознакомиться по ссылке: [get-nsi.json](#).

## 5 Интеграция редактора структурированных данных цифрового документа

Редактор структурированных данных «Портала ГосЭДО» является локальным сервисом для формирования и редактирования структурированных данных цифровых документов (XML-файлов).

Для обеспечения возможности формирования файла структурированных данных цифрового документа (файла digital.xml) предлагается интегрировать редактор структурированных данных «Портала ГосЭДО» в СЭД так, чтобы пользователь создавал и изменял структурированные данные в карточке документа СЭД.

Данный способ интеграции позволит:

- формировать файл структурированных данных цифрового документа (XML-файла) в карточке документа СЭД;
- просматривать визуализацию XML-файла, изменять и дополнять его;
- дополнять контейнер документа сформированным XML-файлом в качестве вложения до его отправки в МЭДО;
- выполнять все действия по подготовке и отправке транспортного контейнера документа без необходимости дополнительного перехода в конвертер документов «Портала ГосЭДО».

При реализации такого способа интеграции необходимо реализовать средствами СЭД:

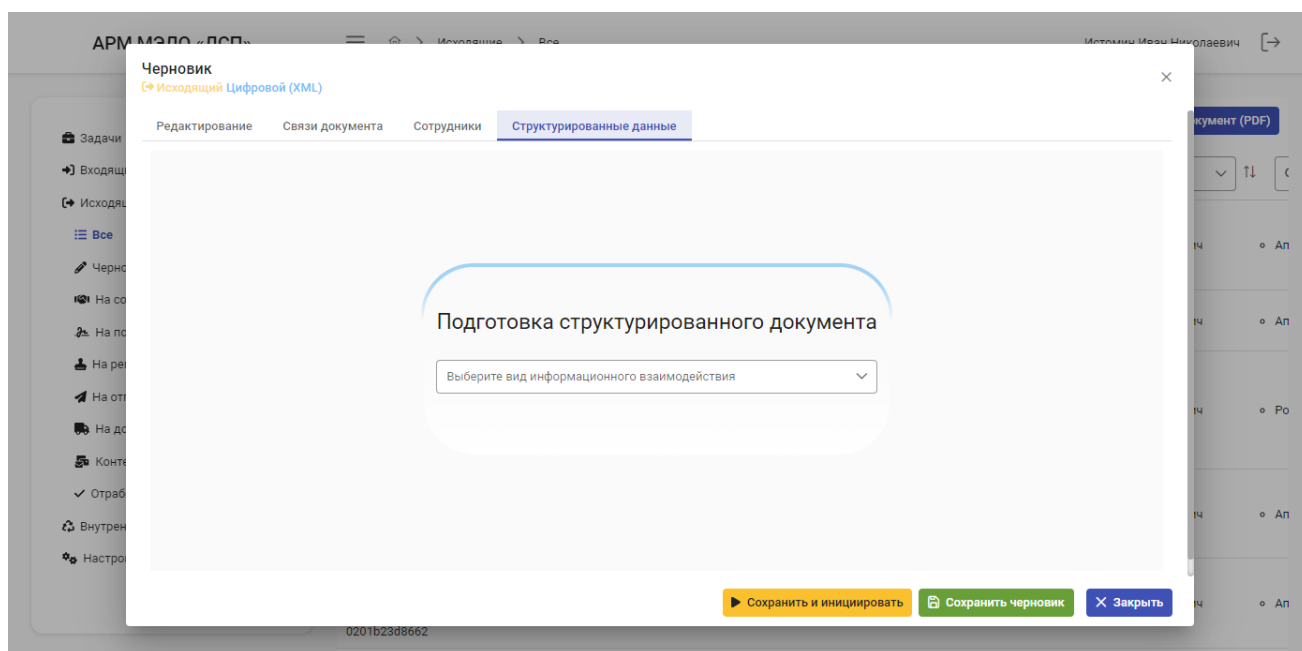
- вызов окна редактора с передачей контекста текущей задачи;
- сохранение результирующего XML-файла в привязке к карточке документа;
- просмотр HTML визуализации XML-файла и печатной формы в виде PDF-файла (при необходимости);
- размещение XML-файла с именем «**digital.xml**» среди вложений транспортного контейнера МЭДО 2.7.1;

- добавление события регистрации и ограничения доступа в XML-файл (при возникновении соответствующих событий);
- автоматическое создание PDF-файла основного документа из структурированных данных файла XML (при необходимости).

Редактор включается в веб-интерфейс СЭД посредством элемента `<iframe>`, с указанием `src="http://server-url/#/constructor/selector"`, где «server-url» - URL-адрес доступа к используемому экземпляру «Портала ГосЭДО». Для выбора нужной темы оформления после адреса редактора дополняется строка `</?theme=[dark/light]>`, где указывается одно из значений: «dark» - для тёмной темы оформления, «light» - для светлой темы оформления.

Пример кода интеграции окна редактора структурированных данных со светлой темой оформления (Рисунок 5.1):

```
<iframe src="http://10.77.142.21:84/#/constructor/selector/?theme=light">
// где
// 10.77.142.21:84 – IP-адрес:порт, на котором доступен модуль «Портал ГосЭДО»
// theme=light – признак светлой темы оформления
```



**Рисунок 5.1. Пример интеграции окна редактирования структурированных данных в веб-интерфейс СЭД со светлой темой оформления**

В зависимости от концепции интерфейса СЭД редактор структурированных данных может быть размещен на отдельной вкладке

карточки документа или открываться в модальном окне, например при нажатии на отдельную кнопку «Ввод цифрового документа (XML)».

В любом случае СЭД должна перехватывать события окна редактирования, предусматривать сигнализацию пользователю ошибки заполнения (при их наличии), а также сохранять результирующий XML-файл в карточке документа СЭД в качестве приложения к основному документу.

## 5.1 Метод передачи данных и команд в форму редактирования

Для передачи данных и команд в окно формы редактирования данных используется интерфейс окна содержимого: атрибут **contentWindow** элемента **<iframe>**. Методом **postMessage** необходимо передать данные в окно **iframe** в виде объекта следующей структуры:

- **data** (объект): {
  - **action** (строка) — команда для выполнения в форме редактирования;
  - **data** ({объект}) — параметры команды, необходимые для её выполнения}

В окно формы редактирования могут быть переданы следующие команды:

- **pressSaveDocumentBtn** – сохранение введенных структурированных данных в файл XML;
- **setIframesContent** – загрузка в редактор данных из файла XML;
- **initData** – инициализация контекста формы редактирования документа;
- **addStyleTag** – добавление CSS-стилей на веб-страницу редактора.

Подробное описание команд и их параметров приведены в курсе далее.

### Пример реализации в JavaScript:

```
<script>
// Получение открытого окна редактора по id="editor" в переменную editorWindow
let editorWindow = document.querySelector('#editor').contentWindow;
// Отправка команды pressSaveDocumentBtn в открытое окно редактора
```

```

    editorWindow.postMessage( { data: { action: 'pressSaveDocumentBtn', data:
{ } } }, '*');
</script>
<!-- фрейм с подключенной формой редактирования данных Портала ГосЭДО-->
<iframe class="editor" id="editor"
src="http://10.77.142.21:84/#/constructor/selector/?theme=light"></iframe>

```

## 5.2 Подключение обработчика событий и получение данных из формы

Для получения событий и введенных данных из открытого окна редактора структурированных данных внутри `<iframe>`, необходимо до открытия `<iframe>` подключить обработчик событий.

### Пример реализации в JavaScript:

```

<script>
// Подключение обработчика событий, выполняется ДО открытия iframe
window.addEventListener('message', handlePostMessage);

// Обработчик событий редактора структурированных данных
async function handlePostMessage(event) {
//здесь указать действия, которые необходимо выполнить с данными, полученными из
окна iframe при наступлении события message. Для получения данных использовать
следующие свойства объекта event:
// event.origin – строка, адрес источника входящего события
// event?.data?.data.xml – строка, текст сформированного в редакторе XML-файла
// event?.data?.data.action – строка, содержит команду действия, произошедшего в
окне редактирования
// event?.data?.data.errors – массив, ошибки сохранения XML, если они есть
}
</script>
    <iframe class="editor" id="editor"
src="http://10.77.142.21:84/#/constructor/selector/?theme=light"></iframe>

```

В функции, выполняемой при наступлении события **message**, следует указать действия, выполняемые с полученными данными. При этом можно обратиться к следующим свойствам объекта `event`:

- **event.origin** (строка) — адрес источника входящего события (пример возвращенного значения: "http://10.77.142.21:84" );
- **event?.data?.data.xml** (строка) – текст сформированного в редакторе XML-файла (пример возвращенного значения:

```
"<?xml version="1.0" encoding="UTF-8"?><doc:Документ xmlns:doc="urn
:Doc01-004-00001:DocumentMeta:v0.1.0"...
```

```
...всё_тело_XML_в_соответствии_с_заполненной_формой_структуриро
ванных_данных...</doc:Документ>";
```

- **event?.data?.data.action** (строка) – возвращает код действия, произошедшего в окне редактирования.

