

---

# Read the Docs Template Documentation

*Выпуск 1.0*

Read the Docs

апр. 09, 2026



<b>1</b>	<b>О платформе Synergy</b>	<b>1</b>
1.1	Назначение платформы . . . . .	1
1.2	Какие задачи решает Synergy . . . . .	1
1.3	Роль аналитика в платформе . . . . .	2
1.4	Архитектура платформы (логический обзор) . . . . .	2
<b>2</b>	<b>Приложение в Synergy</b>	<b>3</b>
2.1	Назначение приложения . . . . .	3
2.2	Создание приложения . . . . .	3
2.3	Структура приложения . . . . .	4
2.4	Добавление объектов в приложение . . . . .	4
2.5	Управление приложением . . . . .	5
2.6	Связь приложения с дальнейшей настройкой . . . . .	7
<b>3</b>	<b>Форма</b>	<b>9</b>
3.1	Что такое форма в Synergy . . . . .	9
3.2	Создание формы . . . . .	9
3.2.1	Первичная настройка формы . . . . .	10
3.3	Структура формы . . . . .	11
3.3.1	Использование таблиц для разметки . . . . .	11
3.3.2	Вертикальная структура формы . . . . .	11
3.3.3	Горизонтальная структура формы . . . . .	12
3.3.4	Вложенные таблицы . . . . .	12
3.3.5	Динамическая таблица . . . . .	13
3.3.6	Общие принципы построения структуры . . . . .	13
3.4	Компоненты формы . . . . .	14
3.4.1	Текстовые компоненты . . . . .	14
3.4.2	Элементы выбора . . . . .	15
3.4.3	Файловые компоненты . . . . .	16
3.4.4	Специальные компоненты . . . . .	17
3.5	Валидация и формат данных . . . . .	18
3.5.1	Обязательные поля . . . . .	19
3.5.2	Маска ввода . . . . .	19
3.5.3	Регулярные выражения . . . . .	19
3.5.4	Блокировка от изменений . . . . .	19
<b>4</b>	<b>Форма завершения</b>	<b>21</b>

4.1	Типы форм завершения . . . . .	21
4.2	Форма завершения типа «Комментарий» . . . . .	21
4.3	Форма завершения типа «Файл» . . . . .	21
4.4	Форма завершения типа «Документ» . . . . .	22
4.5	Форма завершения типа «Форма» . . . . .	22
4.6	Форма завершения типа «Без результата» . . . . .	22
4.7	Связь с маршрутами бизнес-процесса . . . . .	23
<b>5</b>	<b>Подписание документов с использованием ЭЦП</b>	<b>25</b>
5.1	Назначение ЭЦП в системе . . . . .	25
5.2	Технические требования . . . . .	25
5.3	Использование этапа «Согласование» . . . . .	26
5.4	Настройка требований ЭЦП . . . . .	26
5.5	Возврат к настройке маршрута . . . . .	26
5.6	Поведение системы при подписании . . . . .	26
<b>6</b>	<b>Условные переходы</b>	<b>29</b>
6.1	Раздел «Переходы» . . . . .	29
6.2	Редактирование переходов . . . . .	29
6.3	Структура условия перехода . . . . .	31
6.4	Сравнение в условии . . . . .	31
6.5	Операторы сравнения . . . . .	31
6.5.1	Для числовых значений . . . . .	31
6.5.2	Для строковых значений . . . . .	31
6.6	Переход по умолчанию . . . . .	32
6.6.1	Запуск маршрута по шаблону . . . . .	32
6.6.2	Переход к этапу . . . . .	32
6.7	Рекомендации по использованию . . . . .	32
<b>7</b>	<b>Шаблоны маршрутов</b>	<b>35</b>
7.1	Назначение шаблонов маршрутов . . . . .	35
7.2	Создание шаблона маршрута . . . . .	36
7.3	Настройка основных параметров . . . . .	36
7.4	Использование шаблона маршрута . . . . .	36
7.5	Логика выполнения . . . . .	37
7.6	Рекомендации по использованию . . . . .	37
<b>8</b>	<b>Функции искусственного интеллекта</b>	<b>39</b>
8.1	Распознавание текста (OCR) . . . . .	39
8.1.1	Назначение функции . . . . .	39
8.1.2	Техническая реализация . . . . .	39
8.1.3	Настройка модели ИИ . . . . .	40
8.1.4	Подготовка формы . . . . .	40
8.1.5	Настройка этапа маршрута . . . . .	41
8.1.6	Использование функции . . . . .	41
8.1.7	Поведение системы . . . . .	41
<b>9</b>	<b>Импорт и экспорт данных реестров</b>	<b>43</b>
9.1	Экспорт и импорт реестров приложения . . . . .	43
9.2	Импорт и экспорт реестра через CSV . . . . .	43
9.2.1	Экспорт реестра в CSV . . . . .	43
9.2.2	Импорт реестра из CSV . . . . .	44

## 1.1 Назначение платформы

Synergy - это low-code платформа для проектирования, автоматизации и исполнения бизнес-процессов, а также для создания прикладных бизнес-приложений без необходимости классической разработки.

Основная идея Synergy заключается в том, что бизнес-логика, пользовательский интерфейс и структура данных описываются конфигурационно и визуально, а платформа берет на себя их исполнение, хранение и сопровождение.

---

**Примечание:** В контексте обучения под разработкой в Synergy понимается не программирование, а проектирование и настройка бизнес-решений средствами платформы.

---

## 1.2 Какие задачи решает Synergy

Платформа Synergy предназначена для решения следующих задач:

- автоматизация бизнес-процессов;
- снижение времени внедрения бизнес-решений;
- уменьшение зависимости от квалификации разработчиков;
- обеспечение единого подхода к данным, процессам и пользовательским интерфейсам.

Synergy позволяет реализовать полный жизненный цикл бизнес-процесса: от ввода данных пользователем до их обработки, маршрутизации, контроля исполнения и хранения результата.

## 1.3 Роль аналитика в платформе

В типовом сценарии работы аналитик получает описание бизнес-процесса (ордер, техническое задание, регламент) и на его основе:

- определяет вход и выход процесса;
- описывает структуру данных;
- проектирует пользовательские формы;
- настраивает реестры для хранения и просмотра данных;
- моделирует маршрут выполнения процесса;
- определяет роли и участников процесса.

Все перечисленные действия выполняются средствами Synergy без необходимости написания серверного кода.

## 1.4 Архитектура платформы (логический обзор)

С точки зрения пользователя платформа Synergy состоит из двух ключевых частей:

- **Среда проектирования (Synergy IDE)** - используется для создания и настройки приложений, форм, реестров и маршрутов;
- **Среда исполнения (Synergy Server)** - отвечает за выполнение процессов, хранение данных, работу пользователей и интеграции.

Аналитик работает преимущественно в Synergy IDE, создавая конфигурацию будущего приложения. После публикации конфигурации приложение становится доступным пользователям в рабочей среде Synergy.

## 2.1 Назначение приложения

Приложение в Synergy представляет собой изолированный набор объектов конфигурации, предназначенный для реализации конкретной бизнес-задачи.

В рамках приложения создаются и настраиваются формы, реестры, маршруты, справочники и другие объекты платформы. Все объекты приложения логически связаны между собой и управляются в рамках одного проекта.

## 2.2 Создание приложения

Работа с приложением выполняется в подсистеме Synergy IDE.

Для перехода в среду проектирования необходимо открыть в браузере адрес проекта с указанием пути `/designer`:

```
http://<адрес_сервера>/designer
```

После авторизации, при отсутствии созданных приложений, система предлагает создать новое приложение.


При создании приложения необходимо указать:

- наименование приложения;
- при необходимости - использовать структуру по умолчанию.

Использование структуры по умолчанию позволяет автоматически создать базовый набор папок для размещения объектов приложения.

## Новое приложение

\* Наименование

\* Код

Использовать структуру по умолчанию

Рис. 1: Создание нового приложения

## 2.3 Структура приложения

После создания приложения пользователю отображается его структура.

В зависимости от выбранных настроек приложение может содержать:

- одну корневую папку;
- или иерархию папок, созданную по умолчанию.

Папки используются для логической группировки объектов приложения и не влияют на их функциональное поведение.

Внутри папок могут размещаться следующие объекты:

- формы;
- реестры;
- маршруты;
- справочники;
- другие конфигурационные элементы.

## 2.4 Добавление объектов в приложение

Создание объектов выполняется из контекстного меню выбранной папки.

