

# Действия системы

Версия 8.0



Эта документация предоставляется с ограничениями на использование и защищена законами об интеллектуальной собственности. За исключением случаев, прямо разрешенных в вашем лицензионном соглашении или разрешенных законом, вы не можете использовать, копировать, воспроизводить, переводить, транслировать, изменять, лицензировать, передавать, распространять, демонстрировать, выполнять, публиковать или отображать любую часть в любой форме или посредством любые значения. Обратный инжиниринг, дизассемблирование или декомпиляция этой документации, если это не требуется по закону для взаимодействия, запрещены.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления и не может гарантировать отсутствие ошибок. Если вы обнаружите какие-либо ошибки, сообщите нам о них в письменной форме.

# Содержание

<b>Элемент процесса [Читать данные]</b>	<b>4</b>
Читать первую запись из выборки	4
Считать функцию	5
Считать количество записей	5
Читать коллекцию записей	5
Примеры использования	6
<b>Элемент процесса [Добавить данные]</b>	<b>6</b>
Добавить одну запись	7
Добавить результат выборки	7
Примеры использования	8
<b>Элемент процесса [Изменить данные]</b>	<b>8</b>
Примеры использования	9
<b>Элемент процесса [Удалить данные]</b>	<b>9</b>
Примеры использования	9
<b>Элемент процесса [Формула]</b>	<b>9</b>
<b>Элемент процесса [Вызвать веб-сервис]</b>	<b>10</b>
Настроить параметры элемента Вызвать веб-сервис	11
Передать параметры вызова типа “коллекция”	13
Проверить ошибки вызова веб-сервисов	16
Обработать отклик веб-сервиса	16
Настроить параметры ответа типа “коллекция”	18
<b>Элемент процесса [Изменить права доступа]</b>	<b>19</b>
Настроить свойства элемента Изменить права доступа	20
Примеры использования	22
<b>Элемент процесса [Прогнозировать данные]</b>	<b>22</b>
Примеры использования	23
<b>Элемент процесса [Задание-сценарий]</b>	<b>23</b>
Методы Get и Set	25
Соответствие типов параметров Creatio и C#	25
Примеры работы с различными типами параметров	26
<b>Элемент процесса [Привязать процесс к объекту]</b>	<b>29</b>
<b>Элемент процесса [Выполнить действие процесса]</b>	<b>30</b>
<b>Элемент процесса [Обработать файл]</b>	<b>32</b>
Настроить обработку файлов в ходе процесса	34
Настроить генерацию отчетов в ходе процесса	36
<b>Элемент процесса [Поиск и объединение дублей]</b>	<b>39</b>

# Элемент процесса [Читать данные]

ПРОДУКТЫ: **ВСЕ ПРОДУКТЫ**

Элемент [ *Читать данные* ] ([Рис. 1](#)) получает из базы данных значения полей необходимых записей для дальнейшего использования в ходе бизнес-процесса.

Рис. 1 — Элемент процесса [ *Читать данные* ]

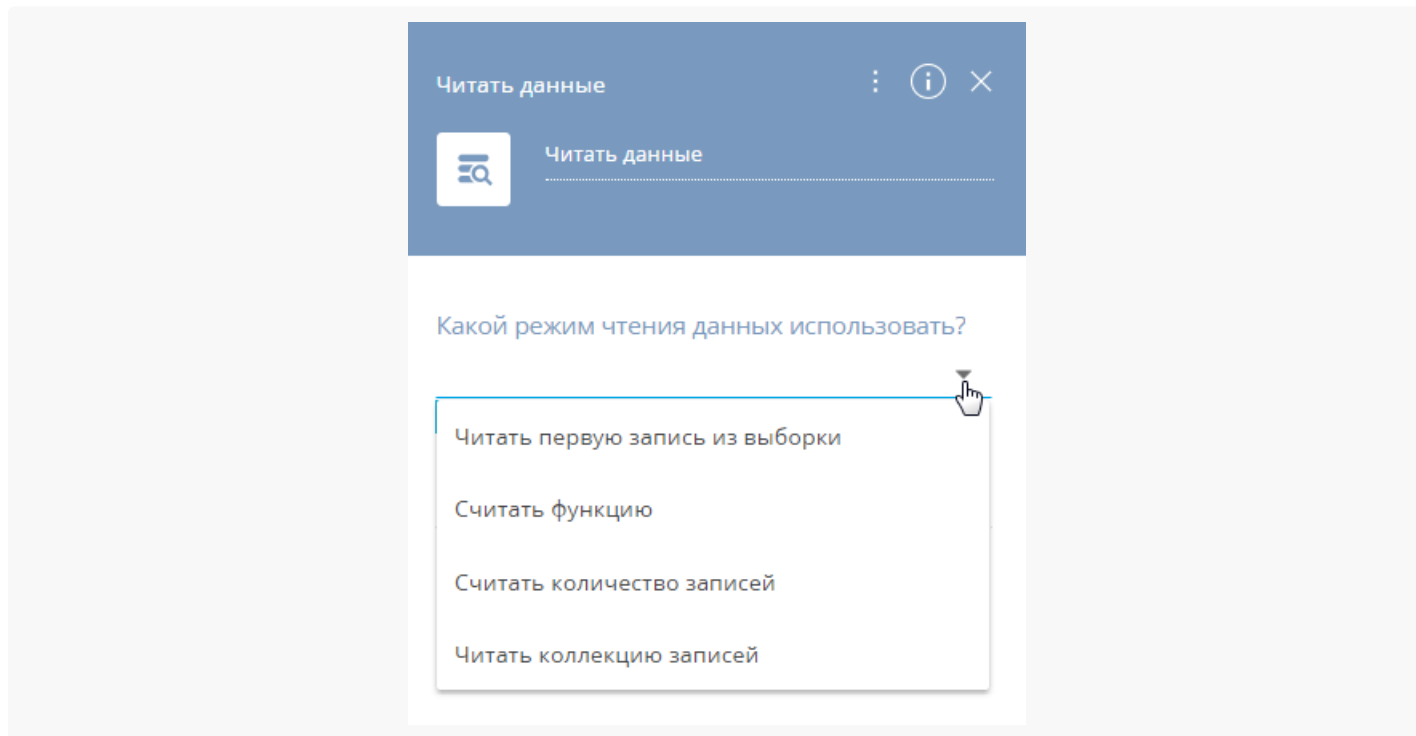


Всякий раз, когда процессу необходимо работать с базой данных Creatio, должны быть получены (“вычитаны”) значения соответствующих полей. Полученные данные хранятся в исходящих параметрах элемента [ *Читать данные* ].

Элемент может вычитать данные из любого объекта независимо от прав доступа пользователя, который запускает процесс.

Есть четыре режима чтения данных ([Рис. 2](#)).

Рис. 2 — Выбор режима чтения данных



## Читать первую запись из выборки

Элемент [ *Читать данные* ] получает список записей, которые соответствуют фильтру, сортирует их и

читает значения полей первой записи из отфильтрованного и отсортированного списка. Например, можно вычитать заголовок, тип определенной активности или ответственного по ней.