Возможные значения, возвращенные в свойстве «event?.data?.data.action»:

- **"portalReady"** — произошло событие успешной загрузки формы,
- **"XMLSend"** — нажата кнопка «Сохранить данные» формы редактирования (событие происходит при нажатии кнопки пользователем или после передачи в форму команды `pressSaveDocumentBtn`).
- **event?.data?.data.errors** ([массив]) – ошибки при сохранении XML, например список незаполненных обязательных полей (пример возвращенного значения при 2 незаполненных полях формы структурированных данных:

```
["/Документ/ДанныеДокумента/СообщенияДокумента/
ВыдачаПоручения/ПредметыСобытия/Поручение/
ДанныеДляВизуализации/ТекстПоручения", "/Документ/
ДанныеДокумента/СообщенияДокумента/ВыдачаПоручения/
ПредметыСобытия/Поручение/ДанныеДляКоординации/
СрокПоручения"]).
```

### 5.3 Получение данных из формы в формате XML

Для сохранения введенных данных из формы ввода в XML необходимо отправить команду **pressSaveDocumentBtn** в окно формы редактирования.

**Команда:** `pressSaveDocumentBtn`

**Действие:**

Вызывает событие нажатия кнопки сохранения документа.

## Параметры:

нет (без параметров)

## Возвращаемый результат:

Данные возвращаются в обработчик событий внутри объекта `event` свойства `event.data.data.xml` в виде строки с XML-данными, сформированными на основе заполненной формы редактирования (см. раздел 5.2 Подключение обработчика событий и получение данных из формы). Результат возвращается в любом случае, даже если не все данные в форме были заполнены.

## Пример реализации в JavaScript:

```
<script>
// Подключение обработчика событий
window.addEventListener('message', handlePostMessage);
// Обработчик событий редактора структурированных данных
async function handlePostMessage(event) {
// если произошло событие 'XMLSend' (событие нажатия кнопки «Сохранить данные»),
то выполнить нижеследующую обработку
if (event?.data?.data?.action == 'XMLSend') {
    let link = document.createElement('a');
    link.href = 'data:text/plain;charset=utf-8,' +
        encodeURIComponent(event?.data?.data.xml); //обращение к сформированным
Порталом данным XML в свойстве event?.data?.data.xml
    link.download = 'digital.xml';
    link.click();// инициировать загрузку файла
}
function pressSaveBtn() {
// Получение открытого окна редактора по id="editor" в переменную editorWindow
let editorWindow = document.querySelector('#editor').contentWindow;
// Отправка команды pressSaveDocumentBtn в открытое окно редактора:
// Параметры: нет
editorWindow.postMessage( { data: { action: 'pressSaveDocumentBtn', data:
{ } } }, '*');
}
}
</script>
<!--фрейм с подключенной формой редактирования данных Портала ГосЭДО-->
<iframe class="editor" id="editor"
src="http://10.77.142.21:84/#/constructor/selector/?theme=light"></iframe>
<!--кнопка, инициирующая выгрузку файла XML-->
<button onclick="pressSaveBtn()">Выгрузить в XML</button>
```

С выполнением демонстрационного примера по описанной функции можно ознакомиться в разделе 4.3 Демонстрация выгрузки данных из формы в XML.

## 5.4 Загрузка данных в форму из XML

Для загрузки данных из документа XML в форму ввода необходимо использовать команду **setIframesContent**, которая позволяет загрузить данные из структурированных данных XML в форму ввода. Например, команда может быть использована для редактирования ранее сформированных данных, которые хранятся в виде XML в карточке ранее созданного документа-черновика.

**Команда:** `setIframesContent`

**Действие:**

Загружает данные из строки с заполненной XML-структурой документа в форму редактирования. После выполнения команды в форме редактирования автоматически устанавливается форма необходимого вида документа и заполняется данными из XML.

**Параметры:**

**data** (объект): {

- **xml** (строка)};

**Требования к содержимому строки XML:**

Данные документа в структуре XML могут быть заполнены не полностью, если ранее ввод документа не был завершен. Загрузка данных выполняется там, где это возможно, исходя из заполненных элементов и атрибутов.

**Пример реализации в JavaScript:**

```
<script>
//функция, которая выполняется при нажатии кнопки «Загрузить в форму из файла»
function setDigitalXML() {
  //Получаем файл в переменную file
  let file = document.querySelector('#file').files[0];
  var txtXML = '';
  //Создаем объект FileReader для считывания файла
  var reader = new FileReader();
  //При успешном чтении файла выполняем функцию
```



```

reader.onload = function (evt) {
// Присваиваем в переменную txtXML содержимое файла XML
  txtXML = evt.target.result;
//Получение открытого окна редактора цифровых документов по id="editor"
  let editorWindow = document.querySelector('#editor').contentWindow;
  // Отправка команды в открытое окно редактора: setIframesContent
  // Параметры: data.xml = строка содержимого XML
  editorWindow.postMessage( { data: {
    action: 'setIframesContent', data: { xml: txtXML } } }, '*');
}
// считываем файл
reader.readAsText(file);
}
</script>
<!-- фрейм с подключенной формой редактирования данных Портала ГосЭДО-->
<iframe width="1000px" height="700px" id="editor"
src="http://10.77.142.21:84/#/constructor/selector/?theme=light"></iframe>
<!--поле выбора файла XML (digital.xml)-->
<input id="file" type="file">
<!-- кнопка «Загрузить в форму из файла»-->
<button onclick="setDigitalXML()"> Загрузить в форму из файла</button>

```

С выполнением демонстрационного примера по описанной функции можно ознакомиться в разделе 4.4 Демонстрация загрузки данных из файла XML в редактор.

## 5.5 Передача контекста для формы редактирования

При заполнении полей в форме редактирования может быть важен контекст, в рамках которого заполняются структурированные данные: исходные данные текущей карточки СЭД или другие параметры, которые необходимо учитывать при заполнении полей. Например: список сотрудников организации или список приложений к документу. Чтобы передать в форму редактирования данные, необходимо использовать команду **initData**. Команду рекомендуется выполнять до заполнения формы, т. к. некоторые переданные параметры применяются как фильтрация значений, доступных для выбора пользователю.

**Команда:** initData

**Действие:**

Передаёт в форму редактирования контекст ввода документа из СЭД: то есть исходные данные, которые необходимо передать в форму редактирования для использования при заполнении полей.

### Параметры:

**data** (объект): {

- **session\_id** (строка не менее 16 символов, опционально),  
ID сессии (песочницы) в рамках которой организована изоляция данных. Если параметр задан, в форме редактирования будут доступны только процессы по документам с соответствующим ID сессии. Если параметр не задан, то доступны данные процессов по всем документам. Параметр «session\_id» назначается цифровому документу при регистрации во внутреннем модуле координации (см. раздел 6.8 Регистрация цифрового документа во внутреннем модуле координации «Портала ГосЭДО»).
- **selector** (объект, опционально): {  
Объект позволяет передать вид документа, который заполняется в форме. Если данные объекта заполнены, в форме редактирования сразу откроется нужная форма ввода документа, таким образом позволяя пропустить выбор вида информационного взаимодействия, вида события и вида документа.
  - **InformationInteractionsType** (строка),  
Идентификатор вида информационного взаимодействия согласно справочнику НСИ Cat01-002-00003 "Виды информационных взаимодействий". Например, "Ехс01-003-00001" для процесса "Служебная переписка".
  - **InformationInteractionsEvent** (строка),  
Идентификатор вида события согласно справочнику НСИ Cat01-002-00006 "Виды событий" для вида информационного взаимодействия, указанного в свойстве

InformationInteractionsType. Например, "Exc01-004-00003-Event0001" для события "Отправка письма".