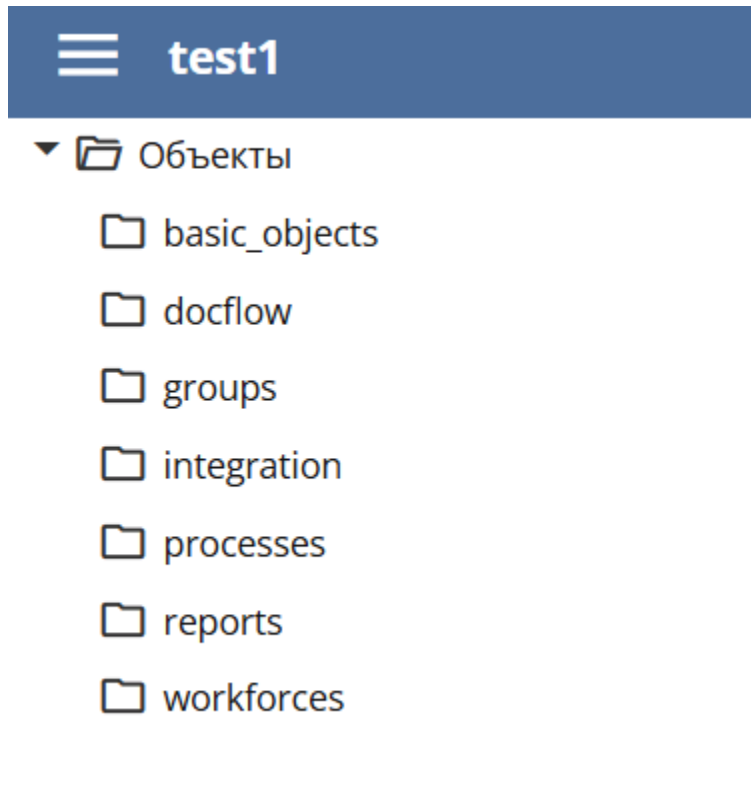


Рис. 2: Структура по умолчанию приложения в Synergy IDE

Для добавления объекта необходимо:

1. Выбрать папку в структуре приложения.
2. Нажать правой кнопкой мыши.
3. Выбрать пункт **Добавить**.
4. Выбрать тип создаваемого объекта.

Созданный объект автоматически становится частью текущего приложения и наследует его настройки доступа.

## 2.5 Управление приложением

В Synergy IDE предусмотрены возможности управления приложением, включая:

- редактирование объектов конфигурации;
- группировку объектов по папкам;
- перемещение объектов внутри приложения;
- импорт и экспорт приложения;
- управление правами доступа.

Экспорт и импорт приложения позволяют переносить конфигурацию между различными экземплярами платформы без потери структуры и зависимостей.

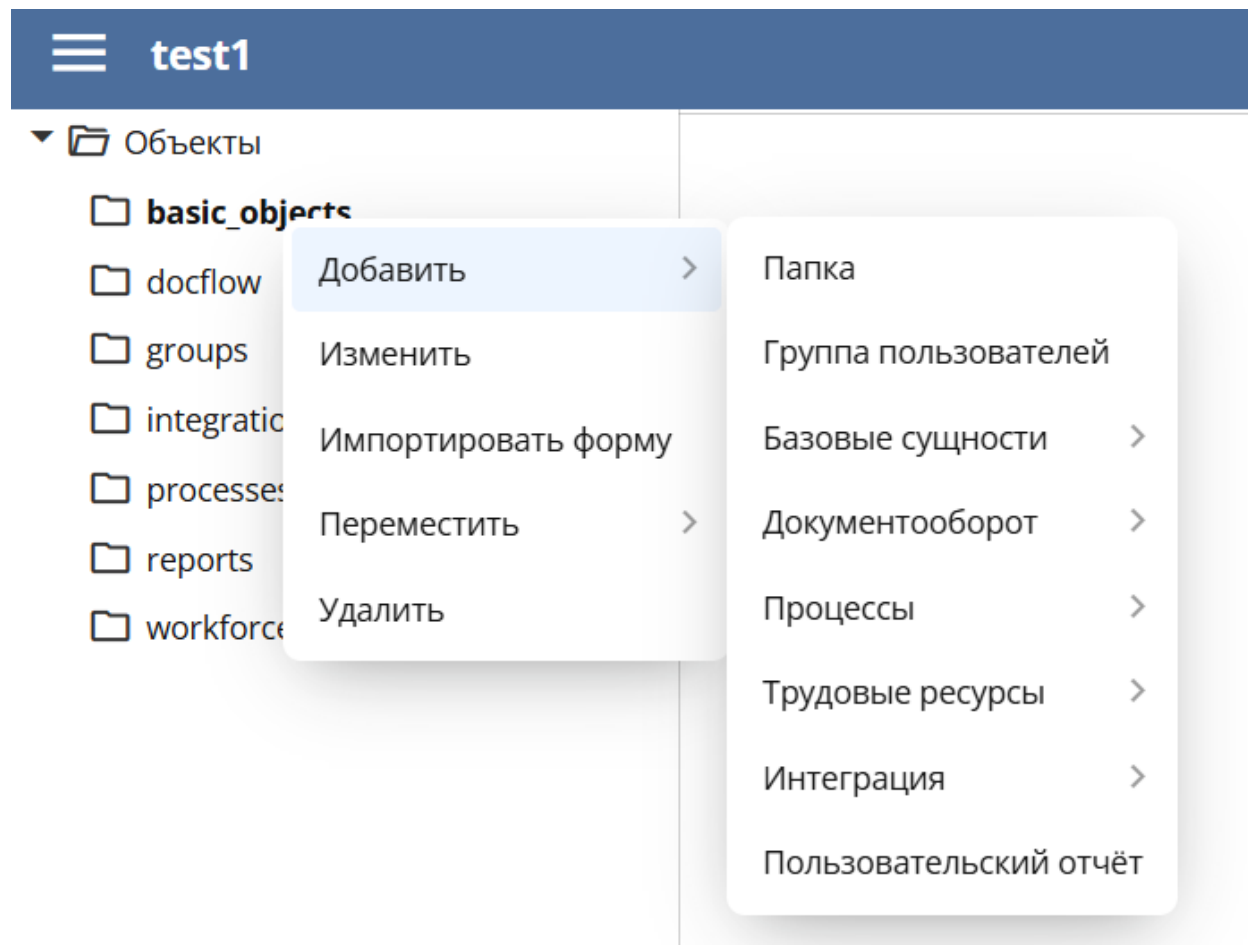


Рис. 3: Добавление объекта в приложение

## 2.6 Связь приложения с дальнейшей настройкой

Приложение является основой для последующей работы с платформой.

В следующих разделах рассматривается:

- создание форм для ввода данных;
- настройка реестров для хранения информации;
- моделирование маршрутов бизнес-процессов.

Все перечисленные элементы создаются и настраиваются внутри приложения.



### 3.1 Что такое форма в Synergy

Форма в Synergy является основным инструментом ввода и отображения данных в системе.

Форма представляет собой макет документа или заявки, структура которого определяется на основе ордера (технического задания) и описывает набор полей, их типы и правила заполнения.

В контексте бизнес-процесса форма используется как входная точка: пользователь заполняет форму, после чего данные передаются в систему для дальнейшего хранения, обработки и выполнения маршрута.

Форма определяет:

- какие данные вводятся пользователем;
- типы данных каждого поля;
- обязательность заполнения полей;
- визуальную структуру документа.

Создание формы выполняется в среде проектирования Synergy IDE и является одним из базовых этапов настройки бизнес-процесса.

### 3.2 Создание формы

Создание формы выполняется в среде проектирования Synergy IDE внутри ранее созданного приложения.

Для перехода в среду проектирования необходимо открыть в браузере адрес проекта с указанием пути `/designer` и пройти авторизацию в системе.

После создания или выбора приложения форма создается внутри нужной папки структуры приложения.

Для создания формы необходимо:

1. Выбрать папку в структуре приложения.
2. Открыть контекстное меню правой кнопкой мыши.
3. Выбрать пункт **Добавить** → **Базовые сущности** → **Форма**.

