

«ГАЗПРОМБАНК» (Акционерное общество)

|  |  |
| --- | --- |
|  | **УТВЕРЖДАЮ** |
|  |  |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г. |

**Редактор форм**

**Частное техническое задание на разработку Компонента «Редактор форм» для Платформы «G2 корпорация»**

на 12 листах

Москва 2023 г.

**Термины и сокращения**

|  |  |
| --- | --- |
| **Сокращение** | **Расшифровка** |
| **JavaScript** | <https://ru.wikipedia.org/wiki/JavaScript> |
| React | JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов. <https://reactjs.org> |
| SPA | Single Page Application – веб-приложение, использующее единственный HTML-документ как оболочку для всех веб-страниц |
| Typescript | <https://www.typescriptlang.org> |
| JSON | (англ. JavaScript Object Notation) – текстовый формат обмена данными, основанный на JavaScript. <https://json.org> |
| Компонент | Основная программная единица React-приложения, которая может инкапсулировать состояние, логику, и дизайн. |
| Функциональный компонент | Тип компонентов React, реализованный как функция. <https://reactjs.org/docs/components-and-props.html> |
| Платформа | Платформа «G2 корпорация». <https://reestr.digital.gov.ru/reestr/1258385/?sphrase_id=2518368> |
| Система | Информационная система, реализованная на платформе. |
| Операция | 1. Функциональный элемент Системы. Пользователь инициирует выполнение функций Системы путем выполнения операций через интерфейс пользователя. |
| Форма операции | Интерфейс пользователя для задания параметров операции и выполнения операции. |
| Параметры операции | Входные данные операции. |
| Параметр операции | Элемент входных данных операции. Характеризуется кодом, типом данных и дополнительными свойствами, определяющими его представление на форме операции. |
| Конфигурация формы операции | Текстовое представление формы операции в формате JSON |

**Общие положения**

Настоящий документ содержит требования для разработки Компонента «Редактор форм» для Платформы.

Входными данными Компонента является файл в формате JSON, содержащий список Параметров операции.

Результатом работы Компонента является файл в формате JSON, содержащий Конфигурацию формы операции.

Требования в данном документе определяют возможности интерфейса пользователя, типы входных и выходных данных, средства разработки и параметры программного окружения.

**Оглавление**

[1 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ 5](#_Toc125465678)

[2 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ 7](#_Toc125465679)

[Приложение 1 Эскиз интерфейса Компонента 8](#_Toc125465680)

[Приложение 2 Типы Параметров и типы полей ввода 10](#_Toc125465681)

[Приложение 3 Метаданные свойств элементов формы 11](#_Toc125465682)

[Приложение 4 Примеры входных и выходных данных, описание типов и интерфейсов 12](#_Toc125465683)

# ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

Функциональные требования для реализации Компонента представлены в таблице 1.