- **ElectronicDocumentsType** (строка)

Идентификатор вида документа согласно справочнику НСИ Cat01-002-00002 "Виды электронных документов" или код формы, из перечня видов документов для события, указанного в свойстве InformationInteractionsEvent. Например, "Doc01-004-00001". Также в значении параметра возможно указать строку с наименованием вида документа: например "Универсальный документ".

},

- **organization\_guid** (строка, опционально),

GUID организации согласно ГАС, от имени которой создается документ. Если параметр не задан, то значение берется из настроек «Портала ГосЭДО» (раздел «Настройки» → «МЭДО», поле «GUID-адрес организации»).

- **organization\_title** (строка, опционально),

Наименование организации, от имени которой создается документ. Если параметр не задан, то значение берется из настроек «Портала ГосЭДО» (раздел «Настройки» → «МЭДО», поле «Наименование организации»).

- **attachments** ([массив объектов], опционально): [

Массив объектов с информацией о вложенных файлах. Используется для формирования перечня приложений к документу. Например, переданный массив документов будет доступен для выбора в блоке «ПриложенияДокумента» при заполнении формы по виду документа «Служебное письмо».

Каждый объект массива имеет следующую структуру:

{

- **id** (строка),

Уникальное техническое имя файла в контейнере по шаблону ГосЭДО. Согласно формату ГосЭДО техническое имя файла является строкой из символов [a-z0-9\_-.] длиной от 1 до 60 и является частью ИД приложения, которое формируется по маске: enclosure\_[a-z0-9\_-.]{1,60}\.(gosx|pdf|zip|odt|doc|docx|ods|xls|xlsx|odp|ppt|pptx|tiff|html). Пример: в форму передано значение id='d09de43' и имя файла 'письмо.pdf', в структурированных данных имя файла приложения будет сформировано как «enclosure\_d09de43.pdf».

- **name** (строка),

Пользовательское имя файла, не имеющее ограничений на шаблон.

- **description** (строка)

Назначенное пользователем описание файла.

}

],

- **linked\_docs** ([массив объектов], опционально): [

Массив информации о связанных документах (контейнерах). Используется для фильтрации доступных для выбора процессов в форме редактирования. Например, при формировании доклада на поручение список связанных документов используется «Порталом ГосЭДО», чтобы при выборе значения в блоке «ИнформацияОПроцессе» отобразить только процессы по связанным документам. Если параметр не задан, то пользователю доступны все активные процессы (в рамках заданного идентификатора сессии).

Каждый объект массива имеет следующую структуру:

{

- **guid** (строка),

GUID контейнера связанного документа.

- **title** (строка),

Пользовательское название документа для отображения при выборе.

- **reg\_num** (строка),

Регистрационный номер документа для отображения при выборе.

- **reg\_date** (число)

Дата регистрации документа (unix time) для отображения при выборе.

}

],

- **users** ([массив], опционально): [

Массив сотрудников организации (пользователей). Используется для выбора в соответствующих полях формы редактирования, где необходимо указать сотрудника организации (например при выборе сотрудника в форме цифрового документа «Акт об ознакомлении с документом ДСП»).

Каждый объект массива имеет следующую структуру:

{

- **id** (строка),

Уникальный идентификатор сотрудника организации.

- **fio** (строка)

Фамилия, имя, отчество сотрудника организации.

}

]

}

**Пример реализации в JavaScript:**

```

<script>
//функция, которая выполняется при нажатии кнопки «Инициализировать контекст»
function initContext() {
  jsonData = {
    //идентификатор сессии
    session_id: document.querySelector('#sessionId').value,
    //данные вида информационного взаимодействия, вида события и документа
    selector: {
      InformationInteractionsType: 'Exc01-004-00003', //вид информационного
взаимодействия
      InformationInteractionsEvent: 'Exc01-004-00003-Event0002', //вид события
      ElectronicDocumentsType: 'Универсальный документ' //вид документа
    },
    //GUID организации согласно ГАС
    organization_guid: 'b23a5a35-dde3-48ef-aa07-621eb56417db',
    //Наименование организации согласно ГАС
    organization_title: 'Минфин России',
    //Приложения к документу (массив объектов), в данном примере 2 приложения:
проект.docx и схема.pdf
    attachments: [
      {id: 'd0725f36', name: 'проект.docx', description: 'проект договора'},
      {id: 'f09daf08', name: 'схема.pdf', description: 'схема проезда'}
    ],
    //список связанных документов (массив объектов), в данном
примере 2 связанных документа
    linked_docs: [
      {guid: '3367493a-097a-468e-8ff3-7366858303e2',
title: 'Служебное письмо №3', reg_num: '3', reg_date: 1691096400},
      {guid: '4367493a-097a-468e-8ff3-7366858303e2',
title: 'Служебное письмо №4', reg_num: '4', reg_date: 1691200000}
    ],
    //список сотрудников организации, которые будут доступны для выбора в полях
формы, где нужно выбрать значение из справочника сотрудников
    users: [
      {id: 1, fio: 'Иванов Иван Иванович'},
      {id: 2, fio: 'Петров Петр Петрович'},
      {id: 3, fio: 'Сидоров Сергей Сергеевич'}
    ]
  };
  // Получение в переменную editorWindow открытого окна редактора цифровых
документов по id=editor
  let editorWindow = document.querySelector('#editor').contentWindow;
    // Передача данных в форму редактирования методом POST
    // Команда initData
    // Параметры - объект jsonData, заполненный необходимыми данными
ВЫШЕ
  editorWindow.postMessage( { data: {
    action: 'initData', data: jsonData } }, '*');
}
</script>
  <!-- фрейм с подключенной формой редактирования данных Портала ГосЭДО-->

```

```

<iframe width="1000px" height="700px" id="editor"
src="http://10.77.142.21:84/#/constructor/selector/?theme=light"></iframe>
<!-- поле для ввода идентификатора сессии -->
<input id="sessionId" type="text" placeholder="session_id (от 15 знаков)"/>
<!-- кнопка, по нажатию на которую произойдет загрузка контекста в форму --
>
<button onclick="initContext()"> Инициализировать контекст </button>

```

С выполнением демонстрационного примера по описанной функции можно ознакомиться в разделе 4.2 Пример заполнения структурированных данных и передача контекста в редактор.

## 5.6 Применение CSS-стилей к форме

Для применения CSS-стилей к форме редактирования данных следует использовать команду **addStyleTag**.

**Команда:** addStyleTag

**Действие:**

Добавляет описание CSS-стилей в виде блока

`<style>переданная_строка_CSS-стиля</style>` внутри блока `<head>` формы редактирования.

**Параметры:**

**data** (структура): {

- **text** (строка) };

В параметре `data.text` необходимо передать строку стилей CSS с указанием классов, используемых в форме ввода. Например, переданная в параметр `data.text` строка `'.container{width: 500pt !important;}'` выполнит изменение ширины формы ввода.

**Пример реализации в JavaScript:**

```

<script>
    // Функция, которая выполняется при нажатии кнопки «Стилизовать форму»
    function addStyleTag() {
        // Присвоение в переменную txtStyle строки CSS-стиля, например изменение
        ширины формы - '.container {width: 500pt !important;}'
        txtStyle = '.container {width: 500pt !important;}';
        // Получение открытого окна редактора цифровых документов по id=editor
        let editorWindow = document.querySelector('#editor').contentWindow;

```

```
// Отправка команды в открытое окно редактора: addStyleTag
// Параметры: data.text = строка стиля
// В параметр data.text передается строка CSS-стиля из переменной txtStyle
editorWindow.postMessage( { data: {
  action: 'addStyleTag', data: { text: txtStyle } } }, '*');
}
</script>
<!--далее фрейм с подключенной формой редактирования данных Портала ГосЭДО-->
<iframe width="1000px" height="700px" id="editor"
src="http://10.77.142.21:84/#/constructor/selector/?theme=light"></iframe>
<!-- кнопка-->
<button onClick="addStyleTag()"> Стилизовать форму </button>
```

С выполнением демонстрационного примера по описанной функции можно ознакомиться в разделе 4.5 Демонстрация передачи стилей в редактор.