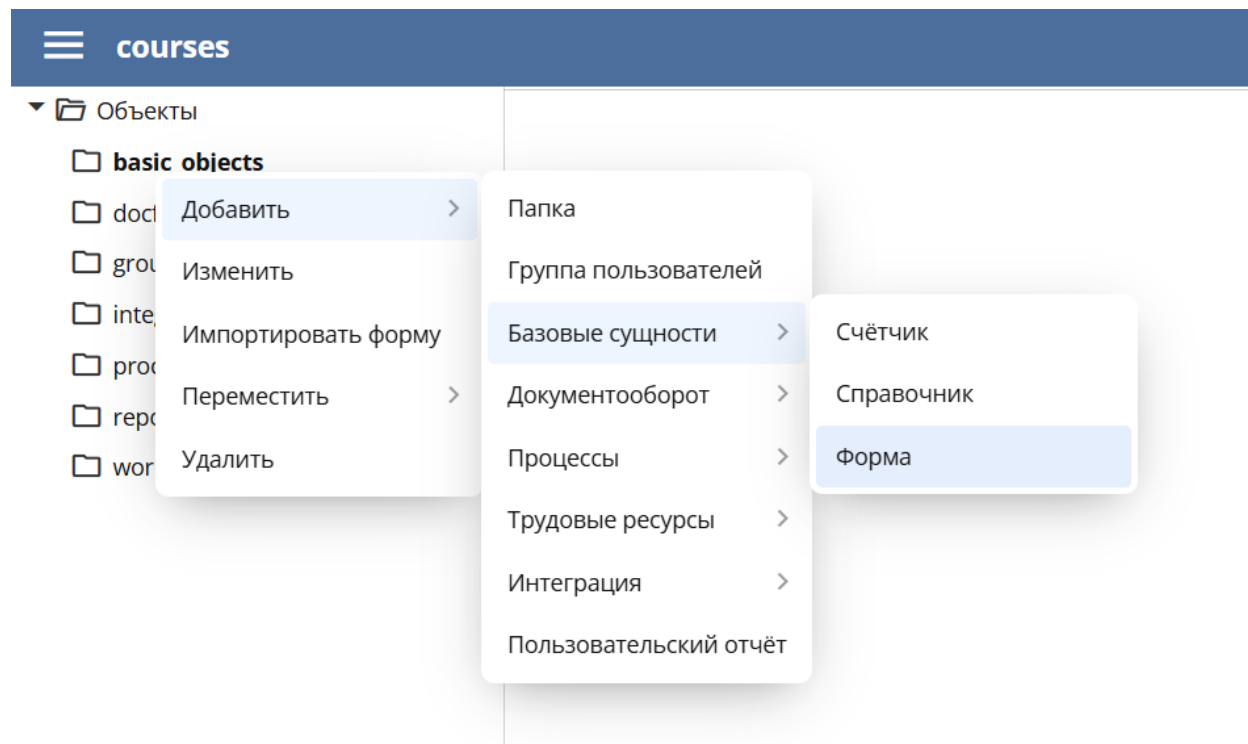


Рис. 1: Создание новой формы

После этого открывается окно редактирования формы. Созданная форма является макетом документа, который в дальнейшем будет использоваться для ввода данных пользователем.

### 3.2.1 Первичная настройка формы

Первичная настройка формы выполняется во вкладке «Форма» на правой панели редактора.

На данном этапе указываются:

- наименование формы;
- код формы.

При вводе наименования система автоматически формирует код формы на основе транслитерации. При необходимости код может быть изменен вручную.

После задания основных параметров выполняется переход к добавлению и настройке компонентов формы.

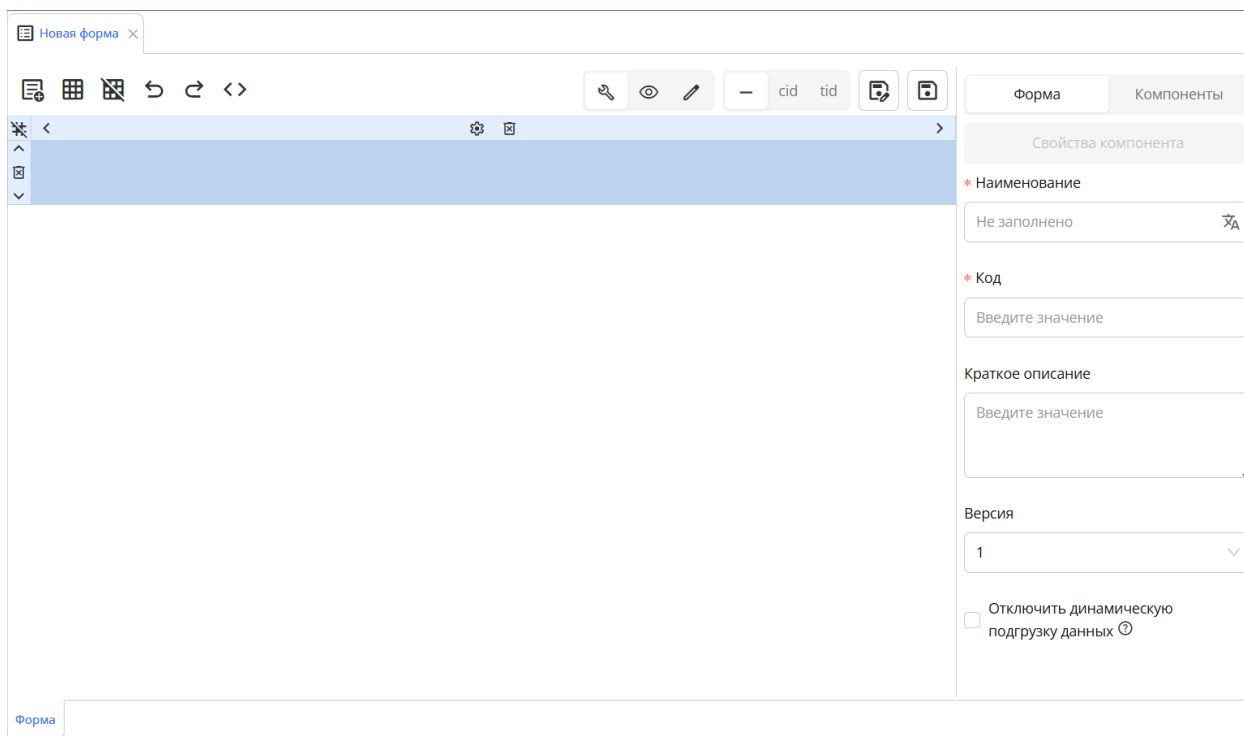


Рис. 2: Настройка формы

### 3.3 Структура формы

Структура формы определяется требованиями ордера и используется для удобного и логичного ввода данных пользователем.

Для построения структуры формы в Synergy применяется компонент «Таблица».

#### 3.3.1 Использование таблиц для разметки

Компонент «Таблица» используется для разметки формы и структурирования элементов внутри нее.

Таблица позволяет:

- разделять форму на логические блоки;
- выравнивать элементы по колонкам и строкам;
- группировать поля по смыслу;
- создавать динамические области для ввода данных.

В рамках формы таблицы применяется для организации пользовательского интерфейса.

#### 3.3.2 Вертикальная структура формы

Наиболее распространенная схема построения формы - вертикальное деление на две колонки:

- левая колонка - наименования полей;

- правая колонка - элементы для ввода данных.

Такая структура обеспечивает читаемость формы и упрощает восприятие пользователем вводимой информации.

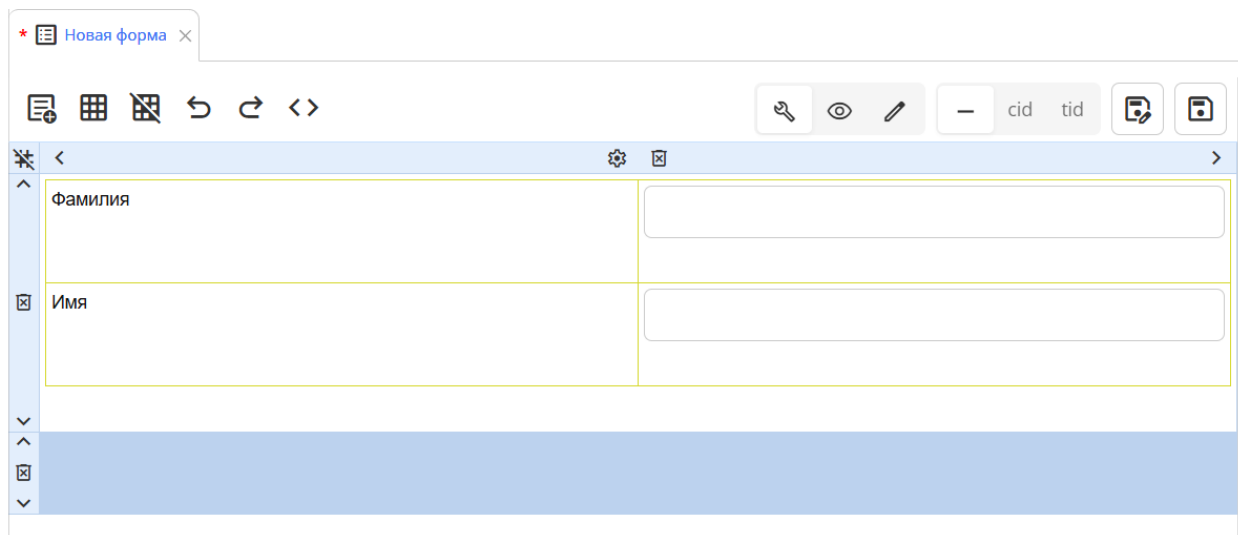


Рис. 3: Вертикальная структура формы

### 3.3.3 Горизонтальная структура формы

Помимо вертикального деления, форма может быть разделена горизонтально на несколько логических блоков.