Таблица

|  |  |
| --- | --- |
| **Код** | **Требование** |
| **ФТ1** | **Интерфейс Компонента**  Компонент должен состоять из следующих областей:   1. Командная панель; 2. «Параметры» -- область входных данных; 3. «Форма» -- область структуры формы; 4. «Компоненты» -- область доступных для выбора элементов формы; 5. «Конфигурация» -- центральная область, область визуального и текстового представления Конфигурации формы; 6. «Настройки компоненты» -- область настраиваемых свойств элементов формы;   Эскиз интерфейса Компонента представлен в [Приложении 1](#Приложение1) |
| **ФТ2** | **Командная панель**  Командная панель должна содержать следующие команды:   * «Загрузить». При нажатии должно появиться модальное окно загрузки входных данных (см. [ФТ8](#ЗагрузкаВходныхДанных)). * «Сохранить». При нажатии должен скачаться файл, содержащий текущую конфигурацию формы в формате JSON. * «Режим». При нажатии должен меняться режим области «Конфигурация».   Командная панель может содержать другие команды.  Примеры файлов входных и выходных данных представлены в [Приложении 4](#_Приложение_4_Примеры) |
| **ФТ3** | **Область «Параметры»**  Область должна содержать список Параметров, полученных в результате загрузки входных данных (см. [ФТ8](#ЗагрузкаВходныхДанных)).  Область должна позволять осуществлять перетаскивание (drag-and-drop) Параметра в область «Конфигурации».  Типы параметров, типы полей ввода и их соответствие представлены в [Приложении 2](#Приложение2) |
| **ФТ4** | **Область «Форма»**  Область должна содержать текущую структуру Конфигурации формы.  Область должна позволять осуществлять:   * изменение состава, порядка и иерархии элементов дерева путем перетаскивания элементов внутри дерева (drag-and-drop); * удаление элементов из дерева;   Изменение порядка, иерархии, состава элементов в области «Форма» должно отражаться в области «Конфигурация» |
| **ФТ5** | **Область «Компоненты»**  Область должна содержать доступные для выбора элементы формы:   * Страницы * Страница * Группа * Поле ввода   Область должна позволять перетаскивать элементы в область «Конфигурация» |
| **ФТ6** | **Область «Конфигурация»**  Область должна иметь два режима работы: графический и текстовый.  В графическом режиме область должна позволять осуществлять:   * выбор элементов, расположенных на форме; * изменение порядка элементов формы путем перетаскивания (drag-and-drop); * изменение иерархии элементов формы: перемещение группы в группу, поля ввода в группу и из группы путем перетаскивания (drag-and-drop); * удаление элементов формы;   Изменение порядка, иерархии, состава элементов в области «Конфигурация» должно отражаться в области «Форма».  Точное позиционирование элементов на форме, отступы и интервалы не являются частью конфигурации формы – определяются Платформой.  В текстовом режиме область должна отображать текстовое представление текущей конфигурации формы. |
| **ФТ7** | **Область «Настройки компоненты»**  Область должна содержать свойства выбранного в области «Форма» или области «Конфигурация» элемента формы, доступные для редактирования пользователем.  Изменения значений свойств должны отражаться в области «Конфигурация».  Набор свойств различных элементов формы и полей ввода определяются на основании метаданных, приведенных в [Приложении 3](#Приложение3) |
| **ФТ8** | **Загрузка входных данных**  Загрузка входных данных должна осуществляться путем выбора файла на компьютере пользователя и его разбора.  В результате успешной загрузки входных данных:   * полученные Параметры должны отобразиться в области «Параметры»; * должна сформироваться конфигурация формы по умолчанию и отразиться в областях «Форма» и «Конфигурация».   В случае неуспешного разбора файла должно отобразиться сообщение об ошибке на русском языке. |

# НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

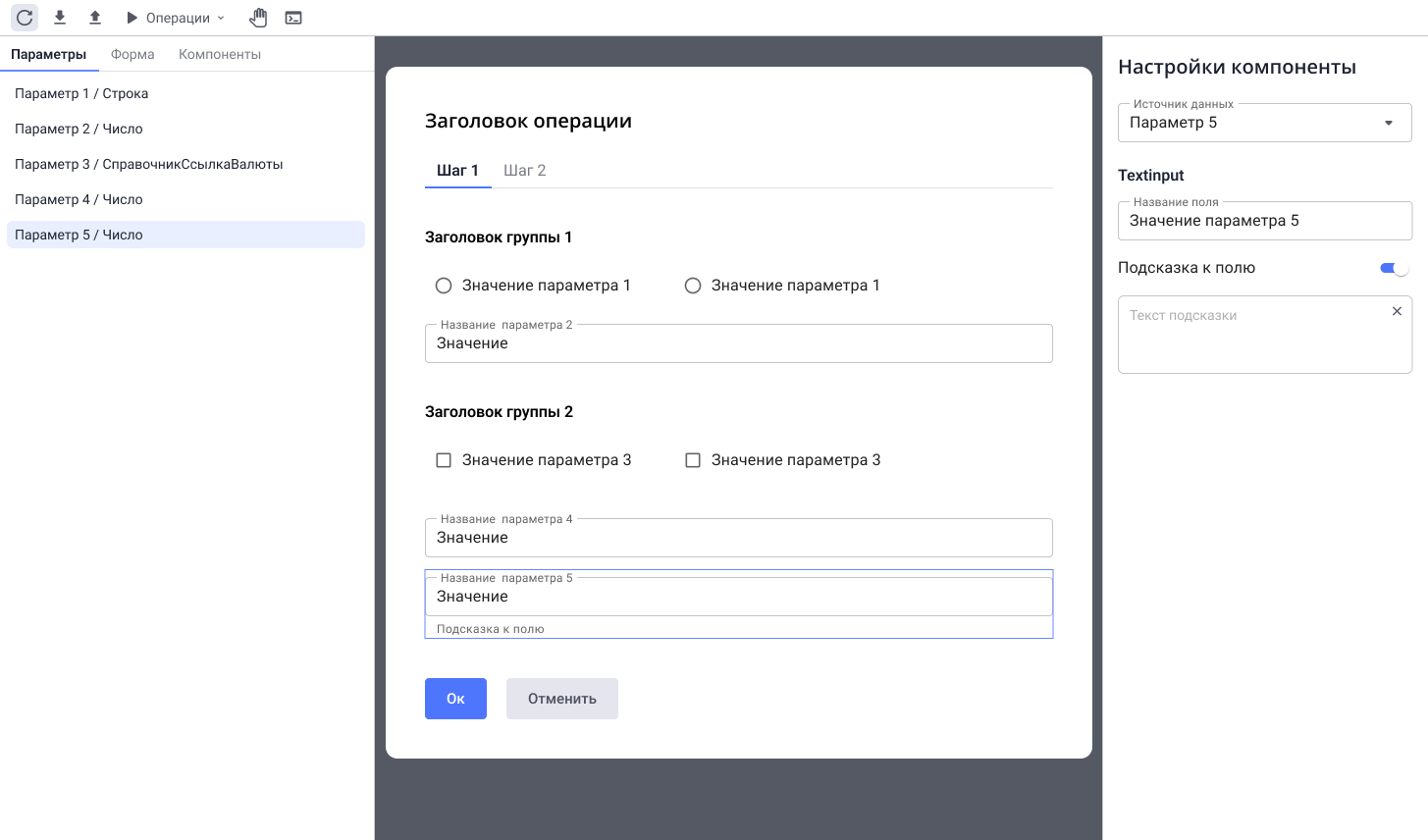
Нефункциональные требования для реализации Компонента представлены в таблице 2.