## Считать функцию

Элемент [ *Читать данные* ] получает список записей, которые соответствуют фильтру, и подсчитывает указанную функцию (сумму, среднее, минимальное или максимальное значение). Например, можно подсчитать длительность всех активностей определенного пользователя за период. Возможные варианты функций:

- Сумма (только по числовым полям) — элемент вычисляет сумму значений в определенных полях записей, соответствующих условиям фильтрации.
- Среднее (только по числовым полям) — элемент вычисляет среднее арифметическое значений определенного поля в записях, соответствующих условиям фильтрации.
- Минимальное (по датам и числовым полям) — элемент анализирует соответствующие условиям фильтра записи и определяет минимальное значение в указанном поле.
- Максимальное (по датам и числовым полям) — элемент анализирует соответствующие условиям фильтра записи и определяет максимальное значение в указанном поле.

Результат функции передается в исходящий параметр элемента.

## Считать количество записей

Элемент [ *Читать данные* ] получает список записей, которые соответствуют фильтру, а затем подсчитывает количество записей в списке и передает результат в свой исходящий параметр. Например, можно подсчитать количество контрагентов определенного менеджера по работе с клиентами. Также этот режим используется для проверки наличия отдельных записей в системе, например, есть ли в базе данных контакт с определенным ФИО или номером телефона.

## Читать коллекцию записей

Элемент [ *Читать данные* ] получает список записей, которые соответствуют фильтру, сортирует их и вычитывает значения полей указанного количества записей (например, первые 50 записей из отфильтрованного и отсортированного списка).

Данные, полученные из множества записей, передаются в специальный параметр типа “Коллекция”. Параметры типа “коллекция” содержат вложенные параметры ([Рис. 3](#)), которые представляют собой колонки вычитанных записей. Например, если вычитать имена, даты рождения и должности всех контактов определенного контрагента, то результирующий параметр коллекции будет содержать вложенные параметры “ФИО”, “Дата рождения” и “Название должности”.

Рис. 3 — Пример параметра коллекции

ФИО	Должность	Дата рождения
Ольга Варенская	Менеджер по работе с клиентами	12.04.1991
Андрей Валевский	Менеджер по продажам	24.10.1985
Александра Ульяненко	Системный администратор	05.12.1989

## Примеры использования

- ## Элемент процесса [Добавить данные]

Для автоматического добавления записей в объект, например, в раздел, справочник или на деталь,

используйте элемент [ *Добавить данные* ] ([Рис. 1](#)).

Рис. 1 — Элемент процесса [ *Добавить данные* ]

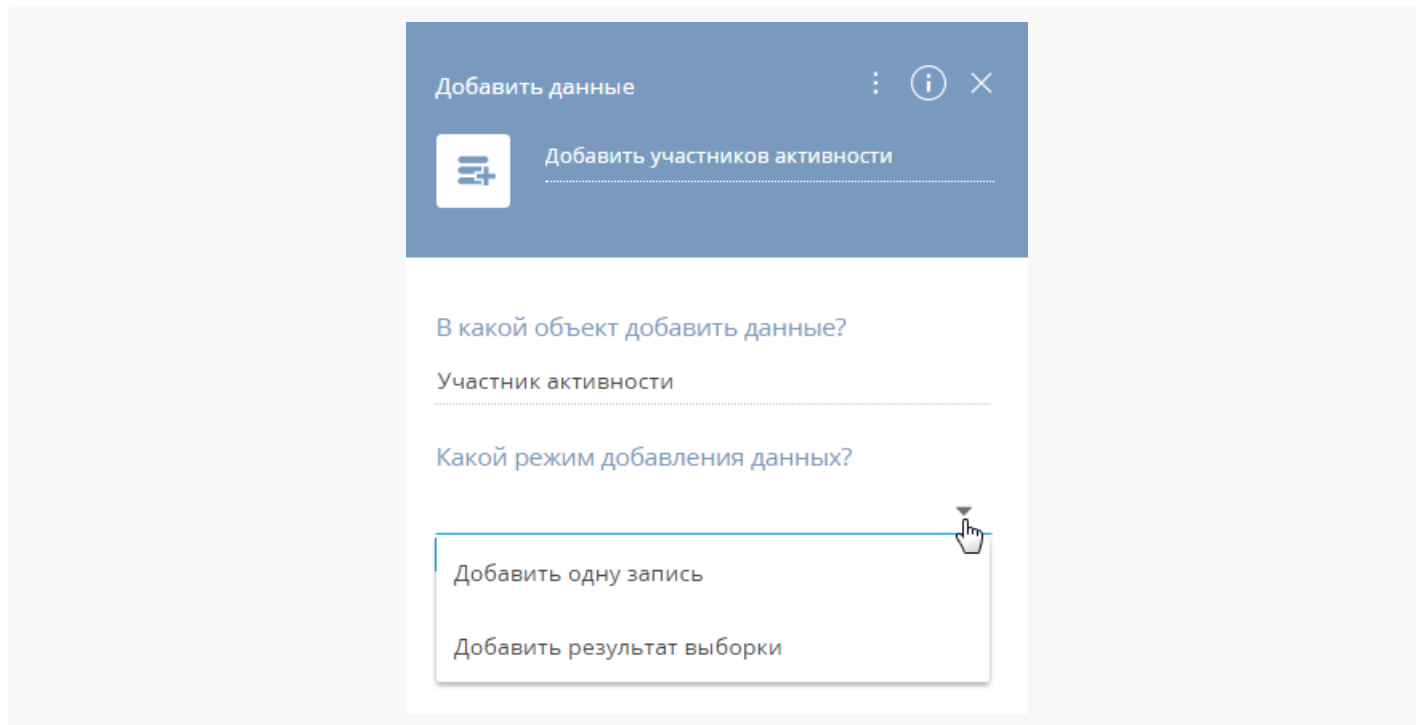


Записи добавляются в фоновом режиме, без открытия страницы записи. Это удобно, если необходимо автоматически наполнить данными раздел или деталь без пользовательского интерфейса (например, без страницы редактирования записи).

Элемент [ *Добавить данные* ] добавит данные в любой объект, независимо от прав доступа пользователя, который запускает процесс.

Есть два режима добавления данных ([Рис. 2](#)).

Рис. 2 — Выбор режима для элемента [ *Добавить данные* ]



## Добавить одну запись

Элемент добавляет новую запись с указанными значениями полей в выбранный объект (в раздел, справочник или на деталь). Например, можно добавить текущего пользователя в качестве участника встречи, в результате которой был запущен процесс. Элемент возвращает только идентификатор добавленной записи. Для получения остальных данных добавленной записи используйте элемент [ *Читать данные* ] после элемента [ *Добавить данные* ].