Каждый блок, как правило, представляет собой отдельную таблицу с заголовком и содержит группу полей, относящихся к одной части заявки или документа.

Для визуального выделения блока:

- создается отдельная таблица;
- добавляется строка заголовка;
- заголовок выравнивается по центру.

### 3.3.4 Вложенные таблицы

Для сложных форм допускается использование вложенных таблиц.

Вложенные таблицы применяются для:

- отделения одного логического раздела от другого;
- создания многоуровневой структуры формы;
- оформления анкетных и дополнительных разделов.

Каждая таблица на форме настраивается независимо и может иметь собственные параметры отображения.

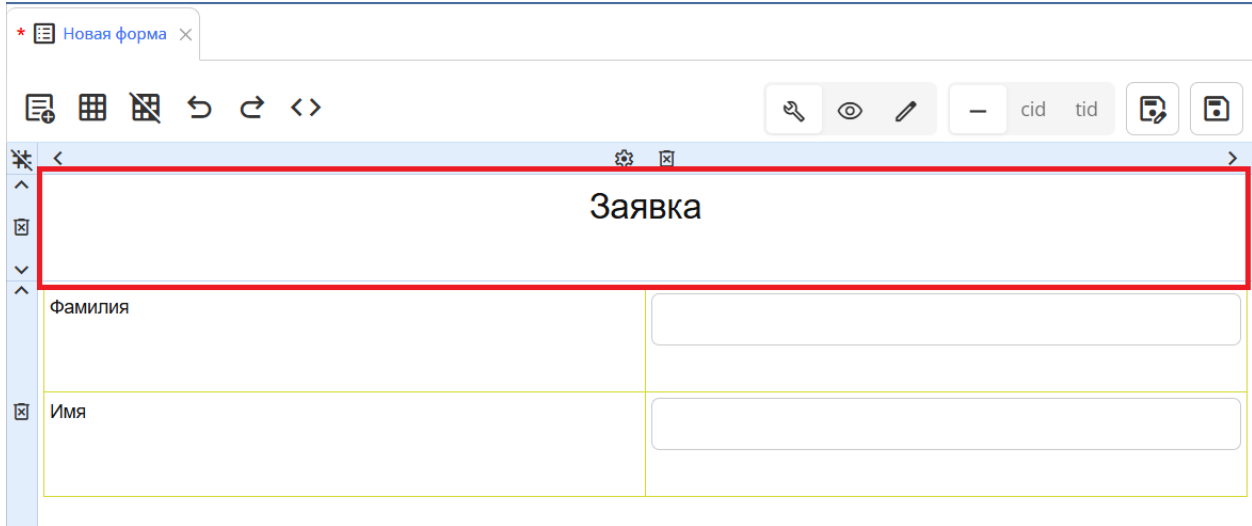


Рис. 4: Горизонтальное разделение формы на блоки

### 3.3.5 Динамическая таблица

Динамическая таблица используется в формах Synergy в тех случаях, когда пользователю необходимо вводить повторяющиеся данные с возможностью добавления новых строк во время заполнения формы.

Типовыми примерами использования динамической таблицы являются:

- перечень документов;
- список вложений;
- список позиций заказа;
- дополнительные сведения, количество которых заранее неизвестно.

Для создания динамической таблицы используется компонент «Таблица», для которого в настройках включаются параметры:

- «Добавлять строки в режиме заполнения»;
- «Добавить заголовок динамической таблицы».

После включения данных настроек таблица позволяет пользователю самостоятельно добавлять строки при заполнении формы.

### 3.3.6 Общие принципы построения структуры

При проектировании структуры формы рекомендуется:

- придерживаться структуры, описанной в ордере;
- избегать избыточного количества вложенных таблиц;
- группировать поля по смыслу;
- обеспечивать визуальную читаемость формы.

Корректно спроектированная структура формы упрощает дальнейшую настройку компонентов, реестров и маршрутов бизнес-процесса.

## 3.4 Компоненты формы

Компоненты формы используются для ввода, отображения и выбора данных пользователем.

Каждый компонент соответствует определенному типу данных и имеет набор настроек, определяющих правила ввода, отображения и поведения поля на форме.

Компоненты добавляются на форму во вкладке «Компоненты» и размещаются внутри таблиц, определяющих структуру формы.

В зависимости от назначения компоненты формы можно разделить на несколько основных групп.

### 3.4.1 Текстовые компоненты

Текстовые компоненты предназначены для ввода и отображения текстовых значений.

В Synergy используются следующие основные текстовые компоненты.

#### Однострочное поле

Компонент «Однострочное поле» используется для ввода коротких текстовых значений в одну строку без абзацев.

Типовые примеры использования:

- номер заказа;
- наименование;
- ФИО;
- телефон;
- адрес электронной почты.

Компонент поддерживает настройку обязательности заполнения, блокировку ввода данных пользователем, а также ограничение формата вводимых значений.

#### Многострочный текст

Компонент «Многострочный текст» используется для ввода произвольного текста, состоящего из нескольких строк и абзацев.

Типовые примеры использования:

- комментарии;
- описания;
- дополнительные сведения.

В отличие от однострочного поля, данный компонент позволяет вводить неограниченный объем текста с сохранением структуры абзацев.

### Неизменяемый текст

Компонент «Неизменяемый текст» используется для отображения статического текста на форме.

Как правило, данный компонент применяется для:

- наименований полей в левой колонке формы;
- заголовков разделов (блоков) на форме.

Текст задается в поле «Надпись». При необходимости можно добавить перевод.

### 3.4.2 Элементы выбора

Элементы выбора используются в формах Synergy для выбора одного или нескольких значений из заданного набора.

В качестве источника значений элементы выбора могут использовать:

- значения, заданные вручную;
- справочники, ранее созданные в системе.

Элементы выбора добавляются на форму во вкладке «Компоненты» и настраиваются через вкладку «Элементы».

### Выпадающий список

Компонент «Выпадающий список» предназначен для выбора одного значения из раскрывающегося списка.

В режиме заполнения формы пользователю отображаются значения из поля «Наименование», при этом системным значением является поле «Значение».

Список значений может быть:

- создан вручную непосредственно в компоненте;
- выбран из существующих справочников системы.

Компонент используется в тех случаях, когда необходимо выбрать одно значение из фиксированного набора.

### Переключатель вариантов

Компонент «Переключатель вариантов» также предназначен для выбора одного значения, но в отличие от выпадающего списка, варианты выбора отображаются пользователю сразу.

По принципу работы и настройкам компонент аналогичен выпадающему списку и использует те же источники данных.

Компонент применяется в случаях, когда количество вариантов невелико и важно, чтобы пользователь сразу видел все доступные значения.

### Выбор вариантов

Компонент «Выбор вариантов» используется для выбора одного или нескольких значений из списка с возможностью одновременного выбора нескольких пунктов.

Визуально компонент представляет собой список значений с флажками.

Источником значений для компонента, так же как и для других элементов выбора, может являться справочник или вручную заданный список.

Компонент применяется в тех случаях, когда пользователю необходимо отметить несколько вариантов одновременно.

### Дата / время

Компонент «Дата / время» используется для ввода пользователем значения даты и, при необходимости, времени.

В режиме заполнения формы компонент отображается в виде календаря, позволяющего выбрать конкретную дату. При включении соответствующей настройки дополнительно доступен ввод часов и минут.

Компонент применяется в тех случаях, когда необходимо указать дату события, дату создания, срок выполнения или иное календарное значение.

Для компонента доступны следующие основные настройки:

- обязательность заполнения;
- блокировка изменения значения пользователем;
- фильтрация данных;
- автоматическое заполнение текущей датой и временем;
- включение ввода часов и минут;
- задание формата отображения даты.

Формат даты может быть задан вручную с использованием шаблонов, например `#{dd}.#{mm}.#{yyyy}`.

### 3.4.3 Файловые компоненты

Файловые компоненты используются в формах Synergy для добавления и отображения файлов, изображений и ссылок, связанных с заявкой или документом.

Данные компоненты позволяют хранить дополнительные материалы в рамках бизнес-процесса, обеспечивая доступ к ним на всех этапах маршрута.

### Изображение

Компонент «Изображение» предназначен для загрузки и отображения графических файлов на форме.