## 6 Интеграция функций обработки цифровых документов

Инструменты обработки цифровых документов «Портала ГосЭДО» доступны для интеграции в виде веб-сервисов через REST API. Сервисы могут принимать данные для обработки, а также возвращать результаты обработки.

В разделах далее вы можете ознакомиться со следующей информацией:

- метод передачи данных и команд;
- HTML-визуализация документа из файла XML;
- создание документа PDF из XML;
- создание документа PDF из HTML-визуализации;
- конвертация паспорта документа в цифровой документ «Служебное письмо»;
- конвертация контейнера МЭДО в контейнер цифрового документа «Служебное письмо»;
- вставка событий регистрации и ограничения доступа в XML;
- регистрация цифрового документа во внутреннем модуле координации «Портала ГосЭДО»;
- получение информации о состоянии цифрового процесса.

### 6.1 Метод передачи данных и команд

Основные рабочие сервисы доступны посредством POST по URL «<http://server-url/api/service>», где «[server-url](http://server-url/api/service)» - URL-адрес доступа к используемому экземпляру «Портала ГосЭДО». Команда передается в параметре «**action**». Некоторые методы требуют установки ID сессии (песочницы).

#### Пример вызова в консоли Linux:

```
curl 'http://server-url:84/api/service' \  
-H 'Content-Type: application/json' \  
-d '{"action": "getProcessInfo", "process_guid": "445a169e-70f2-4ca0-b6be-5de7ac195e82"}'
```

#### Пример вызова в JavaScript:

```

const formData = new FormData();
formData.append('action', 'getProcessInfo');
formData.append('process_guid', '445a169e-70f2-4ca0-b6be-5de7ac195e82');

// Асинхронный вызов REST API
const resp = await fetch(' http://10.77.142.21:84/api/service',
{ method: 'POST', body: formData })

```

## 6.2 HTML-визуализация документа из файла XML

Через запрос к API «Портала ГосЭДО» можно выполнить визуализацию файла цифрового документа XML в виде HTML-файла с помощью команды `convertXmlToHtml`.

**Команда:** `convertXmlToHtml`

**Действие:**

Визуализирует документ по шаблону НСИ из XML в файл HTML.

**Параметры:**

- `xml` (файл) – цифровой документ в формате XML.

**Возвращаемый результат:**

Возвращает файл HTML, полученный в результате визуализации.

**Пример реализации в JavaScript:**

```

<!-- REST API Портала ГосЭДО: сервисы для цифровых документов -->
<!-- Для вызова используется REST API http(s)://server-url/#/SERVICE -->
<script>
  async function fetchAdapterService(myAction, myKey, myValue, saveAsFile,
mySession = 'NONE') {
    // Пример адреса для обращения к сервисам, если Портал ГосЭДО доступен по IP
10.77.142.21:84
    RESTAPIURL = 'http://10.77.142.21:84/api/service';

    const formData = new FormData();
    formData.append('action', myAction);
    formData.append(myKey, myValue);

    if (mySession !== 'NONE')
      formData.append('session_id', mySession);

    var object = {};
    formData.forEach(function(value, key){ object[key] = value; });

    const response = await fetch(RESTAPIURL, {
      method: 'POST',

```

```

    body: formData
  }).catch((error) => {});

  if (response.status === 200) {
    //Присвоение в переменную file ответа
    let file = await response.blob();
    let link = document.createElement("a");
    link.href = window.URL.createObjectURL(file);
    link.download = saveAsFile;
    link.click();
  }
}
</script>

<!-- REST API Портала ГосЭДО: визуализация документа из файла XML -->
<script>
  // функция, которая выполняется при нажатии кнопки «Визуализировать из файла в
  HTML»
  async function visualizeDocument() {
    //Получение в переменную myValue файла XML, выбранного в поле file
    myValue = document.querySelector('#file').files[0];
    //передача параметров в функцию fetchAdapterService:
    //Команда: convertXmlToHtml
    //Параметр: xml
    //Значение параметра: myValue (файл)
    fetchAdapterService('convertXmlToHtml', 'xml', myValue, 'document.html');
  }
</script>

<!-- поле выбора файла-->
<input id="file" type="file">
<!-- кнопка для запуска HTML-визуализации выбранного XML-файла в поле file-->
<button onclick="visualizeDocument()"> Визуализировать из файла в HTML</button>

```

С выполнением демонстрационного примера по описанной функции можно ознакомиться в разделе 4.6 Демонстрация получения HTML-визуализации из файла XML.

### 6.3 Создание документа PDF из XML

Через запрос к API «Портала ГосЭДО» можно выполнить визуализацию файла цифрового документа XML в виде PDF-файла с помощью команды **convertXmlToPDF**.

**Команда:** `convertXmlToPDF`

**Действие:**

Создает печатный документ по шаблону НСИ из XML в файл PDF.

### Параметры:

- **xml** (файл) – цифровой документ в формате XML.

### Возвращаемый результат:

Возвращает файл PDF, полученный в результате подготовки к печати.

### Пример реализации в JavaScript:

```
<!--REST API Портала ГосЭДО: сервисы для цифровых документов -->
<!--Для вызова используется REST API http(s)://server-url/#/SERVICE-->
<script>
  async function fetchAdapterService(myAction, myKey, myValue, saveAsFile,
mySession = 'NONE') {
    // Пример адреса для обращения к сервисам, если Портал ГосЭДО доступен по IP
10.77.142.21:84
    RESTAPIURL = 'http://10.77.142.21:84/api/service';

    const formData = new FormData();
    formData.append('action', myAction);
    formData.append(myKey, myValue);

    if (mySession !== 'NONE')
      formData.append('session_id', mySession);

    var object = {};
    formData.forEach(function(value, key){ object[key] = value; });

    const response = await fetch(RESTAPIURL, {
      method: 'POST',
      body: formData
    }).catch((error) => {});

    if (response.status === 200) {
      //Присвоение в переменную file ответа
      let file = await response.blob();
      let link = document.createElement("a");
      link.href = window.URL.createObjectURL(file);
      link.download = saveAsFile;
      link.click();
    }
  }
</script>

<!--REST API Портала ГосЭДО: верстка в PDF из файла XML -->
<script>
// функция, которая выполняется при нажатии кнопки «Сверстать в PDF из файла
XML»
  async function preprintDocument() {
```

```
//Получение в переменную myValue файла XML, выбранного в поле file
myValue = document.querySelector('#file').files[0];
//передача параметров в функцию fetchAdapterService:
//Команда: convertXmlToPDF
//Параметр: xml
//Значение параметра: myValue (файл)
fetchAdapterService('convertXmlToPDF', 'xml', myValue, 'document.pdf');
}
</script>

<!--поле выбора файла digital.xml-->
<input id="file" type="file">
<!-- кнопка для запуска PDF-визуализации выбранного XML-файла в поле file-->
<button onclick="preprintDocument()"> Сверстать в PDF из файла XML </button>
```

С выполнением демонстрационного примера по описанной функции можно ознакомиться в разделе 4.7 Демонстрация создания печатной формы PDF из файла XML.

## 6.4 Создание документа PDF из HTML-визуализации

Через запрос к API «Портала ГосЭДО» можно выполнить формирование PDF-файла из файла HTML-визуализации цифрового документа (сформированный ранее командой convertXmlToHtml) с помощью команды convertHTMLToPDF.

**Команда:** convertHtmlToPdf

**Действие:**

Создает печатный документ по шаблону НСИ из HTML в файл PDF.

**Параметры:**

- **html** (файл) – визуализация цифрового документа в файле HTML (например, полученный после выполнения команды convertXmlToHtml см. раздел 6.2 HTML-визуализация документа из файла XML).

**Возвращаемый результат:**

Возвращает файл PDF, полученный в результате подготовки к печати.