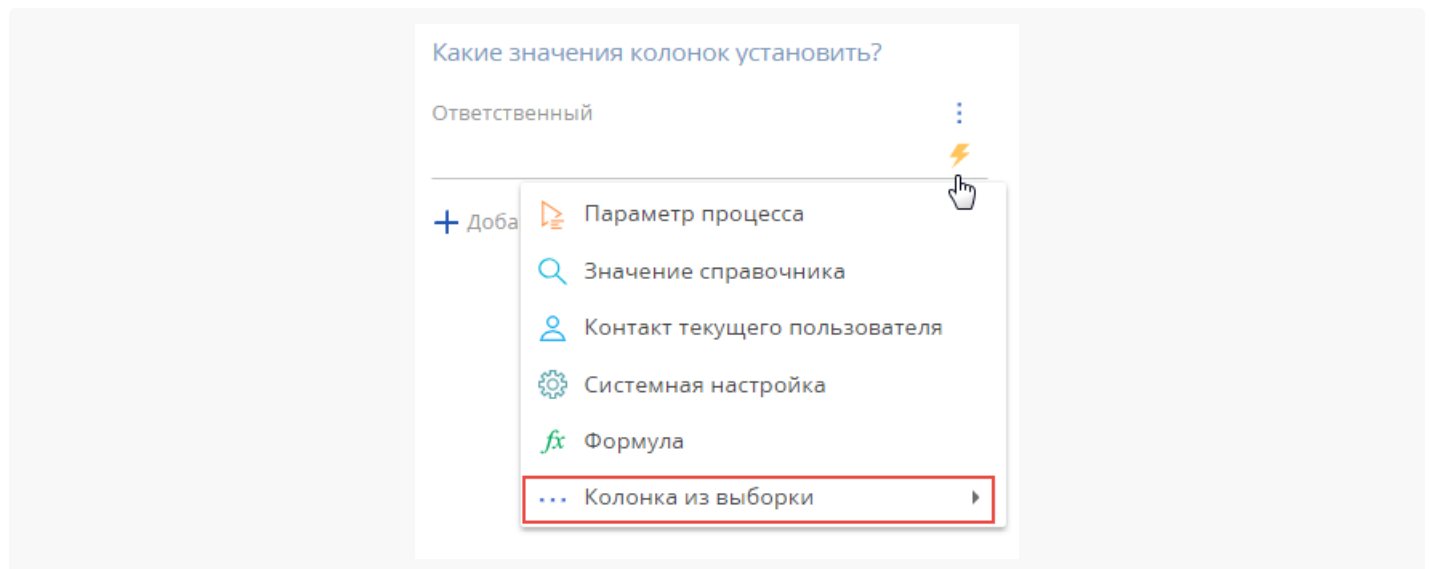
## Добавить результат выборки

Элемент добавляет несколько записей в выбранный объект (в раздел, справочник или на деталь) на

основании отфильтрованного списка записей (“выборки”) объекта — источника выбранных записей. Количество и содержимое записей зависит от данных в выборке:

- По каждой записи из выборки в указанном объекте будет создана отдельная запись.
- Поля новых записей можно заполнить значениями из соответствующих записей в выборке. Для этого используйте команду [ *Колонка из выборки* ] при указании, какими значениями должны быть заполнены колонки добавляемых записей (Рис. 3).

Рис. 3 — Сопоставление (маппинг) значений полей новых записей с данными из выборки



## Примеры использования

- [Добавить одну запись](#)
- [Добавить несколько записей](#)

# Элемент процесса [Изменить данные]

ПРОДУКТЫ: **ВСЕ ПРОДУКТЫ**

Если в ходе выполнения процесса необходимо автоматически изменить значения в некоторых полях записей, то используйте элемент [ *Изменить данные* ] (Рис. 1).

Рис. 1 — Элемент [ *Изменить данные* ]



С помощью элемента можно внести изменения в любое количество записей, но все выполненные изменения будут одинаковыми. Например, вы можете использовать элемент [ *Изменить данные* ] для изменения состояния всех активностей на “Отменена”. Для выполнения разных изменений в разных записях используйте несколько элементов [ *Изменить данные* ].



Записи для внесения изменений выбираются с помощью стандартного фильтра. Например, вы можете внести изменения во все активности, срок выполнения которых был вчера. В выбранных записях можно изменить значения одного или нескольких полей.

Все изменения выполняются в фоновом режиме, без открытия страниц редактирования записей. Используйте элемент [ *Изменить данные* ] для внесения любых изменений, которые не требуют участия пользователя. Также элемент полезен, если необходимо отредактировать записи в объектах, которые не имеют пользовательского интерфейса (например, если нет страницы редактирования записи).

Элемент может изменить любую запись, независимо от прав доступа пользователя, который запускает процесс.

## Примеры использования

- [Изменить несколько записей, которые соответствуют условию](#)
- [Изменить определенную запись](#)

# Элемент процесса [Удалить данные]

ПРОДУКТЫ: **ВСЕ ПРОДУКТЫ**

Элемент [ *Удалить данные* ] ([Рис. 1](#)) удаляет выбранные записи.

Рис. 1 — Элемент [ *Удалить данные* ]



Записи для удаления выбираются с помощью стандартного фильтра. Например, вы можете удалить все запланированные активности в состоянии “Отменена” или только одну определенную активность, созданную ранее по процессу.

После выполнения элемента записи удаляются автоматически, без участия пользователя. Элемент [ *Удалить данные* ] удалит запись в любом объекте, независимо от прав доступа пользователя, который запускает процесс.

## Примеры использования

- [Удалить все записи, которые соответствуют условию](#)
- [Удалить определенную запись](#)

# Элемент процесса [Формула]

ПРОДУКТЫ: **ВСЕ ПРОДУКТЫ**

Используя элемент процесса [ *Формула* ], вы можете производить автоматические вычисления в ходе

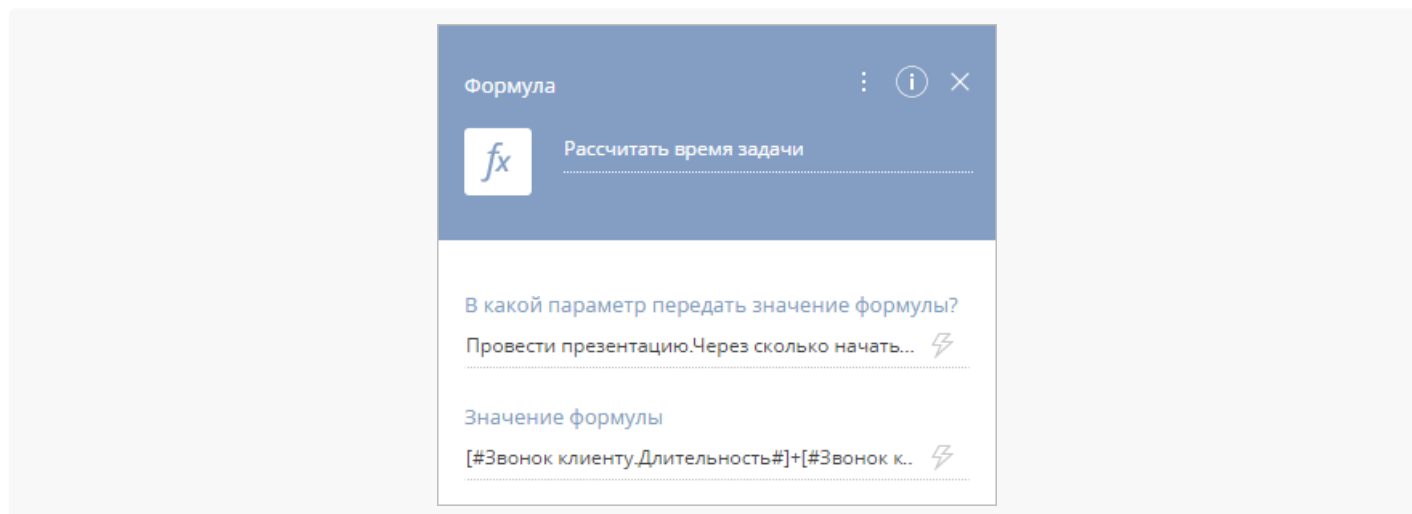
выполнения процесса и использовать результаты вычислений в качестве значений параметров других элементов.