После загрузки изображение отображается непосредственно в интерфейсе формы, что позволяет пользователю сразу видеть прикрепленный материал без необходимости скачивания файла.

Компонент применяется в случаях, когда визуальное представление данных имеет значение, например:

- сканы документов;
- фотографии;

- изображения, связанные с объектом заявки.

Загруженные изображения сохраняются в системе и доступны для просмотра на последующих этапах бизнес-процесса.

### Файл

Компонент «Файл» используется для загрузки файлов произвольного формата.

Компонент позволяет прикреплять документы и другие файлы, которые необходимо сохранить в рамках заявки или процесса.

В зависимости от настроек компонент может:

- открывать выбор файла с устройства пользователя;
- открывать выбор файла из хранилища системы;
- ограничивать допустимые форматы загружаемых файлов.

Загруженные файлы сохраняются в системе и доступны для скачивания пользователями, имеющими доступ к форме.

### Ссылка

Компонент «Ссылка» используется для сохранения URL-адреса внешнего ресурса, связанного с заявкой или документом.

Компонент позволяет указать текст надписи и сам URL-адрес, который открывается при нажатии.

Типовые примеры использования:

- ссылка на внешний документ;
- ссылка на информационный ресурс;
- ссылка на связанный процесс или систему.

На форме ссылка отображается в виде кликабельного элемента и может открываться в текущем или отдельном окне в зависимости от настроек.

## 3.4.4 Специальные компоненты

Специальные компоненты формы предназначены для работы с системными объектами Synergy, связями между данными и автоматической генерацией значений.

В отличие от стандартных компонентов ввода, специальные компоненты чаще всего используются для построения бизнес-логики формы и интеграции с другими элементами системы.

### Объекты Synergy

Компонент «Объекты Synergy» используется для выбора системных объектов в зависимости от заданного типа данных.

Тип данных определяет, какие объекты будут доступны для выбора. Возможные варианты включают, например:

- пользователей;

- должности;
- подразделения.

В зависимости от выбранного типа данных компонент позволяет:

- выбирать один или несколько объектов;
- автоматически заполнять значение данными создающего пользователя (например, должностью или подразделением);
- управлять отображением групп и структур объектов;
- настраивать язык отображения значений.

Компонент применяется в тех случаях, когда форма должна быть привязана к организационной структуре или пользователям системы.

### Ссылка на реестр

Компонент «Ссылка на реестр» используется для установления связи между формой и данными, хранящимися в реестрах Synergy.

С его помощью можно:

- выбрать запись из существующего реестра;
- разрешить или запретить создание новой записи;
- открыть реестр в диалоговом окне;
- управлять возможностью редактирования выбранной записи;
- использовать мультивыбор.

Дополнительно компонент поддерживает настройку сопоставления данных, которая позволяет автоматически связывать поля формы с полями выбранной записи реестра.

### Номер

Компонент «Номер» предназначен для автоматической генерации уникального значения на основе заданного шаблона.

Чаще всего данный компонент используется для формирования:

- номеров заявок;
- регистрационных номеров;
- внутренних идентификаторов документов.

Значение номера формируется системой автоматически и не требует ручного ввода пользователем.

Компонент поддерживает настройку шаблона номера, что позволяет использовать его в различных бизнес-сценариях.

## 3.5 Валидация и формат данных

Валидация и формат данных используются для контроля корректности вводимой информации и обеспечения целостности данных в форме.

С помощью настроек валидации можно ограничить ввод пользователя и задать правила заполнения полей в соответствии с требованиями бизнес-процесса.

### 3.5.1 Обязательные поля

Обязательное поле — это компонент формы, который должен быть заполнен пользователем перед сохранением или отправкой формы.

Если обязательное поле не заполнено, система не позволит продолжить работу с формой и уведомит пользователя о необходимости ввода значения.

Использование обязательных полей позволяет гарантировать наличие ключевых данных в записи.

### 3.5.2 Маска ввода

Маска ввода используется для задания фиксированного формата данных при вводе значения.

Она позволяет ограничить ввод символов и задать структуру значения, например для:

- телефонных номеров;
- идентификационных кодов;
- форматированных числовых значений.

Маска ввода помогает пользователю вводить данные в корректном виде и снижает вероятность ошибок.

### 3.5.3 Регулярные выражения

Регулярные выражения используются для проверки вводимых данных на соответствие заданному шаблону.

С их помощью можно реализовать более гибкую и точную валидацию, например для:

- адресов электронной почты;
- специальных кодов;
- пользовательских форматов данных.

Если введенное значение не соответствует заданному регулярному выражению, система уведомляет пользователя об ошибке.

### 3.5.4 Блокировка от изменений

Блокировка от изменений используется для запрета редактирования значения компонента пользователем.

Данный параметр применяется в случаях, когда значение:

- заполняется автоматически;
- рассчитывается системой;
- не должно быть изменено вручную.

Заблокированное поле отображается в форме, но недоступно для редактирования.



---

## Форма завершения

---

Форма завершения - это отдельная форма, которая открывается пользователю при завершении работы в рамках маршрута бизнес-процесса.

Форма завершения отображается в виде модального окна и определяет, какие действия должен выполнить пользователь для корректного завершения работы, а также какие данные или результаты будут зафиксированы системой.

### 4.1 Типы форм завершения

В системе предусмотрено пять типов форм завершения. Тип формы определяет, какой результат будет получен после завершения работы.

### 4.2 Форма завершения типа «Комментарий»

Данный тип используется, когда результатом завершения работы должен быть текстовый комментарий пользователя.

Особенности:

- пользователь вводит комментарий в текстовое поле;
- комментарий сохраняется в системе;
- комментарий может использоваться для истории работы или анализа причин принятого решения.

Тип «Комментарий» применяется, когда не требуется загрузка файлов или создание документов.

### 4.3 Форма завершения типа «Файл»

Форма завершения типа «Файл» используется в случаях, когда результатом работы является файл.

Пользователь может выбрать файл одним из следующих способов:

- загрузить файл с локального компьютера;
- скопировать файл из хранилища;
- выбрать файл, связанный с текущей работой: \* из раздела «Приложения»; \* из раздела «Прочие».

Таким образом обеспечивается гибкий выбор источника файла без необходимости повторной загрузки данных.

## 4.4 Форма завершения типа «Документ»

Данный тип формы завершения предназначен для выбора документа из списка, который формируется системой автоматически.

Список доступных документов определяется дополнительными параметрами, в том числе:

- является ли документ дочерним по отношению к документу текущей работы;
- зарегистрирован ли документ;
- относится ли документ к определенному реестру.

Если для формы завершения указан реестр, пользователь может создать новый документ, который будет являться дочерним по отношению к исходному документу.

Для этого используется специальная кнопка действия, которая по умолчанию называется «Создать ответ», но может быть переименована при необходимости.

## 4.5 Форма завершения типа «Форма»

Форма завершения типа «Форма» по логике работы аналогична типу «Файл», однако имеет принципиальное отличие.

В данном случае:

- файл не загружается из внешнего источника;
- файл формируется системой автоматически на основе указанной формы.

Этот тип используется, когда результат работы должен быть оформлен в виде структурированного документа, созданного по шаблону формы.

## 4.6 Форма завершения типа «Без результата»

Тип «Без результата» не предполагает создания каких-либо артефактов.

При использовании данного типа:

- не создается комментарий;
- не прикладывается файл;
- не создается документ или форма.

Этот тип имеет смысл использовать только в том случае, если для завершения работы не требуется подтверждение результата.

Если же подтверждение требуется, форма завершения типа «Без результата» по своему поведению аналогична обычному завершению работы без формы завершения.

## 4.7 Связь с маршрутами бизнес-процесса

Форма завершения привязывается к конкретному этапу маршрута.

После настройки формы завершения:

- пользователь не сможет завершить работу без выполнения заданных условий;
- система зафиксирует результат завершения;
- дальнейшее выполнение маршрута будет зависеть от корректного завершения этапа.



---

## Подписание документов с использованием ЭЦП

---

В системе ArtaSynergy предусмотрена возможность подписания документов с использованием электронной цифровой подписи (ЭЦП).

Подписание ЭЦП применяется в бизнес-процессах для юридически значимого подтверждения действий пользователя, например при согласовании или утверждении заявки.

### 5.1 Назначение ЭЦП в системе

Электронная цифровая подпись используется для:

- подтверждения согласия пользователя с содержанием документа;
- фиксации юридически значимого действия;
- обеспечения подлинности и неизменности данных;
- соответствия требованиям регламентов и законодательства.

В рамках платформы подписание ЭЦП реализуется через этап **«Согласование»** маршрута бизнес-процесса.