**Пример реализации в JavaScript:**

```
<!-- REST API Портала ГосЭДО: сервисы для цифровых документов -->
```

```

<!-- Для вызова используется REST API http(s)://server-url/#/SERVICE -->
<script>
  async function fetchAdapterService(myAction, myKey, myValue, saveAsFile,
mySession = 'NONE') {
    // Пример адреса для обращения к сервисам, если Портал ГосЭДО доступен по IP
10.77.142.21:84
    RESTAPIURL = 'http://10.77.142.21:84/api/service';

    const formData = new FormData();
    formData.append('action', myAction);
    formData.append(myKey, myValue);

    if (mySession != 'NONE')
      formData.append('session_id', mySession);

    var object = {};
    formData.forEach(function(value, key){ object[key] = value; });

    const response = await fetch(RESTAPIURL, {
      method: 'POST',
      body: formData
    }).catch((error) => {});

    if (response.status == 200) {
      //Присвоение в переменную file ответа
      let file = await response.blob();
      let link = document.createElement("a");
      link.href = window.URL.createObjectURL(file);
      link.download = saveAsFile;
      link.click();
    }
  }
</script>

<!-- REST API Портала ГосЭДО: верстка в PDF из файла HTML -->
<script>
  // функция, которая выполняется при нажатии кнопки «Сверстать в PDF из
файла HTML»
  async function preprintHTMLDocument() {
    //Получение в переменную myValue файла HTML, выбранного в поле file
myValue = document.querySelector('#file').files[0];
    //передача параметров в функцию fetchAdapterService:
    //Команда: convertHtmlToPdf
    //Параметр: html
    //Значение параметра: myValue (файл)
    //Сохранить результат как document.pdf
    fetchAdapterService('convertHtmlToPdf', 'html', myValue, 'document.pdf');
  }
</script>

<!-- поле выбора файла -->

```

```
<input id="file" type="file">
<!-- кнопка для запуска PDF-визуализации выбранного HTML-файла в поле file-->
<button onclick="preprintHTMLDocument()"> Сверстать в PDF из файла HTML </button>
```

## 6.5 Конвертация паспорта документа в цифровой документ «Служебное письмо»

Через запрос к API «Портала ГосЭДО» можно выполнить конвертацию паспорта документа в электронном виде формата МЭДО 2.7.1 (passport.xml) в цифровой документ по виду документа «Служебное письмо» (digital.xml) с помощью команды **convertMedoXmlToDigitalXML**.

**Команда:** convertMedoXmlToDigitalXML

**Действие:**

Конвертирует паспорт документа в электронном виде формата МЭДО 2.7.1 (passport.xml) в цифровой документ по виду документа «Служебное письмо» (вид документа «Дос01-004-00001» согласно справочнику НСИ Cat01-002-00002 «Виды электронных документов»). При этом в результирующем файле digital.xml файл основного документа (PDF) указывается как приложение к цифровому документу.

**Параметры:**

- **xml** (файл) – паспорт документа МЭДО в формате XML (passport.xml).

**Возвращаемый результат:**

Возвращает файл XML (digital.xml), полученный в результате конвертации.

**Пример реализации в JavaScript:**

```
<!-- REST API Портала ГосЭДО: сервисы для цифровых документов -->
<!-- Для вызова используется REST API http(s)://server-url/#/SERVICE -->
<script>
  async function fetchAdapterService(myAction, myKey, myValue, saveAsFile,
  mySession = 'NONE') {
    // Пример адреса для обращения к сервисам, если Портал ГосЭДО доступен по IP
    10.77.142.21:84
    RESTAPIURL = 'http://10.77.142.21:84/api/service';

    const formData = new FormData();
    formData.append('action', myAction);
    formData.append(myKey, myValue);
```

```

if (mySession !== 'NONE')
  formData.append('session_id', mySession);

var object = {};
formData.forEach(function(value, key){ object[key] = value; });

const response = await fetch(RESTAPIURL, {
  method: 'POST',
  body: formData
}).catch((error) => {});

if (response.status === 200) {
  //Присвоение в переменную file ответа
  let file = await response.blob();
  let link = document.createElement("a");
  link.href = window.URL.createObjectURL(file);
  link.download = saveAsFile;
  link.click();
}
}
</script>

<!-- REST API Портала ГосЭДО: конвертация passport.xml в "Служебное письмо"
digital.xml -->
<script>
  // функция, которая выполняется при нажатии кнопки «Конвертировать паспорт в
"Служебное письмо»
  async function convertPassport() {
    //Получение в переменную myValue файла XML, выбранного в поле file
    myValue = document.querySelector('#file').files[0];
    //передача параметров в функцию fetchAdapterService:
    //Команда: convertMedoXmlToDigitalXML
    //Параметр: xml
    //Значение параметра: myValue (файл)
    //сохранить результат как digital.xml
    fetchAdapterService('convertMedoXmlToDigitalXML', 'xml', myValue,
'digital.xml');
  }
</script>

<!--поле выбора файла, нужно выбрать файл passport.xml из транспортногo контейнера
МЭДО-->
<input id="file" type="file">
<!-- кнопка для запуска конвертации паспорта (passport.xml), выбранного в поле
file-->
<button onClick="convertPassport()"> Конвертировать паспорт в "Служебное
письмо"</button>

```



## 6.6 Конвертация контейнера МЭДО в контейнер цифрового документа «Служебное письмо»

Через запрос к API «Портала ГосЭДО» можно выполнить конвертацию контейнера документа в электронном виде формата МЭДО 2.7.1 в контейнер цифрового документа «Служебное письмо» с помощью команды **convertMedoContainerToDigitalContainer**.

**Команда:** `convertMedoContainerToDigitalContainer`

**Действие:**

Конвертирует контейнер документа в электронном виде формата МЭДО 2.7.1 (`document.edc.zip`) в контейнер цифрового документа (`document.gosx`) вида «Служебное письмо» (вид документа «Doc01-004-00001»). При этом файл основного документа (PDF) помещается в контейнер как приложение к цифровому документу.

**Параметры:**

- **zip** (файл) – контейнер документа МЭДО в формате ZIP (`document.edc.zip`).

**Возвращаемый результат:**

Возвращает контейнер цифрового документа (`document.gosx`), полученный в результате конвертации. Формат цифрового документа «GOSX» является ZIP-архивом, включающим файл структурированных данных документа (XML), файлы приложений к цифровому документу и файлы подписей.

**Пример реализации в JavaScript:**

```
<!-- REST API Портала ГосЭДО: сервисы для цифровых документов -->
<!-- Для вызова используется REST API http(s)://server-url/#/SERVICE -->
<script>
  async function fetchAdapterService(myAction, myKey, myValue, saveAsFile,
mySession = 'NONE') {
    // Пример адреса для обращения к сервисам, если Портал ГосЭДО доступен по IP
10.77.142.21:84
    RESTAPIURL = 'http://10.77.142.21:84/api/service';

    const formData = new FormData();
    formData.append('action', myAction);
```

```

formData.append(myKey, myValue);

if (mySession !== 'NONE')
  formData.append('session_id', mySession);

var object = {};
formData.forEach(function(value, key){ object[key] = value; });

const response = await fetch(RESTAPIURL, {
  method: 'POST',
  body: formData
}).catch((error) => {});

if (response.status === 200) {
  //Присвоение в переменную file ответа
  let file = await response.blob();
  let link = document.createElement("a");
  link.href = window.URL.createObjectURL(file);
  link.download = saveAsFile;
  link.click();
}
}
</script>

<!-- REST API Портала ГосЭДО: конвертация контейнера МЭДО в контейнер цифрового
документа "Служебное письмо" -->
<script>
// функция, которая выполняется при нажатии кнопки «Конвертировать контейнер»
async function convertContainer() {
  //Получение в переменную myValue файла document.edc.zip, выбранного в поле
file
  myValue = document.querySelector('#file').files[0];
  //передача параметров в функцию fetchAdapterService:
  //Команда: convertMedoContainerToDigitalContainer
  //Параметр: zip
  //Значение параметра: myValue (файл)
  //сохранить результат как document.gosx
  fetchAdapterService('convertMedoContainerToDigitalContainer', 'zip', myValue,
'document.gosx');
}
</script>

<!-- поле выбора файла, нужно выбрать файл document.edc.zip из транспортного
контейнера МЭДО-->
<input id="file" type="file">
<!-- кнопка для запуска конвертации контейнера, выбранного в поле file-->
<button onClick="convertContainer()"> Конвертировать контейнер</button>

```

## 6.7 Вставка событий регистрации и ограничения доступа в XML

Через запрос к API «Портала ГосЭДО» можно добавить в цифровой документ XML события делопроизводства, связанные с официальной регистрацией или ограничением доступа по документам ДСП с помощью команды **insertEventToDigitalXml**.