Таблица 2

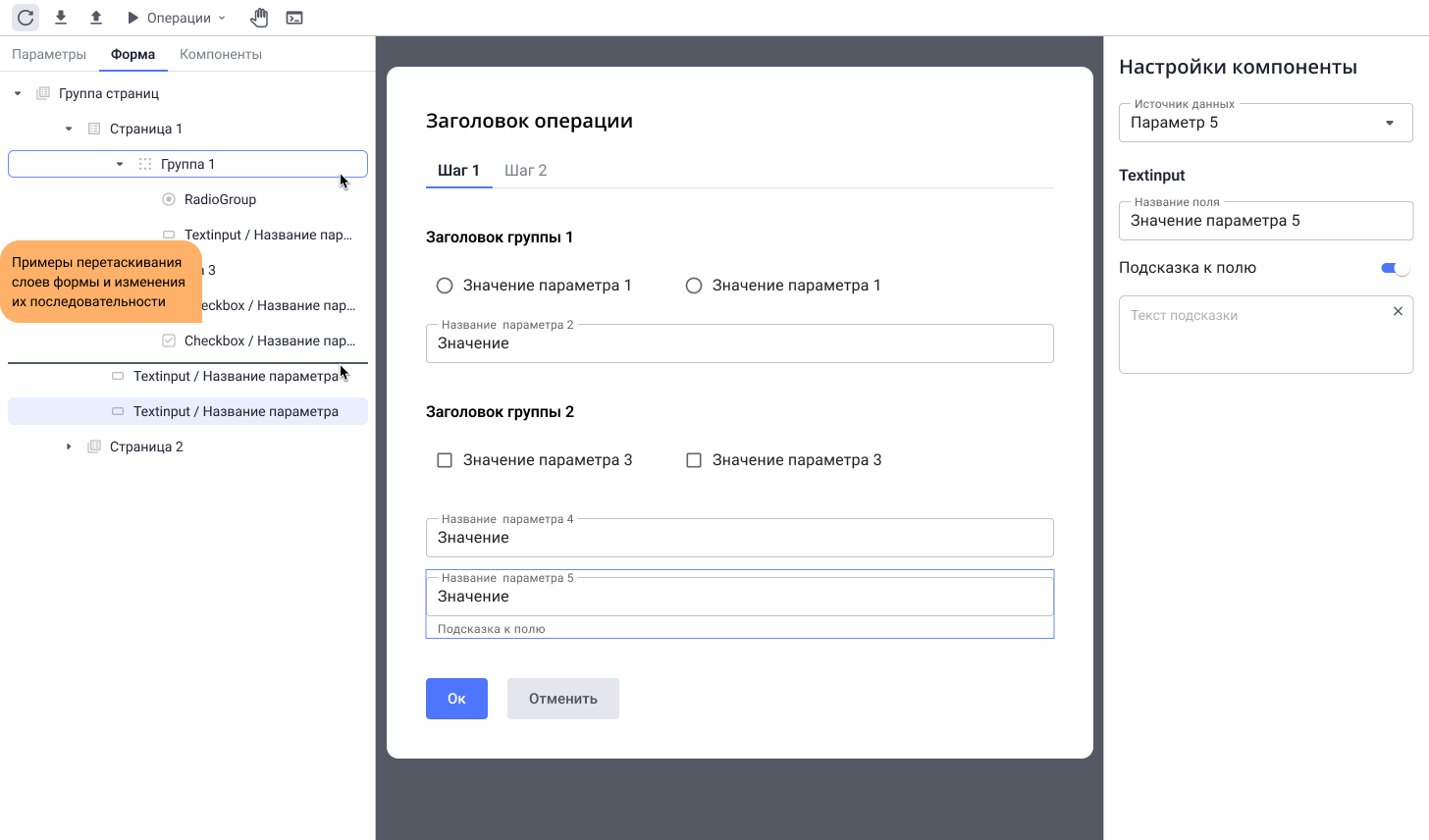
|  |  |
| --- | --- |
| **Код** | **Требование** |
| **НТ1** | Приложение должно быть выполнено в виде React SPA. |
| **НТ2** | При реализации приложения необходимо использовать шаблон из репозитория <https://gitlab.tmgdev.ru/art.zhiltsoff/form-editor> |
| **НТ3** | Для реализации визуальных компонентов необходимо использовать библиотеку Material UI. <https://mui.com> |
| **НТ4** | Все компоненты приложения должны быть Функциональными компонентами |
| **НТ5** | Для реализации эффекта drag-and-drop необходимо использовать библиотеку “react-smooth-dnd” |