### 5.2 Технические требования

Для использования ЭЦП необходимо выполнение следующих условий:

- на рабочем месте пользователя должен быть установлен **NcaLayer**;
- в NcaLayer должен быть установлен модуль подписания **ArtaSynergy**;
- пользователь должен иметь действующий ключ и сертификат ЭЦП.

Без выполнения указанных требований подписание ЭЦП будет недоступно.

## 5.3 Использование этапа «Согласование»

Для реализации подписания ЭЦП в маршруте используется этап «Согласование».

Сам по себе этап «Согласование» не всегда требует наличия ключа и сертификата. Требование ЭЦП определяется системными настройками платформы и может быть включено или отключено администратором.

## 5.4 Настройка требований ЭЦП

Для включения подписания ЭЦП необходимо изменить настройки платформы.

Для этого:

1. Нажмите на иконку профиля в правом верхнем углу приложения.
2. Перейдите в раздел «**Настройки платформы**».
3. Откройте раздел «**Документооборот**».
4. Перейдите в подраздел «**Настройки документооборота**».

В открывшемся окне необходимо включить следующие параметры:

- **Подписывать согласования** — включает возможность подписания этапов согласования с использованием ЭЦП;
- **Требовать ключ и сертификат** — делает наличие ключа и сертификата обязательным для выполнения этапа согласования.

После установки параметров нажмите кнопку **Сохранить**.

## 5.5 Возврат к настройке маршрута

После сохранения настроек:

1. Вернитесь в дизайнер приложений.
2. Откройте ранее создаваемый маршрут бизнес-процесса.
3. Продолжите настройку этапов маршрута.

С этого момента этапы согласования, для которых включено подписание, будут требовать использование ЭЦП.

## 5.6 Поведение системы при подписании

При выполнении этапа согласования:

- пользователю будет предложено выбрать ключ ЭЦП;
- подписание выполняется через NcaLayer;
- результат подписания фиксируется в системе;
- без корректного подписания завершение этапа невозможно.

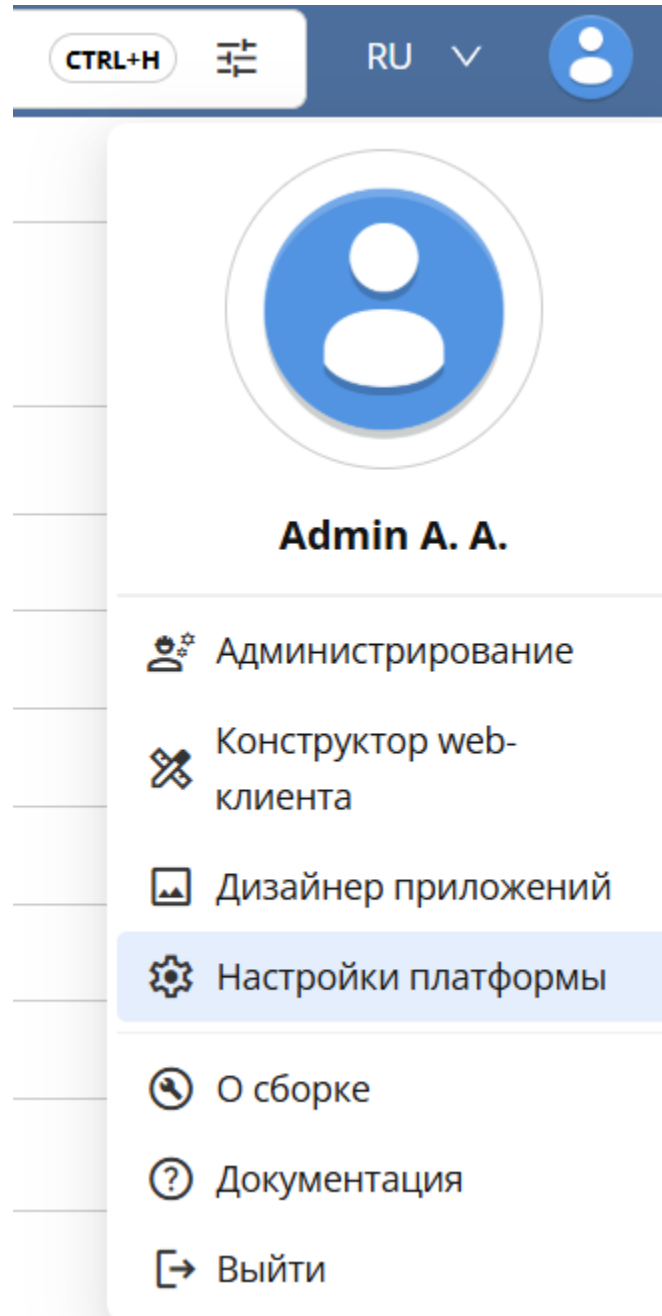


Рис. 1: Переход к настройкам документооборота

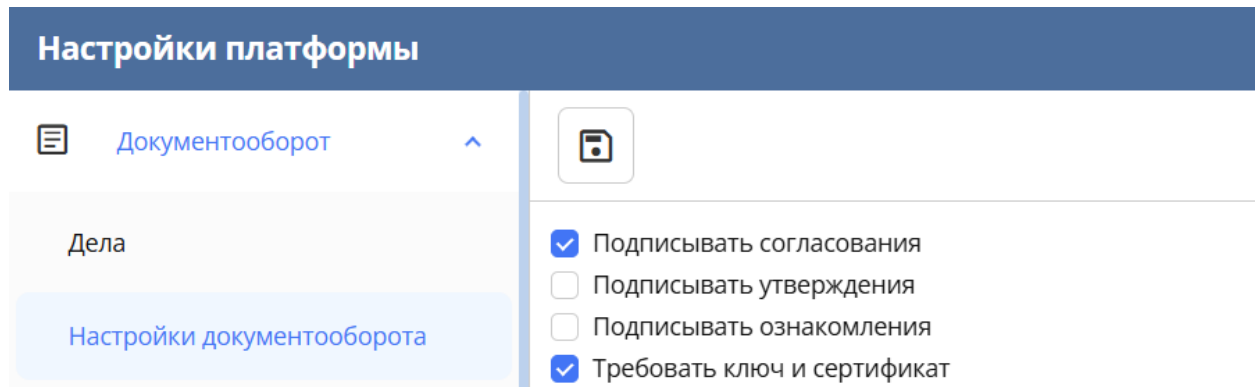


Рис. 2: Настройки подписания ЭЦП

---

## Условные переходы

---

Условные переходы используются в маршрутах бизнес-процессов для управления дальнейшим выполнением маршрута в зависимости от значений данных и результатов выполнения этапов.

С помощью условных переходов можно реализовать разветвленную логику, когда процесс развивается по разным сценариям в зависимости от введенных пользователем данных или состояния документа.

### 6.1 Раздел «Переходы»

Настройка условных переходов выполняется во вкладке **«Переходы»** соответствующего этапа маршрута.

Вкладка содержит:

- кнопку **«Редактировать»**;
- список настроенных переходов;
- переход по умолчанию с указанием результата выполнения.

### 6.2 Редактирование переходов

Для настройки условных переходов:

1. Откройте этап маршрута.
2. Перейдите во вкладку **«Переходы»**.
3. Нажмите кнопку **«Редактировать»**.

После этого откроется диалоговое окно **«Редактирование переходов»**, в котором настраиваются условия и результаты переходов.

**Параметры процесса “Условного перехода”**

**Переходы**

Действие 1

Если   =

То

**Переходы по умолчанию**

Рис. 1: Вкладка «Переходы» этапа маршрута

## 6.3 Структура условия перехода

Каждое условие перехода представляет собой набор сравнений, объединенных логическими операторами «И» или «ИЛИ».

Условие считается выполненным, если результат вычисления выражения равен *Истина*.

## 6.4 Сравнение в условии

Каждое сравнение состоит из трех элементов:

- **Левый операнд** — идентификатор (id) компонента формы или поля реестра, значение которого участвует в проверке;
- **Оператор сравнения** — тип сравнения между значениями;
- **Правый операнд** — значение, с которым выполняется сравнение.

## 6.5 Операторы сравнения

В системе доступны следующие операторы сравнения.

### 6.5.1 Для числовых значений

- =
- <
- >
- <=
- >=
- <>

### 6.5.2 Для строковых значений

- совпадает
- не совпадает
- начинается с
- не начинается с
- заканчивается на
- не заканчивается на
- содержит
- не содержит

## 6.6 Переход по умолчанию

Переход по умолчанию используется в том случае, если ни одно из настроенных условий не было выполнено.

Для перехода по умолчанию доступны следующие варианты действий.

### 6.6.1 Запуск маршрута по шаблону

При выборе варианта «**Запустить маршрут по шаблону**» необходимо указать шаблон маршрута, который будет запущен автоматически.