**Команда:** insertEventToDigitalXml

**Действие:**

Вставляет в цифровой документ XML события делопроизводства, связанные с официальной регистрацией цифровых документов или ограничением доступа по документам ДСП.

**Параметры:**

- **xml** (файл) – цифровой документ в формате XML;
- **params** (строка JSON):
  - **evName** – название события в одинарных кавычках (например: 'НаправлениеИсходящего', 'ОграничениеДоступа').

Возможные значения:

- 'НаправлениеИсходящего' (добавляет в документ XML блок структурированных данных с информацией о регистрации исходящего документа, событие следует выполнять после регистрации исходящего документа в СЭД перед его отправкой);
- 'РегистрацияВходящего' (если передано это событие, функция возвращает XML-файл уведомления о регистрации входящего документа, событие может быть выполнено после регистрации входящего цифрового документа в СЭД);

- 'ОграничениеДоступа' (установка ограничительной пометки ДСП, событие должно быть добавлено в XML в случае, если на документ наложена пометка ДСП);
  - 'СнятиеПометкиСЭкземпляра' (снятие ограничительной пометки ДСП, событие должно быть выполнено при снятии с цифрового документа пометки ДСП);
  - 'СозданиеКопииЭкземпляра' (создание копии экземпляра документа ДСП, событие должно быть добавлено в XML при создании копии цифрового документа с пометкой ДСП).
- **evUuid** – GUID, сгенерированный для этого события (GUID события);
  - **evDate** – дата и время события, например '2024-06-05T12:02:49+03:00';
  - **evAgentUuid** – GUID организации-отправителя согласно ГАС;
  - **evAgentName** – наименование организации-отправителя (наименование участника ГАС).

Далее необходимо в составе параметра «params» передать дополнительные параметры, которые зависят от события. Фиксированные значения также должны быть заключены в одинарные кавычки, так как они передаются в метод select. Разрешается указывать выражения xpath, в таком случае используется синтаксис xpath и одинарные кавычки не нужны.

Дополнительные параметры в зависимости от вида события (evName):

- для события 'НаправлениеИсходящего':
  - **evRegDate** – дата регистрации (например '2023-06-05');
  - **evRegNumber** – номер регистрации (например '65');

- **evDocKindId** — код вида документа согласно справочнику НСИ Cat02-002-00001 «Виды документов МЭДО» (например 'DK00000000');
- **evDocKindName** — наименование вида документа согласно справочнику НСИ Cat02-002-00001 «Виды документов МЭДО» (например 'Другие виды документов');
- **evDocPlaceId** — код места составления документа согласно справочнику НСИ Cat02-002-00002 «Места составления документов» (например 'DP00000099');
- **evDocPlaceName** — наименование места составления документа согласно справочнику НСИ Cat02-002-00002 «Места составления документов» (например 'Иные территории');
- **evDocClassId** — код грифа ограничения доступа согласно справочнику НСИ Cat02-002-00003 «Гриффы ограничения доступа» (например 'DC00000000');
- **evDocClassName** - наименование грифа ограничения доступа согласно справочнику НСИ Cat02-002-00003 «Гриффы ограничения доступа» (например 'Обычная информация');
- для события 'РегистрацияВходящего':
  - **evRegDate** – дата регистрации;
  - **evRegNumber** – номер регистрации;
- для события 'ОграничениеДоступа':
  - **evUserName** – ФИО делопроизводителя (пользователя), установившего ограничительную пометку ДСП;
  - **evUserId** – идентификатор делопроизводителя (пользователя), установившего ограничительную пометку ДСП;
  - **evInstId** – уникальный номер экземпляра документа ДСП по формату ГосЭДО;

- для события 'СнятиеПометкиСЭкземпляра':
  - **evEnclUuid** – GUID цифрового документа «Акт о раскрытии документа ДСП» (ДокументУУИД);
  - **evEnclHash** – хэш цифрового документа «Акт о раскрытии документа ДСП»;
- для события 'СозданиеКопииЭкземпляра':
  - **evCopyId** – уникальный номер копии, добавляемый к номеру экземпляра, согласно формату ГосЭДО.

### Возвращаемый результат:

Для событий: 'НаправлениеИсходящего', 'ОграничениеДоступа', 'СнятиеПометкиСЭкземпляра', 'СозданиеКопииЭкземпляра' - возвращает входной файл цифрового документа в формате XML (digital.xml), дополненный структурированными данными о переданном событии.

Для события 'РегистрацияВходящего' - возвращает файл XML уведомления о регистрации.

### Пример реализации в JavaScript:

```

<!------- REST API Портала ГосЭДО: сервисы для цифровых документов -->
<!------- Для вызова используется REST API http(s)://server-url/#/SERVICE
-->
<script>
  async function fetchAdapterService(endpoint, myAction, myParams, saveAsFile =
'NONE') {
    let address = document.querySelector('#address');
    RESTAPIURL = 'http://10.77.142.21:84/' + endpoint;

    myParams['action'] = myAction;

    if (myParams['xml']) {
      var formData = new FormData();
      Object.entries(myParams).forEach(function([key, value]) {
        formData.append(key, value);
      });
      var REQUEST = {
        method: 'POST',
        body: formData,
      }
    } else {
      var REQUEST = {

```

```

        method: 'POST',
        body: JSON.stringify(myParams),
        headers: {'Content-Type': 'application/json'},
    }
}

        const response = await fetch(RESTAPIURL, REQUEST)
    .catch( (error) => { });
const fileName = ((saveAsFile == 'NONE') ? 'response.json' : saveAsFile);

if (response.status == 200) {
    let file = await response.blob();
    let link = document.createElement("a");
    link.href = window.URL.createObjectURL(file);
    link.download = fileName;
    link.click();
}
}
</script>

<!-- REST API Портала ГосЭДО: вставка события регистрации -->
<script>
// функция, которая выполняется при нажатии кнопки «Вставить событие (в файл)»
    async function insertDocument() {
myParams = {
    xml: document.querySelector('#file').files[0],
    params: JSON.stringify({
        evName: 'НаправлениеИсходящего',
        evUuid: '19d4e3ea-7f86-4d15-9319-8cb09669d77e',
        evDate: '2024-06-05T12:02:49+03:00',
        evAgentUuid: '1d04ca3e-df1a-0cb4-c325-6ef4003ca2ab',
        evAgentName: 'Минцифры России',
        evRegDate: '2024-06-05',
        evRegNumber: '12',
        evDocKindId: 'DK00000000',
        evDocKindName: 'Другие виды документов',
        evDocPlaceId: 'DP00000099',
        evDocPlaceName: 'Иные территории',
        evDocClassId: 'DC00000000',
        evDocClassName: 'Обычная информация', }),
    }
    fetchAdapterService('/api/service', 'insertEventToDigitalXml', myParams,
'registered.xml');
    }
</script>
    <!-- поле выбора файла digital.xml -->
    <input id="file" type="file">
    <!-- кнопка для инициации вставки события регистрации в файл digital.xml -->
    <button class="header2" onclick="insertDocument()"> Вставить событие (в
файл) </button>

```

С выполнением демонстрационного примера по описанной функции можно ознакомиться в разделе 4.10 Демонстрация вставки события регистрации в файл XML.

## **6.8 Регистрация цифрового документа во внутреннем модуле координации «Портала ГосЭДО»**

Через запрос к API «Портала ГосЭДО» можно выполнить регистрацию цифрового документа на «Портале ГосЭДО» с помощью команды **saveDigitalDocument**. Это позволит занести в базу данных «Портала ГосЭДО» сведения о новом цифровом документе в организации.

**Команда:** saveDigitalDocument.

**Действие:** регистрирует цифровой документ во внутреннем модуле координации «Портала ГосЭДО».

**Параметры:**

- **xml** (файл) – цифровой документ в формате XML (digital.xml);
- **session\_id** (строка от 16 символов, опционально) – ID сессии (песочницы), в рамках которой организована изоляция данных. ID сессии фиксируется за цифровым документом во внутреннем модуле координации «Портала ГосЭДО». ID сессии позволяет изолировать работу пользователей с документами только внутри заданного ID. Если другой пользователь начнет работу с другим ID сессии, ему будут видны только документы и процессы с его ID сессии. Это позволяет изолировать данные при использовании одного экземпляра «Портала ГоЭДО» для большого количества организаций, где требуется обеспечить разграничение видимости документов для каждой организации или в других случаях, где требуется изоляция данных.