Например, при помощи элемента [ *Формула* ] вы можете вычислить время, через которое должна начинаться следующая задача, основываясь на фактической продолжительности предшествующих задач. Больше об использовании формул читайте в статье "[Формулы в бизнес-процессах](#)".

Элемент [ *Формула* ] также используется для определения условий перехода по [условным потокам](#).

На панели настройки элемента [ *Формула* ] укажите значения параметров ([Рис. 1](#)).

Рис. 1 — Панель настройки элемента [ *Формула* ]



Поля заполняются при помощи [параметров процесса](#):

1. В верхней части панели настройки элемента введите подпись элемента. Указанное значение отобразится на диаграмме процесса.
2. [ *В какой параметр передать значение формулы?* ] — укажите параметр, значение которого будет содержать результат вычисления. Например, если формула используется для расчета длительности выполнения задачи, укажите параметр [ *Длительность* ] элемента процесса [ *Выполнить задачу* ]. Могут быть указаны параметры, значения которых представляют собой числа, дату и/или время, а также строковые и логические параметры.

**Важно.** Тип данных, которые возвращает формула, зависит от типа данных параметра, указанного в поле [ *Установить значение в* ].

Вы можете указать значения параметров, которые уже существуют в процессе на момент добавления элемента [ *Формула* ]. Каждый элемент [ *Формула* ] рассчитывает значение одного конкретного параметра.

3. [ *Значение формулы* ] — сформируйте формулу, результат которой необходимо передать в параметр. Формула строится в текстовом поле окна определения значения параметра. Для построения используются вкладки [ *Элементы процесса* ], [ *Параметры процесса* ], [ *Системные настройки* ], [ *Выбор из справочника* ], [ *Системные переменные* ], [ *Функции* ], [ *Дата и время* ].

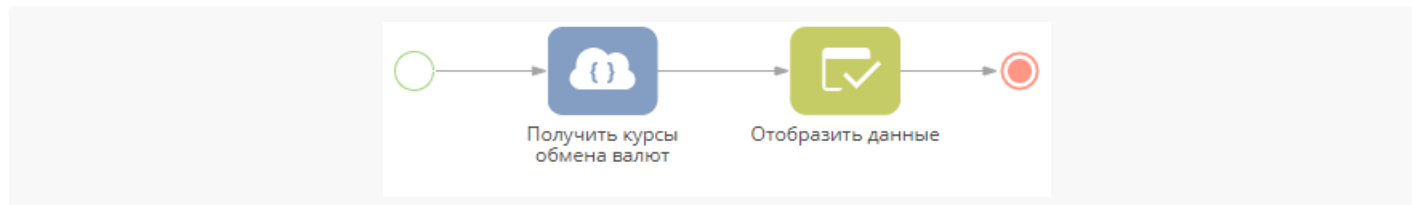
## Элемент процесса [Вызвать веб-сервис]

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Элемент процесса [ *Вызвать веб-сервис* ] используется для вызова методов веб-сервиса и обработки ответа в рамках бизнес-процесса. Перед использованием элемента необходимо настроить [интеграцию веб-сервисов](#) в разделе [ *Веб-сервисы* ] дизайнера системы.

Функциональность элемента [ *Вызвать веб-сервис* ] зависит от веб-сервиса, с которым была настроена интеграция. Например, элемент можно использовать для получения последних данных о курсе валют ([Рис. 1](#)).

Рис. 1 — Вызов веб-сервиса в рамках бизнес-процесса и отображение результатов



**Важно.** Детальные инструкции по настройке интеграции с веб-сервисом "<http://fixer.io/>" приведены в отдельной статье. [Подробнее >>>](#)

При активации входящего потока элемент [ *Вызвать веб-сервис* ]:

1. Выполняет вызов необходимого метода веб-сервиса с помощью специальных параметров вызова.
2. Ожидает отклика веб-сервиса согласно настроек интеграции.
3. Записывает результаты в свои исходящие параметры.
4. Активирует исходящие потоки.

## Настроить параметры элемента [ *Вызвать веб-сервис* ]

Для интеграции веб-сервиса в бизнес-процесс добавьте элемент [ *Вызвать веб-сервис* ] в рабочую область дизайнера процессов, выберите веб-сервис и метод, который необходимо вызвать, и укажите значения параметров на панели настройки элемента ([Рис. 2](#)):

Рис. 2 — Панель настройки элемента [ *Вызвать веб-сервис* ]



на <https://fixer.io/>.

- [ *Конечная точка* ] — конечная точка веб-сервиса, которая будет добавлена сразу после URI и перед параметрами запроса. В примере для <http://fixer.io/> может быть либо “latest” — для запроса последних доступных курсов, либо конкретная дата в текстовом формате — для запроса обменных курсов на эту дату, например “2018-01-03”. Все параметры, кроме параметров типа “Параметр запроса”, обязательны для заполнения. Если поле останется незаполненным, параметры запроса все еще будут частью запроса, но не будут иметь значений. Например: “?параметр-со-значением-1=значение-1&параметр-без-значения-1&параметр-без-значения-2”.