# Приложение 1 Эскиз интерфейса Компонента

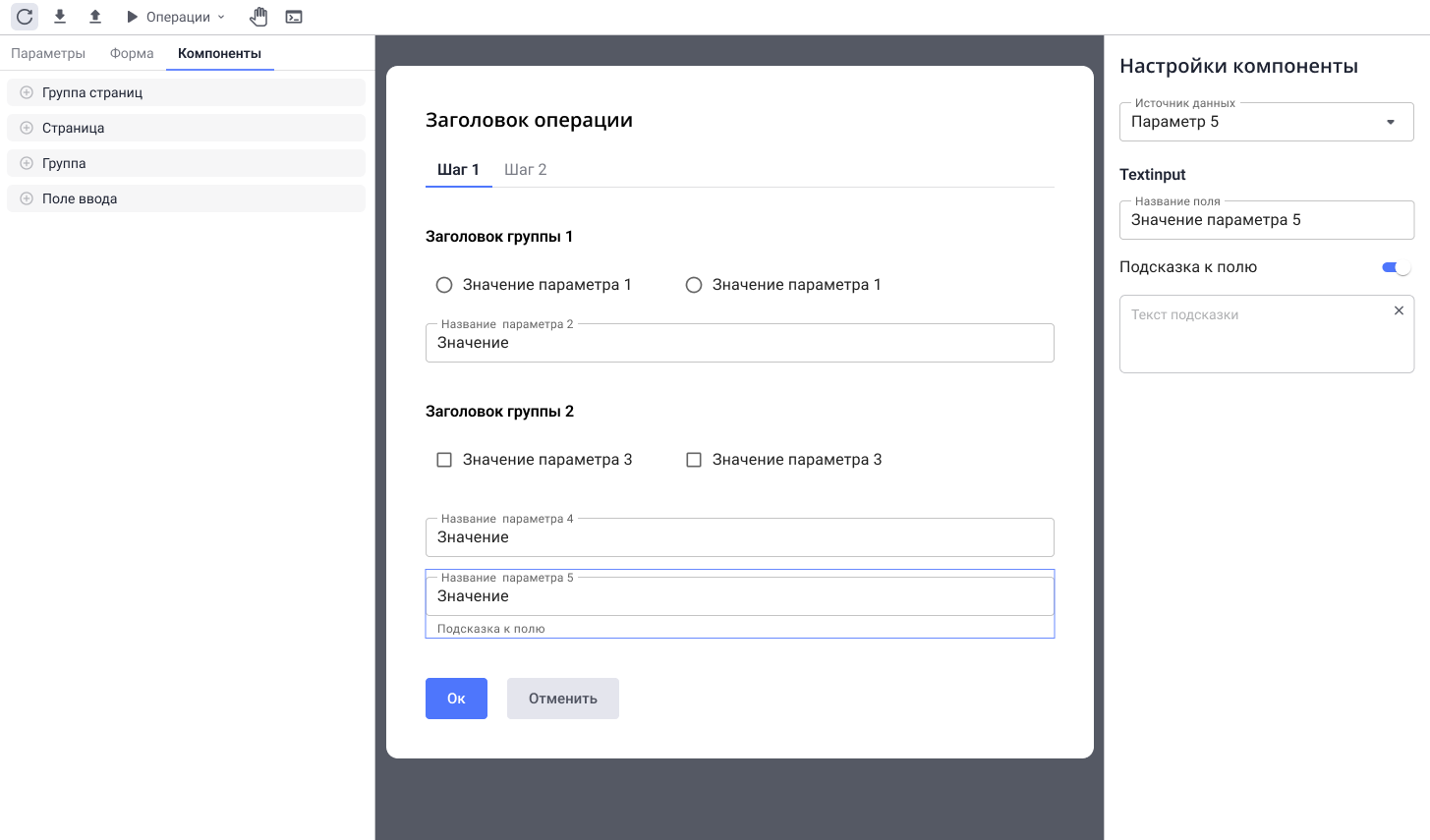
Область Параметры

****

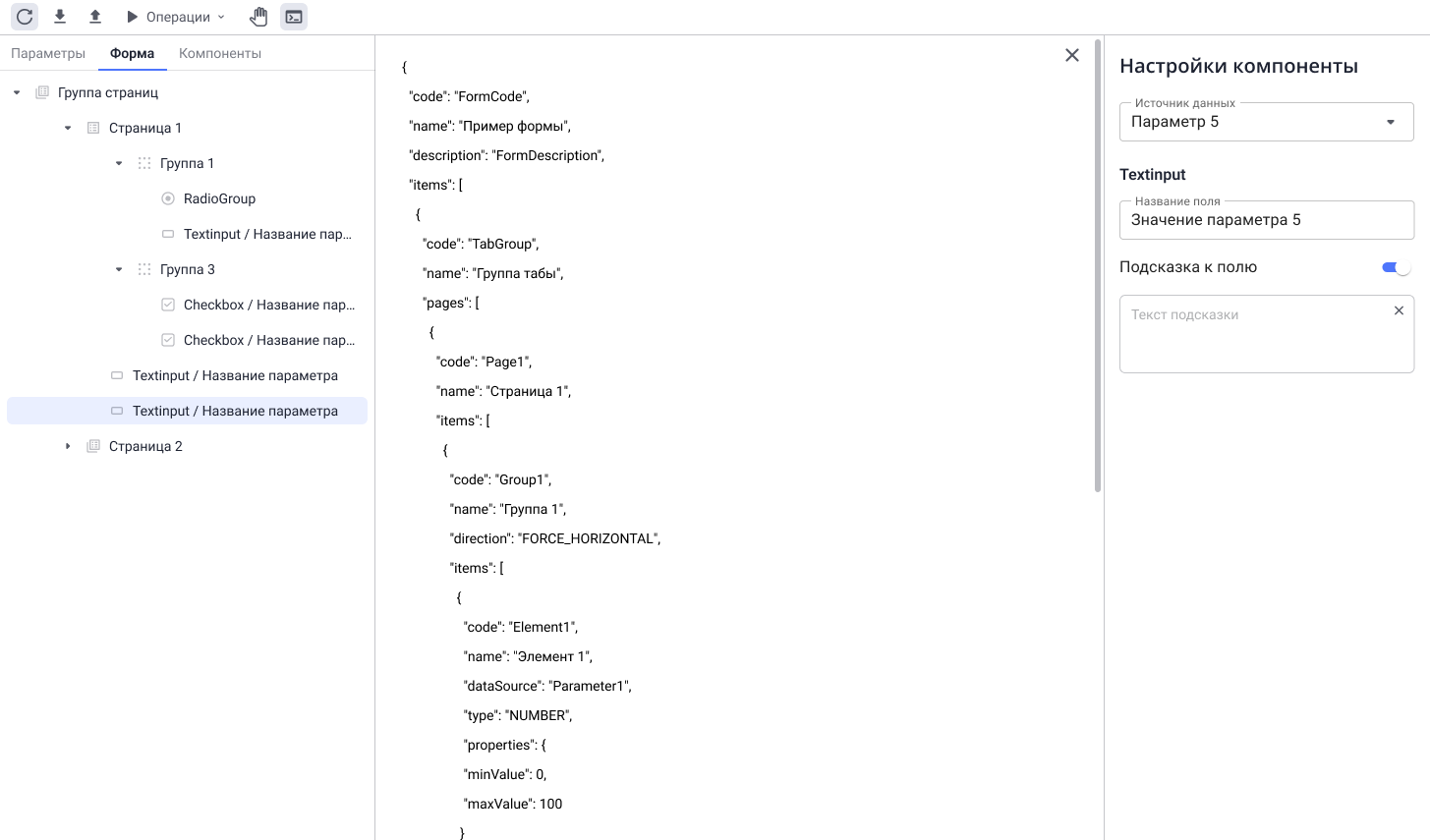
Область Форма

****

Область Компоненты



Текстовое представление конфигурации формы



# Приложение 2 Типы Параметров и типы полей ввода

Типы Параметров

|  |  |
| --- | --- |
| **Код** | **Наименование** |
| STRING | Строка |
| INTEGER | Целое число |
| NUMBER | Число с плавающей точкой |
| BOOLEAN | Булево |
| DATE | Дата |
| DATETIME | Дата-время |
| REF | Ссылка |
| FILE | Файл |

Типы полей ввода

|  |  |
| --- | --- |
| **Код** | **Наименование** |
| TEXT | Поле ввода текста |
| TEXTAREA | Многострочное поле ввода |
| NUMBER | Поле ввода числа |
| CHECKBOX | Чекбокс |
| SWITCH | Переключатель |
| DATEPICKER | Поле вводы даты |
| DATETIMEPICKER | Поле ввода даты и времени |
| FILE | Поле выбора файла |
| LINK | Ссылка |
| COMBOBOX | Выпадающий список |
| SELECT | Выпадающий список с возможностью множественного выбора и поиска |
| RADIOGROUP | Группа радио-кнопок |

Таблица соответствия типов Параметров и типов полей

|  |  |
| --- | --- |
| **Код типа параметра** | **Код типа поля ввода** |
| STRING | TEXT  TEXTAREA |
| INTEGER | NUMBER |
| NUMBER | NUMBER |
| BOOLEAN | CHECKBOX  SWITCH |
| DATE | DATEPICKER |
| DATETIME | DATETIMEPICKER |
| REF | LINK  COMBOBOX  SELECT  RADIOGROUP |
| FILE | FILE |

# Приложение 3 Метаданные свойств элементов формы

Набор свойств элементов формы зависит от:

* Типа элемента формы
* Типа параметра операции
* Типа поля ввода

Начальные метаданные свойств элементов формы представлены в файле properties.json:



Типы и интерфейсы, описывающие формат метаданных, представлены в файле property-metadata.ts



# Приложение 4 Примеры входных и выходных данных, описание типов и интерфейсов

Пример файла входных данных – списка параметров представлен в файле parameters.json:



Пример выходных данных – конфигурации формы представлен в файле form.json:



Файлы описаний типов и интерфейсов представлены в архиве interfaces.zip:



Все указанные материалы также находятся в репозитории проекта: <https://gitlab.tmgdev.ru/art.zhiltsoff/form-editor>