**Возвращаемый результат:**

Возвращает данные в формате json, с результатом регистрации в параметре **success** (булево, true/false).

**Пример реализации в JavaScript:**



```

<script>
<!-- REST API Портала ГосЭДО: сервисы для цифровых документов -->
<!-- Для вызова используется REST API http(s)://server-url/#/SERVICE -->
async function fetchAdapterService(myAction, myKey, myValue, saveAsFile,
mySession = 'NONE') {
  // Пример адреса для обращения к сервисам, если Портал ГосЭДО доступен по IP
10.77.142.21:84
  RESTAPIURL = 'http://10.77.142.21:84/api/service';

  const formData = new FormData();
  formData.append('action', myAction);
  formData.append(myKey, myValue);

  if (mySession != 'NONE')
    formData.append('session_id', mySession);

  var object = {};
  formData.forEach(function(value, key){ object[key] = value; });

  const response = await fetch(RESTAPIURL, {
    method: 'POST',
    body: formData
  }).catch((error) => {});

  if (response.status == 200) {
    //Присвоение в переменную file ответа
    let file = await response.blob();
    let link = document.createElement("a");
    link.href = window.URL.createObjectURL(file);
    link.download = saveAsFile;
    link.click();
  }
}
</script>

```

```

<!--REST API Портала ГосЭДО: регистрация цифрового документа из файла XML во
внутреннем модуле Портала ГосЭДО-->
<script>
  // функция, которая выполняется при нажатии кнопки «Зарегистрировать документ
XML на Портале ГосЭДО»
  async function processDocument() {
    myValue = document.querySelector('#file').files[0];
    //передача параметров в функцию fetchAdapterService:
    //Команда: saveDigitalDocument
    //Параметр xml: xml
    //Значение параметра: myValue (файл)
    //Сохранить результат как response.json
    //Параметр: session_id
    //Значение параметра session_id: идентификатор сессии из поля ввода #sessionId

```

```

    fetchAdapterService('saveDigitalDocument', 'xml', myValue, 'response.json',
document.querySelector('#sessionId').value);
}
</script>

<!-- поле для ввода идентификатора сессии -->
<input id="sessionId" type="text" placeholder="session_id (от 15 знаков)"/>
<!--поле выбора файла, нужно выбрать digital.xml-->
<input id="file" type="file">
<!-- кнопка для запуска процесса регистрации цифрового документа на Портале
ГосЭДО-->
<button onclick="processDocument()"> Зарегистрировать документ XML на Портале
ГосЭДО </button>

```

С выполнением демонстрационного примера по описанной функции можно ознакомиться в разделе 4.8 Демонстрация регистрации цифрового документа XML во внутреннем модуле координации.

## 6.9 Получение информации о состоянии цифрового процесса

Через запрос к API «Портала ГосЭДО» можно запросить актуальное состояние координации по цифровому процессу с помощью команды **getProcessInfo**.

### Команда:

getProcessInfo

### Действие:

Возвращает актуальное состояние координации по процессу.

### Параметры:

- **process\_guid** (строка) – уникальный идентификатор цифрового процесса (ПроцессУУИД).
- **session\_id** (строка от 16 символов, опционально) – ID сессии (песочницы) в рамках которой организована изоляция данных.

### Возвращаемый результат:

Возвращает данные в формате json, содержащие набор массивов со сведениями о процессе:

- **info** ([массив]) - информация о процессе;

- **events** ([массив]) - события в ходе процесса;
- **things** ([массив]) - предметы контроля по процессу;
- **things\_data** ([массив]) - реквизиты предметов контроля;
- **things\_statuses** ([массив]) - состояния предметов контроля.

### Пример реализации в JavaScript:

```

<!-- REST API Портала ГосЭДО: сервисы для цифровых документов -->
<!-- Для вызова используется REST API http(s)://server-url/#/SERVICE -->
<script>
  async function fetchAdapterService(myAction, myKey, myValue, saveAsFile,
mySession = 'NONE') {
    // Пример адреса для обращения к сервисам, если Портал ГосЭДО доступен по IP
10.77.142.21:84
    RESTAPIURL = 'http://10.77.142.21:84/api/service';
    const formData = new FormData();
    formData.append('action', myAction);
    formData.append(myKey, myValue);

    if (mySession !== 'NONE')
      formData.append('session_id', mySession);

    var object = {};
    formData.forEach(function(value, key){ object[key] = value; });

    const response = await fetch(RESTAPIURL, {
      method: 'POST',
      body: formData
    }).catch((error) => {});

    if (response.status === 200) {
      //Присвоение в переменную file ответа
      let file = await response.blob();
      let link = document.createElement("a");
      link.href = window.URL.createObjectURL(file);
      link.download = saveAsFile;
      link.click();
    }
  }
</script>

<!--REST API Портала ГосЭДО: получение данных процесса-->
<script>
  // функция, которая выполняется при нажатии кнопки "Получить данные процесса"
  async function getProcessInfo() {
    myValue = '1a3340e4-207e-4f7d-95c1-4d7c02d9d7ad';
    //передача параметров в функцию fetchAdapterService:
    //Команда: getProcessInfo

```

```
//Параметры: process_guid, session_id
//Значение параметра: myValue = 'ПроцессУИД' (уникальный идентификатор
процесса)
//сохранить результат как response.json
fetchAdapterService('getProcessInfo', 'process_guid', myValue,
'response.json', document.querySelector('#sessionId').value);
}
</script>
<!-- поле для ввода идентификатора сессии -->
<input id="sessionId" type="text" placeholder="session_id (от 15 знаков)"/>
<!-- кнопка инициации выполнения запроса-->
<button onClick="getProcessInfo()">Получить данные процесса</button>
```

С выполнением демонстрационного примера по описанной функции можно ознакомиться в разделе 4.9 Демонстрация получения информации о состоянии цифрового процесса.

## 7 Возможности интеграции сервисов работы с ГАС, НСИ и СППР

Необходимо обеспечить возможность обновления сведений ГАС и НСИ на используемом экземпляре «Портала ГосЭДО» для поддержания в актуальном состоянии справочников, которые являются источником унифицированных сведений для заполнения полей, формируемых Программой (перечень организаций, виды событий, виды документов, шаблоны структурированных данных и т.п.).

Работа с сервисом СППР позволит пользователям запрашивать и получать статистические отчеты.

Для реализации возможности работы пользователя СЭД с запросами к сервисам ГАС, НСИ и СППР можно интегрировать соответствующие разделы «Портала ГосЭДО» в интерфейс СЭД с помощью `<iframe>`.

В этом случае сервисы открываются как веб-страница внутри СЭД. Пользователь может отправлять запросы и просматривать ответы на них, работая в интерфейсе «Портала ГосЭДО», интегрированном в СЭД.

Для отображения интерфейса «Портала ГосЭДО», открытого в нужном разделе, следует использовать следующие адреса:

- <http://server-url/#/gas-requests> — раздел «Запросы в ГАС»;
- <http://server-url/#/gas-reference> — раздел «Справочник ГАС»;
- <http://server-url/#/nsi-requests> — раздел «Запросы в НСИ»;
- <http://server-url/#/nsi-reference> — раздел «Справочники НСИ»;
- <http://server-url/#/sppr-requests> — раздел «Запросы в СППР»;
- <http://server-url/#/sppr-reports> — раздел «Отчеты из СППР».

Необходимо ограничить ролевой доступ к работе с сервисами средствами СЭД, поскольку доступ к «Порталу ГосЭДО» происходит без авторизации.

Рекомендуется предоставить доступ к сервису работы с СППР делопроизводителю (запрос отчетов, просмотр отчетов).

Доступ к функциям работы с ГАС и НСИ рекомендуется предоставить администратору.

Некоторые данные справочников ГАС и НСИ, установленных на «Портале ГосЭДО», можно получить используя REST API.

В следующих разделах вы можете ознакомиться со следующей информацией:

- запрос данных из ГАС через REST API;
- запрос справочников НСИ через REST API.

## 7.1 Запрос данных из ГАС через REST API

С помощью API «Портала ГосЭДО» можно получить данные из справочника версии ГАС, установленной на «Портале ГосЭДО» на момент запроса.