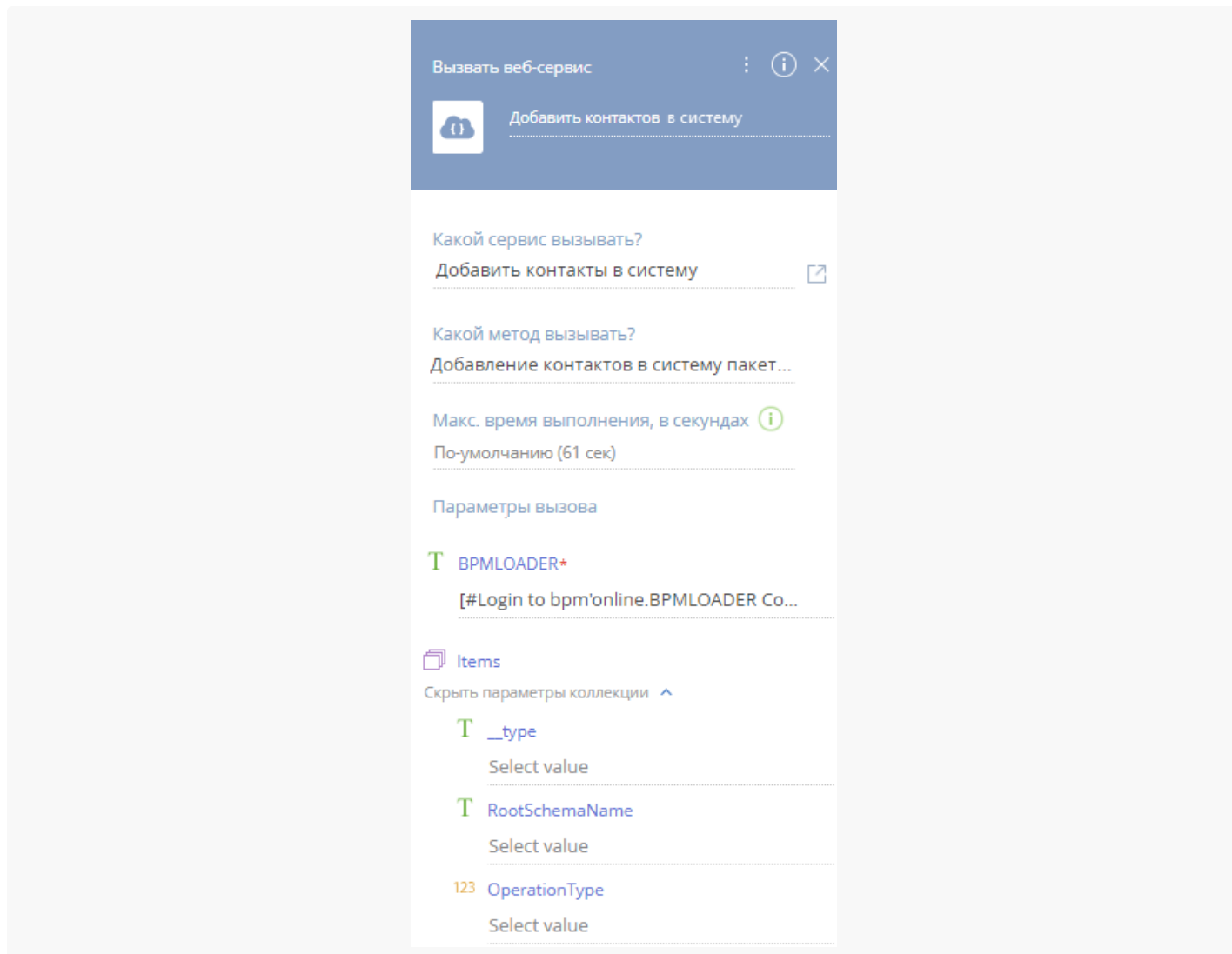
**На заметку.** Можно ввести значения параметра запроса вручную или привязать их к другим параметрам процесса. Работа со значениями параметров рассмотрена в отдельной статье.

[Подробнее >>>](#)

## Передать параметры вызова типа “коллекция”

Если параметр вызова является коллекцией (установлен признак [ *Является массивом* ] в свойствах параметра на странице веб-сервиса), то список вложенных параметров отображается под названием коллекции на панели настройки элемента ([Рис. 3](#)). Так, сервис “пакетных запросов” Creatio позволяет добавлять несколько записей (например, контактов) в систему. Для этого сервису потребуются данные для заполнения полей каждой добавляемой записи (например, ФИО и тип контакта). Нужные значения можно передать в качестве параметра вызова типа “коллекция”, где “ФИО” и “Тип” будут вложенными параметрами, а каждый элемент коллекции будет содержать данные для отдельной записи контакта

Рис. 3 — Пример параметров вызова, один из которых является коллекцией



Значения параметров коллекции элемента [ *Вызвать веб-сервис* ] могут быть сопоставлены с вложенными параметрами другой коллекции элемента [ *Читать данные* ] или [ *Вызвать веб-сервис* ] ([Рис. 4](#), [Рис. 5](#)).