Дополнительно настраиваются параметры:

#### Поле «Запускать от имени»

Определяет, от имени какого пользователя будет запущен маршрут.

Доступные варианты:

- **Не изменять** — маршрут запускается от имени пользователя, инициировавшего родительский процесс (по умолчанию);
- **Из компонента** — пользователь берется из компонента формы типа «**Объекты Synergy → Пользователи**»;
- **Указать** — пользователь выбирается вручную через стандартный компонент выбора пользователя.

#### Поле «После выполнения»

Определяет поведение маршрута после завершения запущенного подмаршрута.

Доступные варианты:

- **Продолжить маршрут** — выполнение продолжается со следующим этапом основного маршрута;
- **Перейти к этапу** — выполняется переход к указанному этапу основного маршрута (по номеру или коду этапа).

### 6.6.2 Переход к этапу

Вариант «**Перейти к этапу**» используется, когда необходимо напрямую перейти к определенному этапу основного маршрута без запуска дополнительного маршрута.

В настройках указывается номер или код этапа, к которому должен быть выполнен переход.

## 6.7 Рекомендации по использованию

При работе с условными переходами рекомендуется:

- использовать понятные и однозначные условия;

- избегать избыточных сравнений;
- предусматривать переход по умолчанию;
- проверять корректность всех веток маршрута.

Корректно настроенные условные переходы обеспечивают предсказуемое и управляемое поведение бизнес-процесса.



---

## Шаблоны маршрутов

---

Шаблон маршрута - это отдельная сущность системы, предназначенная для повторного использования логики маршрута в разных бизнес-процессах.

По своей структуре и функциональности шаблон маршрута аналогичен обычному маршруту: в нем можно настраивать этапы, использовать различные типы действий и управлять логикой выполнения процесса.

Шаблон маршрута является отдельной сущностью системы и не привязан напрямую к реестру или пользовательскому действию.

Он используется в составе других маршрутов для выполнения переиспользуемых участков логики и может быть задействован как на уровне этапов маршрута, так и в логике условных переходов.

### 7.1 Назначение шаблонов маршрутов

Шаблоны маршрутов применяются в случаях, когда бизнес-процесс может развиваться по нескольким сценариям или содержит повторяющиеся участки логики.

Использование шаблонов маршрутов позволяет:

- выносить общие части маршрута в отдельную сущность;
- переиспользовать одну и ту же логику в разных маршрутах;
- упрощать основные маршруты и делать их более наглядными;
- уменьшать дублирование настроек и этапов.

Таким образом, шаблон маршрута выступает в роли переиспользуемого блока логики, который может быть встроен в основной маршрут на определенном этапе.

## 7.2 Создание шаблона маршрута

Для создания шаблона маршрута необходимо:

1. В структуре приложения выбрать папку, в которой будет располагаться шаблон маршрута.
2. Кликнуть правой кнопкой мыши по выбранной папке.
3. В контекстном меню выбрать: **Добавить** → **Процессы** → **Шаблон маршрута**.

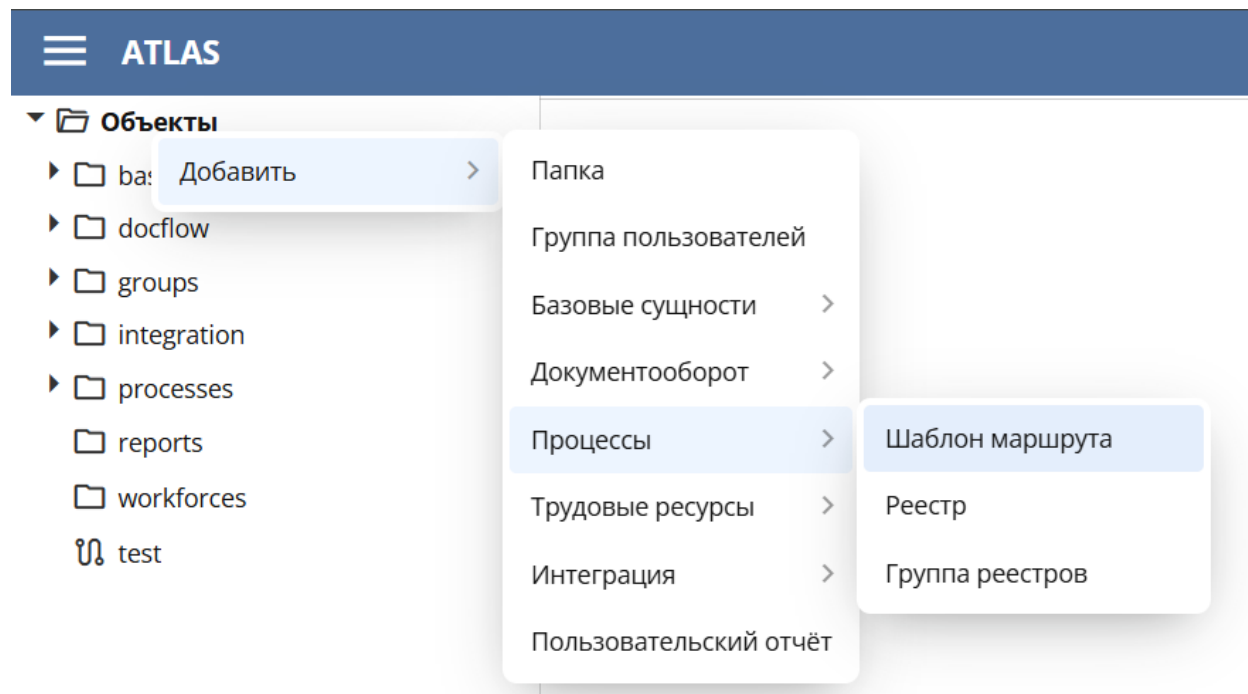


Рис. 1: Создание шаблона маршрута

После этого откроется окно создания шаблона маршрута.

## 7.3 Настройка основных параметров

В окне создания шаблона маршрута необходимо:

1. Указать **Наименование** шаблона маршрута.
2. Задать **Код** шаблона маршрута либо оставить автоматически сгенерированное значение.
3. Сохранить шаблон маршрута, нажав на кнопку сохранения (иконка дискеты на панели).

После сохранения шаблон маршрута становится доступным его настройка и использование в других маршрутах.

## 7.4 Использование шаблона маршрута

Для использования шаблона маршрута в основном бизнес-маршруте необходимо настроить соответствующий этап.

Для этого:

1. Откройте основной маршрут бизнес-процесса.
2. Выберите этап, в котором должен выполняться шаблон маршрута.
3. В настройках этапа в разделе **Тип действия** выберите значение «**Запуск маршрута по шаблону**».
4. В поле «**Шаблон маршрута**» выберите необходимый шаблон из выпадающего списка.

После настройки этапа при выполнении данного этапа основного маршрута будет запускаться выбранный шаблон маршрута.

## 7.5 Логика выполнения

При запуске шаблона маршрута:

- выполняются все этапы, настроенные внутри шаблона;
- шаблон маршрута работает как самостоятельный маршрут;
- после завершения шаблона выполнение основного маршрута продолжается в соответствии с его логикой.