Для отправки команды посредством POST необходимо использовать адрес "**http://server-url/api/gas-reference**", где «**server-url**» - адрес доступа к используемому экземпляру «Портала ГосЭДО». Команда передается в параметре «**action**».

**Команда:** `getTableItems`

**Действие:**

Возвращает данные организаций согласно версии ГАС, установленной на «Портале ГосЭДО» на момент запроса.

**Параметры (для отбора с фильтрацией и делением по страницам):**

```
filter:{"organization":"Минцифры"}
sort:{"organization":"asc/desc"}
paginator:{"page":1,"limit":10}
```

**Возвращаемый результат:**

Возвращает данные в формате json, которые включают массив **tableItems** с информацией об организациях из ГАС.

**Пример вызова в консоли без фильтрации:**

```
curl 'http://10.77.142.21:84/api/gas-reference' -H 'Content-Type: application/json' -d '{"action": "gettableItems"}'
```

### Пример вызова в консоли с фильтрацией:

```
curl 'http://10.77.142.21:84/api/gas-reference' -H 'Content-Type: application/json' -d '{"action": "gettableItems", "filter": {"title": "Аппарат", "isActive": true}, "sort": {"guid": "asc"}, "paginator": {"page":1,"limit":10}}'
```

### Пример реализации в JavaScript:

```
<script>
  async function fetchAdapterService(endpoint, myAction, myParams, saveAsFile = 'NONE') {
    let address = document.querySelector('#address');
    RESTAPIURL = 'http://10.77.142.21:84/' + endpoint;

    myParams['action'] = myAction;

    if (myParams['xml']) {
      var formData = new FormData();
      Object.entries(myParams).forEach(function([key, value]) {
        formData.append(key, value);
      });
      var REQUEST = {
        method: 'POST',
        body: formData,
      }
    } else {
      var REQUEST = {
        method: 'POST',
        body: JSON.stringify(myParams),
        headers: {'Content-Type': 'application/json'},
      }
    }

    const response = await fetch(RESTAPIURL, REQUEST)
    .catch( (error) => { });

    const fileName = ((saveAsFile == 'NONE') ? 'response.json' : saveAsFile);

    if (response.status == 200) {
      let file = await response.blob();
      let link = document.createElement("a");
      link.href = window.URL.createObjectURL(file);
      link.download = fileName;
      link.click();
    }
  }
}
```

```
</script>
<!------- REST API Портала ГосЭДО: получение ГАС -->
<script>
  async function getGAS() {
    myParams = {
      sort: {guid: 'asc'},
      filter: {title: 'Аппарат', },
      paginator: {page: 1, limit: 3},
    }
    fetchAdapterService('/api/gas-reference', 'gettableItems', myParams, 'get-
gas.json');
  }
</script>
<button class="header2" onclick="getGAS()"> Получить данные ГАС </button>
```

С выполнением демонстрационного примера по описанной функции можно ознакомиться в разделе 4.11 Демонстрация получения данных ГАС.

## 7.2 Запрос справочников НСИ через REST API

С помощью API «Портала ГосЭДО» можно получить данные справочников версии НСИ, установленной на «Портале ГосЭДО» на момент запроса.

Данные НСИ могут быть запрошены двумя способами.

### Способ 1

Для отправки команды посредством POST необходимо использовать адрес "**http://server-url/api/constructor**", где «**server-url**» - URL-адрес доступа к используемому экземпляру «Портала ГосЭДО». Команда передается в параметре «**action**».

**Команда:** getRefs

**Действие:**

Возвращает 3 справочника (видов взаимодействий, событий и электронных документов) установленной версии НСИ.

**Возвращаемый результат:**

Возвращает данные в формате json с массивами:

- **InformationInteractionsTypes** ([массив]) — содержимое справочника НСИ Cat01-002-00003 «Виды информационных взаимодействий о процессе»;



- **InformationInteractionsEvents** ([массив]) — содержимое справочника НСИ Cat01-002-00006 «Виды событий»;
- **ElectronicDocumentsTypes** ([массив]) — содержимое справочника НСИ Cat01-002-00002 «Виды электронных документов».

### Пример вызова в консоли:

```
curl 'http://10.77.142.21:84/api/constructor' -H 'Content-Type: application/json' -d '{"action": "getRefs"}'
```

### Пример реализации в Javascript:

```
<!-- REST API Портала ГосЭДО: сервисы для цифровых документов -->
<!-- Для вызова используется REST API http://server-url/#/constructor -->
<script>
  async function fetchAdapterService(myAction, myKey, myValue, saveAsFile,
mySession = 'NONE') {
    // Пример адреса для обращения к сервису НСИ, если Портал ГосЭДО доступен по
IP 10.77.142.21:84
    RESTAPIURL = 'http://10.77.142.21:84/api/constructor';
    const formData = new FormData();
    formData.append('action', myAction);

    if (mySession !== 'NONE')
      formData.append('session_id', mySession);

    var object = {};
    formData.forEach(function(value, key){ object[key] = value; });

    const response = await fetch(RESTAPIURL, {
      method: 'POST',
      body: formData
    }).catch((error) => {});

    if (response.status === 200) {
      //Присвоение в переменную file ответа
      let file = await response.blob();
      let link = document.createElement("a");
      link.href = window.URL.createObjectURL(file);
      link.download = saveAsFile;
      link.click();
    }
  }
</script>

<!--REST API Портала ГосЭДО: регистрация цифрового документа из файла XML
во внутреннем модуле Портала ГосЭДО-->
<script>
```

```

    // функция, которая выполняется при нажатии кнопки
    async function getNSI() {
        //передача параметров в функцию fetchAdapterService:
        //Команда: getRefs
        //сохранить результат как response.json
        fetchAdapterService('getRefs', '', '', 'response.json');
    }
</script>

<!-- кнопка для инициации запроса справочников НСИ-->
<button onClick="getNSI()"> Получить НСИ </button>

```

## Способ 2

Для отправки команды посредством POST необходимо использовать адрес **"http://server-url/api/nsi-reference"**, где «**server-url**» - URL-адрес доступа к используемому экземпляру «Портала ГосЭДО». Команда передается в параметре «**action**».

**Команда:** `getTableItems`

**Действие:**

Возвращает данные справочников НСИ, доступных в разделе «Справочники НСИ» Программы на момент запроса.

**Возвращаемый результат:**

Возвращает данные в формате json, которые включают массив **tableItems** с информацией о справочниках НСИ, доступных в разделе «Справочники НСИ» Программы на момент запроса.

**Параметры (для отбора с фильтрацией и делением по страницам):**

```

sort: {id: 'asc'}
filter: {id: 'Cat01-001-00001', }
paginator: {page: 1, limit: 3}

```

**Пример реализации в Javascript:**

```

<script>
    async function fetchAdapterService(endpoint, myAction, myParams, saveAsFile =
'NONE') {
        let address = document.querySelector('#address');
        RESTAPIURL = 'http://10.77.142.21:84/' + endpoint;

```

```

myParams['action'] = myAction;

if (myParams['xml']) {
  var formData = new FormData();
  Object.entries(myParams).forEach(function([key, value]) {
    formData.append(key, value);
  });
  var REQUEST = {
    method: 'POST',
    body: formData,
  }
} else {
  var REQUEST = {
    method: 'POST',
    body: JSON.stringify(myParams),
    headers: {'Content-Type': 'application/json'},
  }
}

const response = await fetch(RESTAPIURL, REQUEST)
.catch( (error) => { });

const fileName = ((saveAsFile == 'NONE') ? 'response.json' : saveAsFile);

if (response.status == 200) {
  let file = await response.blob();
  let link = document.createElement("a");
  link.href = window.URL.createObjectURL(file);
  link.download = fileName;
  link.click();
}
}
</script>
<!------- REST API Портала ГосЭДО: получение НСИ -->
<script>
  async function getNSI() {
    myParams = {
      sort: {id: 'asc'},
      filter: {id: 'Cat01-001-00001', },
      paginator: {page: 1, limit: 3},
    }
    fetchAdapterService('/api/nsi-reference', 'gettableItems', myParams,
'get-nsi.json');
  }
</script>
<button class="header2" onclick="getNSI()"> Получить данные НСИ </button>

```

С выполнением демонстрационного примера по описанной функции можно ознакомиться в разделе 4.12 Демонстрация получения данных НСИ.