Рис. 4 — Сопоставление значений параметров коллекции

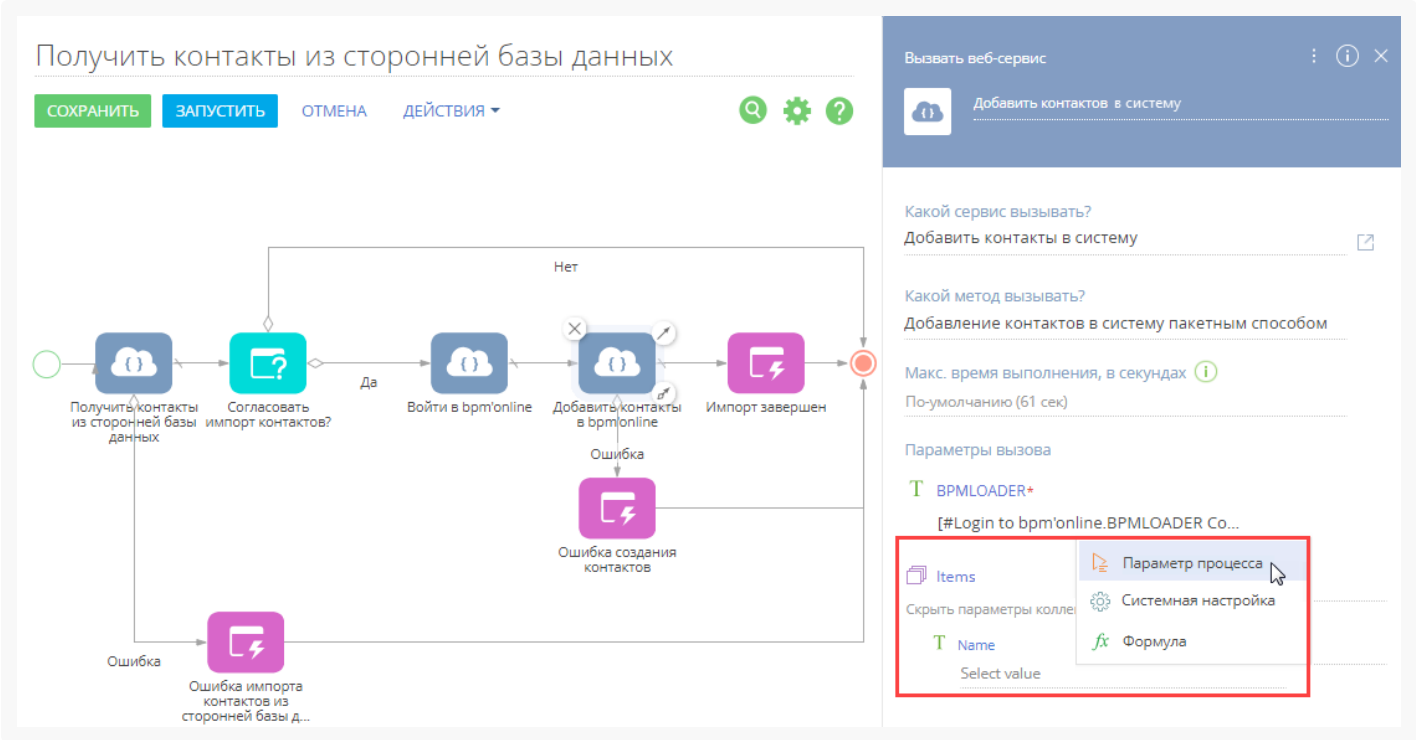
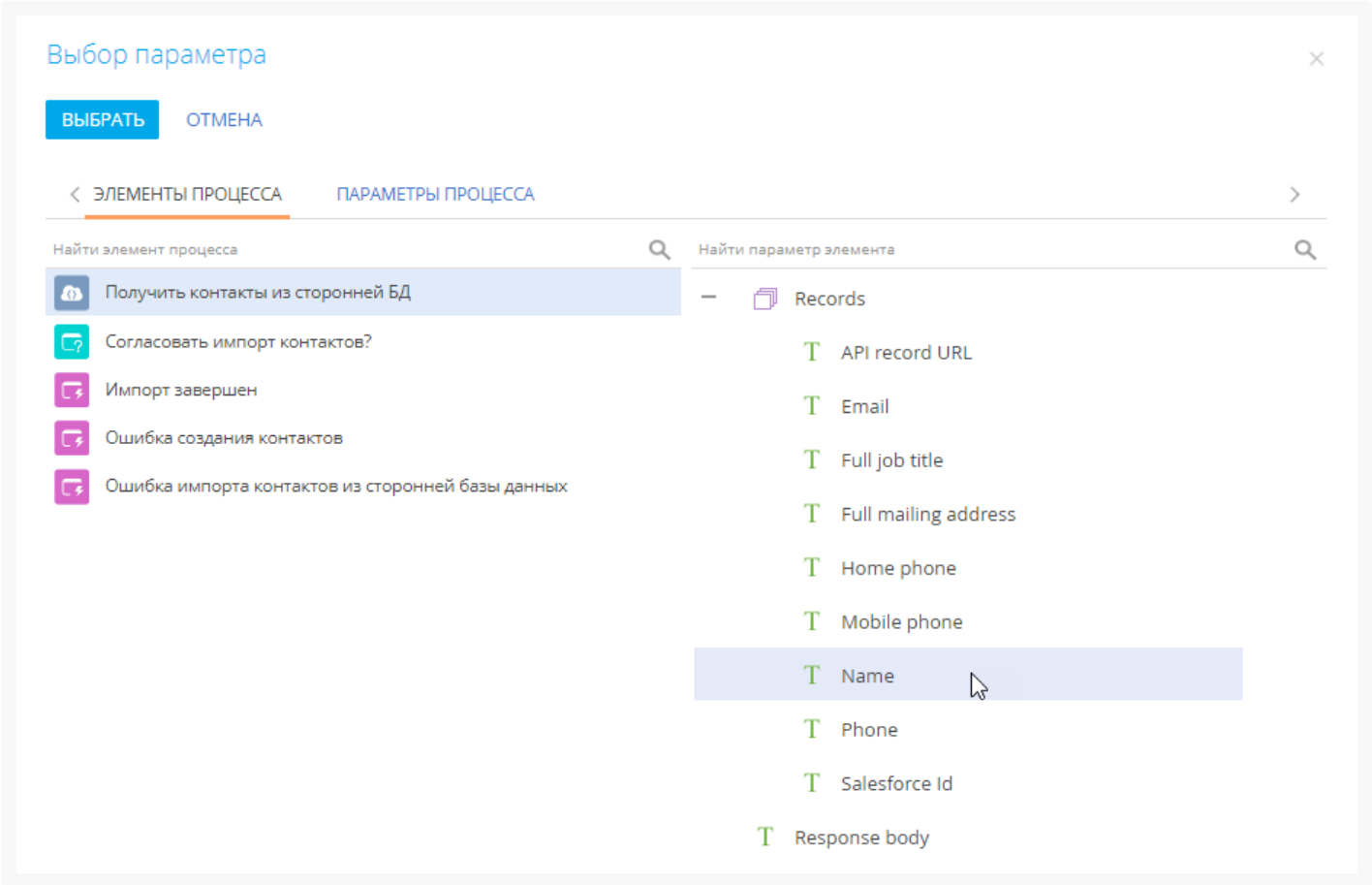


Рис. 5 — Выбор вложенного параметра другой коллекции для сопоставления



Параметры коллекции элемента процесса могут быть сопоставлены с параметром процесса типа “Коллекция значений” (Рис. 6).

Рис. 6 — Добавление параметра процесса

Процесс

Получить контакты из сторонней базы данных

НАСТРОЙКИ ПАРАМЕТРЫ МЕТОДЫ

ДОБАВИТЬ ПАРАМЕТР ▾

Название\*  
Parameter 1

Код\*  
ProcessSchemaParameter1

Тип данных\*  
Коллекция значений

Значение  
[#Get contacts form Salesforce.Records#]

СОХРАНИТЬ ОТМЕНА

## Проверить ошибки вызова веб-сервисов

После активации элемент [ *Вызвать веб-сервис* ] выполняет вызов веб-сервиса, получает и обрабатывает отклик и заполняет исходящие параметры. У каждого элемента [ *Вызвать веб-сервис* ] есть два исходящих параметра для обработки ошибок — [ *Success* ] и [ *Http status code* ].

- Если вызов завершился успешно (код отклика <400), то параметру [ **Success** ] будет установлено значение **“true”**.
- Если вызов завершился с ошибкой (код отклика 400 и выше), то параметру [ **Success** ] будет установлено значение **“false”** и параметр [ **Http status code** ] будет заполнен кодом ошибки. Используйте эти параметры для обработки ошибок вызова веб-сервисов непосредственно в процессе. Например, вы можете автоматически вызвать веб-сервис позже, использовать другие значения параметров вызова или прекратить процесс.

## Обработать отклик веб-сервиса

Каждый параметр отклика, который вы настроили для метода веб-сервиса, будет добавлен в качестве дополнительного исходящего параметра в элемент [ *Вызвать веб-сервис* ], где этот метод выбран в свойстве [ *Какой метод вызывать?* ].

Для использования этих данных в вашем бизнес-процессе свяжите процесс и параметры элементов процесса с исходящими параметрами элемента [ *Вызвать веб-сервис* ].