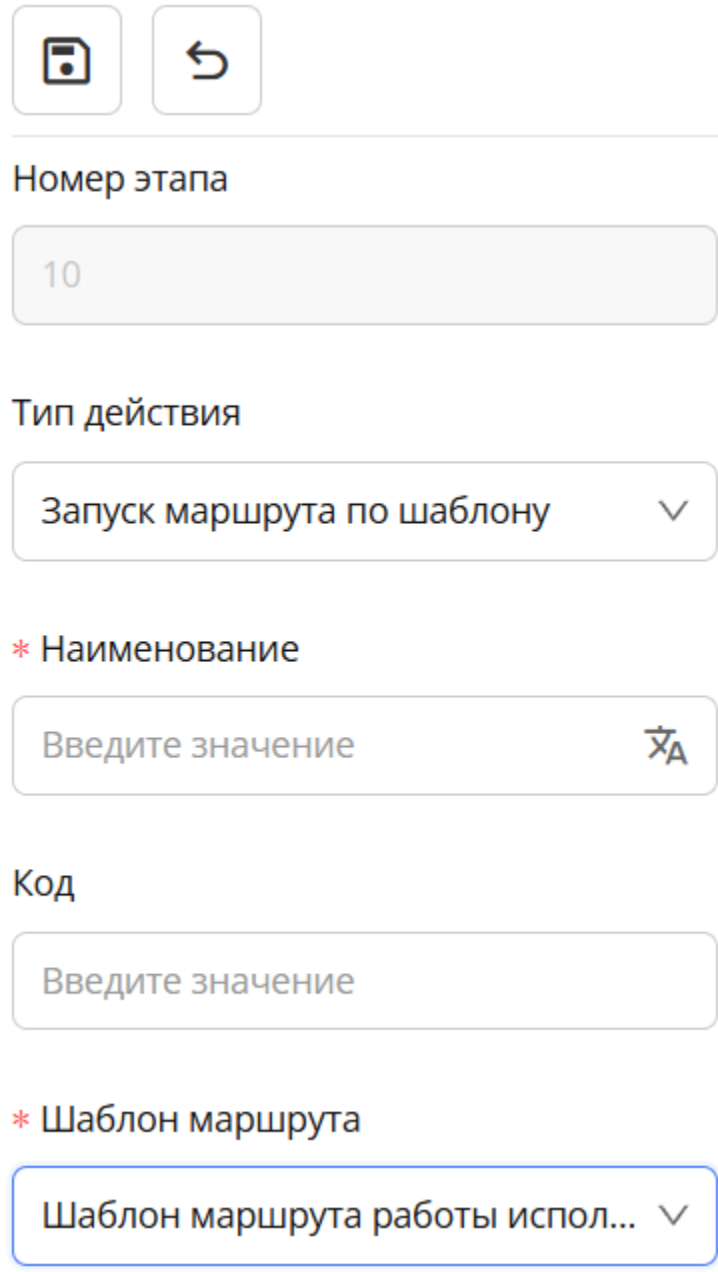
Таким образом, шаблон маршрута встраивается в основной маршрут как отдельный логический блок.

## 7.6 Рекомендации по использованию

При проектировании бизнес-процессов рекомендуется:

- использовать шаблоны маршрутов для повторяющихся участков логики;
- давать шаблонам понятные и однозначные названия;
- не перегружать основной маршрут большим количеством этапов;
- выносить сложные ветки логики в отдельные шаблоны маршрутов.

Корректное использование шаблонов маршрутов упрощает сопровождение системы и повышает читаемость бизнес-процессов.



Номер этапа

10

Тип действия

Запуск маршрута по шаблону

\* Наименование

Введите значение

Код

Введите значение

\* Шаблон маршрута

Шаблон маршрута работы испол...

Рис. 2: Выбор шаблона маршрута в этапе

## 8.1 Распознавание текста (OCR)

Функция распознавания текста (OCR) позволяет автоматически извлекать текст из файлов и бинарного содержимого документов в рамках бизнес-процесса на платформе Synergy.

Функция реализована на уровне маршрута реестра и выполняется как отдельный этап маршрута.

### 8.1.1 Назначение функции

Распознавание текста используется в случаях, когда в процессе обработки документа требуется:

- извлечь текст из загруженного файла (сканированного документа, изображения);
- автоматически заполнить текстовое поле формы на основе содержимого файла;
- исключить ручной ввод данных, уже присутствующих в документе.

Функция работает без участия пользователя: после запуска маршрута система самостоятельно обрабатывает файл и помещает распознанный текст в указанное поле.

### 8.1.2 Техническая реализация

Для обработки файлов используется self-hosted модель, работающая на CPU.

Ключевые характеристики:

- модель размещена в Docker-контейнере на отдельном хосте;
- при наличии GPU модель автоматически использует его для ускорения обработки;
- время обработки зависит от размера файла и объёма бинарного содержимого;
- использование мощных LLM не требуется.

В качестве альтернативы возможно подключение сторонних решений, включая модели OpenAI.

### 8.1.3 Настройка модели ИИ

Перед использованием функции необходимо выполнить настройку в разделе администрирования системы.

Для этого:

1. Перейдите в раздел **Администрирование**.
2. Откройте **Настройки системы**.
3. Перейдите в подраздел **Настройки ИИ**.
4. Нажмите «**Добавить функцию ИИ**»
5. Укажите **Тип функции** и **Адрес хоста**.
6. Опционально укажите **Модель ИИ** и **API ключ**
7. Сохраните настройки.

Рис. 1: Настройка ИИ функций

**Примечание:** В разделе **Настройки ИИ** можно настроить несколько различных моделей. Это позволяет использовать разные модели для разных задач.

### 8.1.4 Подготовка формы

Для работы функции OCR форма реестра должна содержать хотя бы два поля:

- **Поле файла** - поле, в которое пользователь загружает документ для распознавания;
- **Поле текста** - поле, в которое система запишет распознанный текст.

### 8.1.5 Настройка этапа маршрута

Функция OCR подключается на уровне этапа маршрута реестра.

Для настройки этапа:

1. Откройте маршрут реестра в Synergy IDE.
2. Добавьте новый этап или выберите существующий.
3. В настройках этапа в разделе **Тип действия** выберите «**Функция искусственного интеллекта**».
4. В поле «**Тип функции ИИ**» выберите «**Распознавание текста**».
5. В «**Поле-источник данных**» укажите **поле файла** - источник содержимого для распознавания.
6. В «**Поле-приемник данных**» укажите **поле текста** - поле, в которое будет записан результат.
7. Сохраните настройки этапа.

### 8.1.6 Использование функции

После настройки маршрута и этапа OCR работа с функцией выполняется в следующем порядке:

1. Откройте запись в реестре.
2. Загрузите файл в поле файла формы.
3. Запустите маршрут.
4. Дождитесь завершения этапа распознавания текста.
5. Откройте запись - распознанный текст будет отображён в текстовом поле формы.

---

**Примечание:** Время ожидания на этапе распознавания зависит от размера файла. Для больших документов обработка может занять больше времени.

---

### 8.1.7 Поведение системы

При выполнении этапа распознавания текста:

- система автоматически извлекает текст из указанного файла;
- распознанный текст записывается в поле текста документа;
- пользователю не требуется выполнять дополнительных действий;
- после завершения этапа маршрут продолжается в соответствии с настроенной логикой.

Workflow configuration interface for text recognition. The interface includes the following elements:

- Buttons for save and back.
- Тип действия** (Action type): `Функция искусственного интеллекта` (AI function).
- \* Наименование** (Name): `Введите значение` (Enter value).
- Код** (Code): `Введите значение` (Enter value).
- \* Тип функции ИИ** (AI function type): `Распознавание текста (OCR)` (Text recognition (OCR)).
- \* Поле-источник данных** (Data source field): `file_attachment`.
- \* Поле-приемник данных** (Data receiver field): `textarea_application_text`.

Рис. 2: Настройка этапа распознавания текста в маршруте

---

## Импорт и экспорт данных реестров

---

Платформа поддерживает экспорт и импорт данных реестров - как отдельного реестра, так и всех реестров приложения сразу. Это позволяет переносить приложения между средами вместе со справочными данными, без ручного переноса.

### 9.1 Экспорт и импорт реестров приложения

Механизм работает в связке с экспортом и импортом конфигурации приложения. При экспорте можно включить опцию, которая добавит данные реестров в состав архива. При импорте эти данные будут загружены в целевую среду.

Данные переносятся без соблюдения связей. Чтобы проверить целостность перенесённых данных, в каждый архив добавляется хэшсумма - отдельно для каждого реестра и для всех данных в целом.

Для работы с этим функционалом используйте API-эндпоинты `app_export` и `app_import`.

### 9.2 Импорт и экспорт реестра через CSV

Помимо переноса данных вместе с приложением, платформа поддерживает работу с данными реестра через CSV-файлы. Это позволяет загружать данные из внешних источников или выгружать срез записей реестра для дальнейшей обработки.

Обе операции выполняются в фоновом режиме.

#### 9.2.1 Экспорт реестра в CSV

При экспорте система формирует CSV-файл со срезом записей указанного реестра.

Структура файла строится автоматически на основе описания формы реестра - дополнительная настройка не требуется.

При необходимости можно ограничить выборку по диапазону дат создания записей.

Для запуска экспорта используйте API-эндпоинт `/api/v5/registry/csv/export`.

### 9.2.2 Импорт реестра из CSV

При импорте система читает CSV-файл и создаёт записи в указанном реестре по указанной форме.

Чтобы система понимала, какая колонка CSV соответствует какому полю формы, необходимо передать маппинг - описание соответствия колонок и полей.

Маппинг поддерживает два режима:

- **По имени колонки** - если CSV-файл содержит строку заголовков, можно указать имя колонки так, как оно записано в заголовке.
- **По индексу колонки** - если заголовков нет, колонки указываются по их порядковому номеру, начиная с нуля.

Использовать оба режима в одном маппинге нельзя - нужно выбрать один из них.

По умолчанию разделителем колонок считается запятая. При необходимости можно указать другой символ-разделитель.

Для запуска импорта используйте API-эндпоинт `/api/v5/registry/csv/import`.