**Пример.** При выборе веб-сервиса “Fixer” у элемента [ *Вызвать веб-сервис* ] в дополнение к параметрам обработки ошибок появятся следующие исходящие параметры: “Базовая валюта”



(параметр “строка”), “Дата” (параметр “дата”) и отдельный дробный параметр на каждый обменный курс (“RatesAUD”, “RatesEUR” и т.д.). Полученные данные можно отобразить при помощи [преднастроенной страницы](#), связав ее поля с соответствующими исходящими параметрами элемента [ *Вызвать веб-сервис* ] ([Рис. 7](#), [Рис. 8](#)). В результате поля страницы ([Рис. 9](#)) будут содержать данные, полученные в отклике веб-сервиса.

**На заметку.** Детальнее работа со значениями параметров рассмотрена в статье “[Использовать параметры](#)”.

Рис. 7 — Настройка параметров преднастроенной страницы

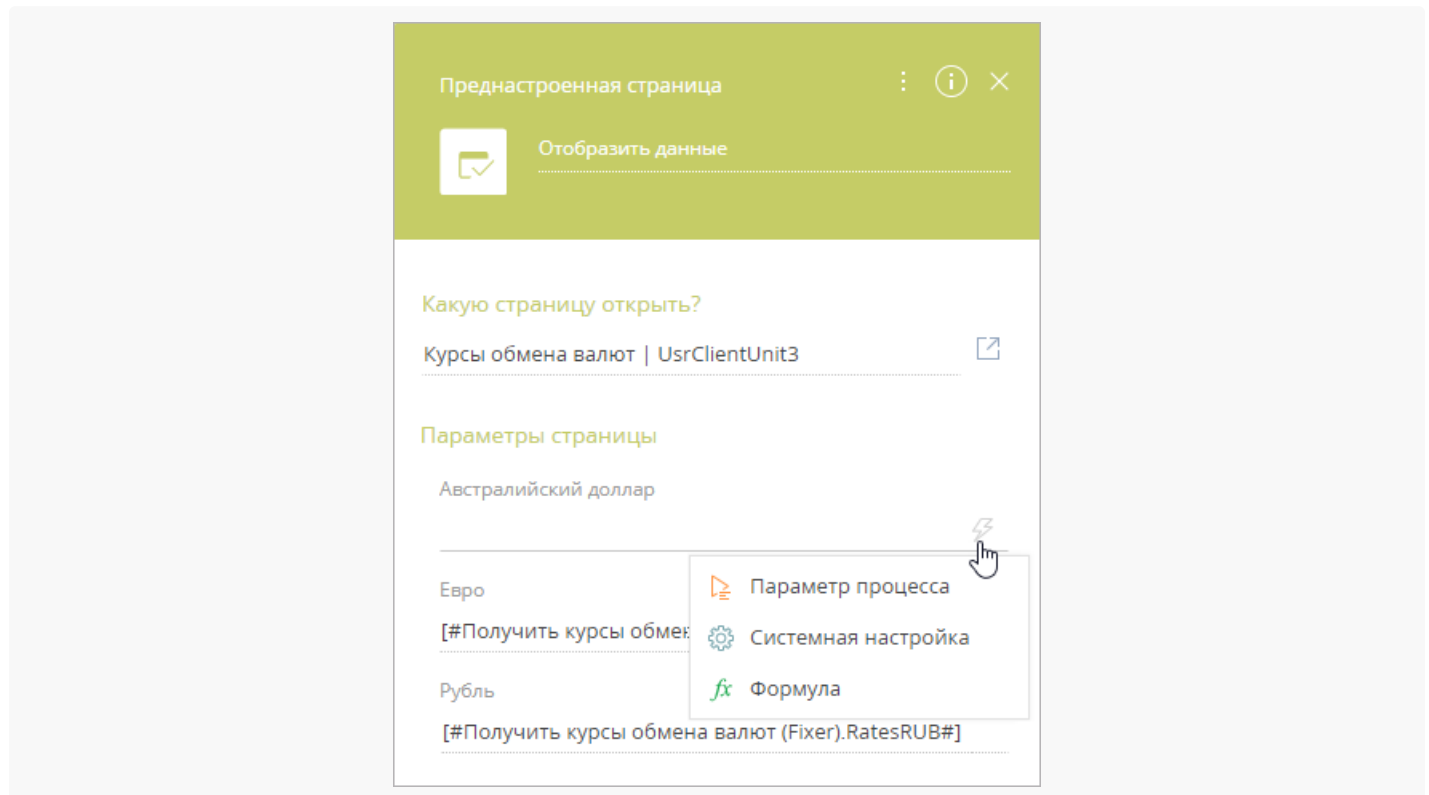


Рис. 8 — Связывание параметров преднастроенной страницы и элемента [ *Вызвать веб-сервис* ]

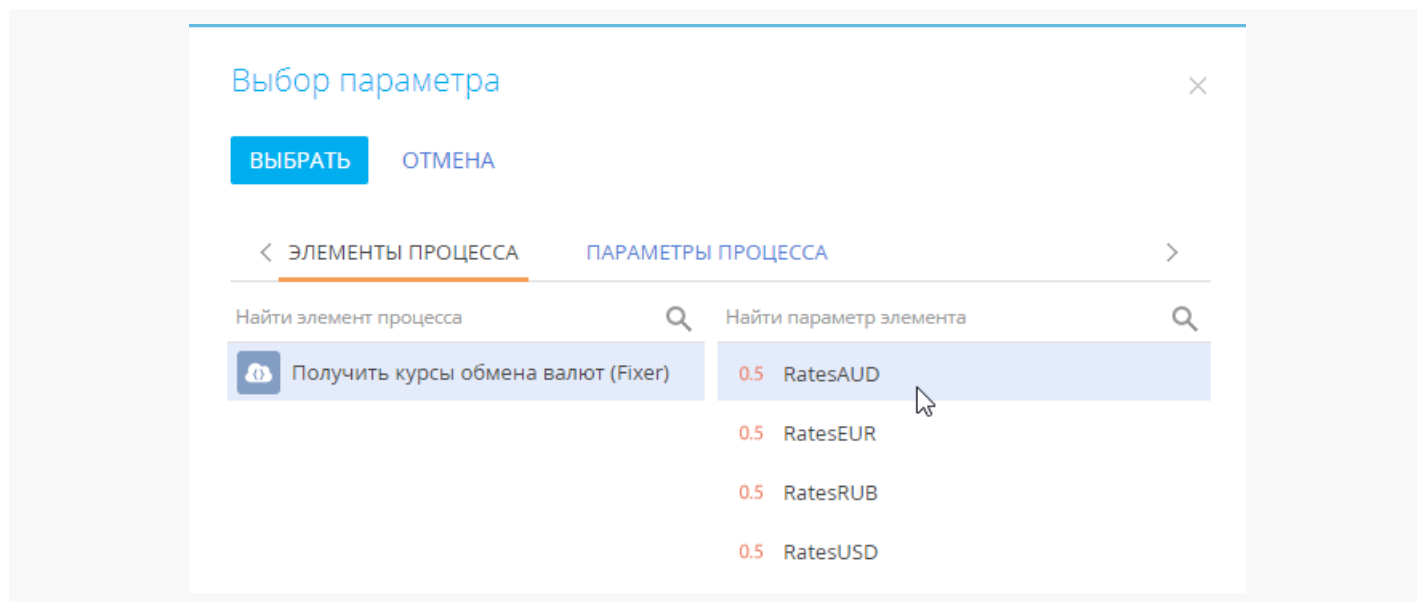
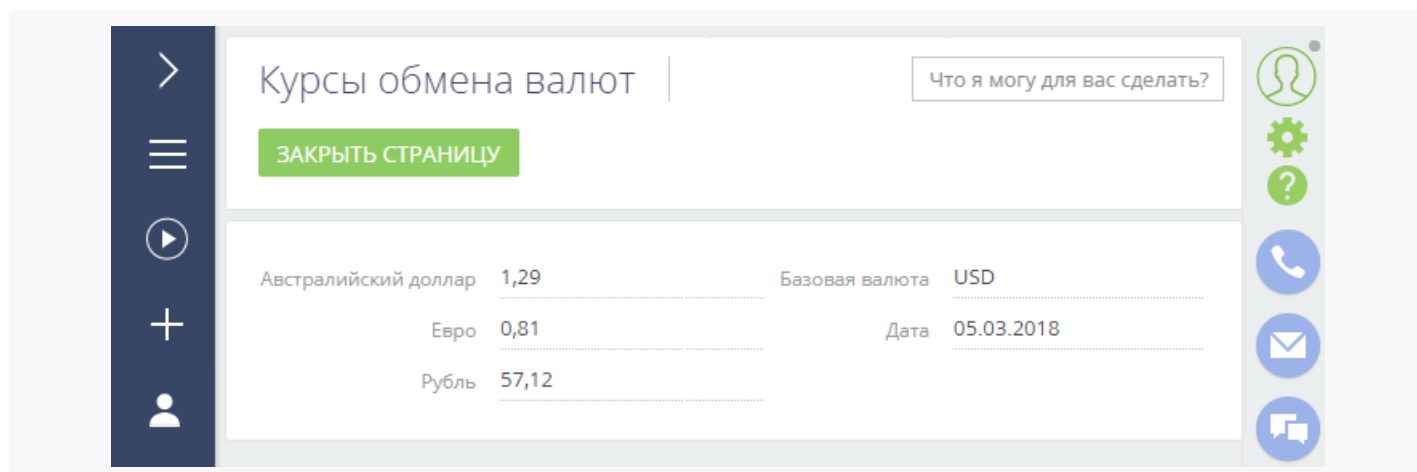


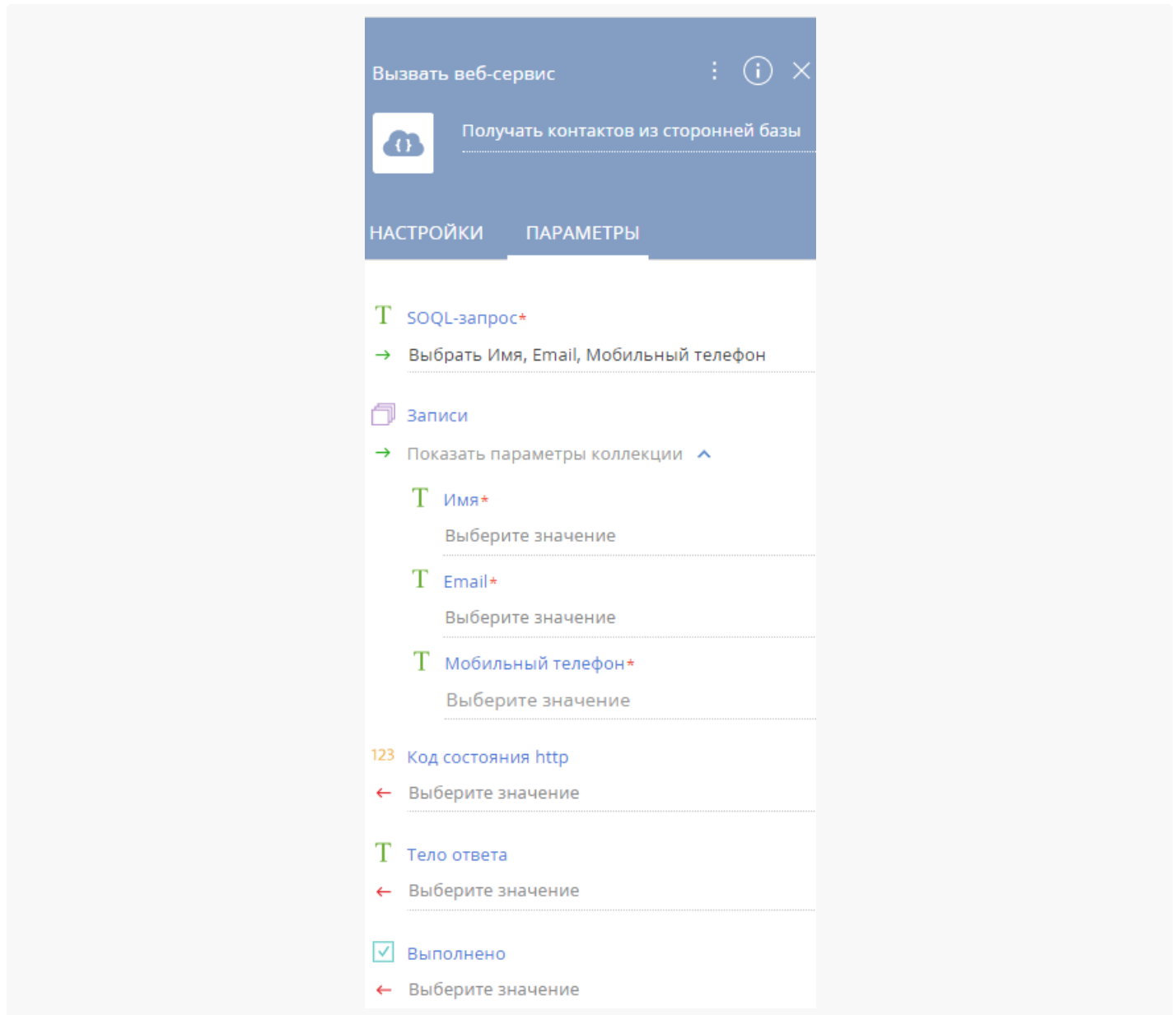
Рис. 9 — Обменные курсы на преднастроенной странице



## Настроить параметры ответа типа “коллекция”

Параметры ответа веб-сервиса могут быть коллекциями. Параметры ответа настраиваются в расширенном режиме на вкладке [ *Параметры* ] панели настройки свойств элемента [ *Вызвать веб-сервис* ] (Рис. 10). Вложенные параметры коллекции отображаются под названием коллекции.

Рис. 10 — Коллекция в параметрах вызова



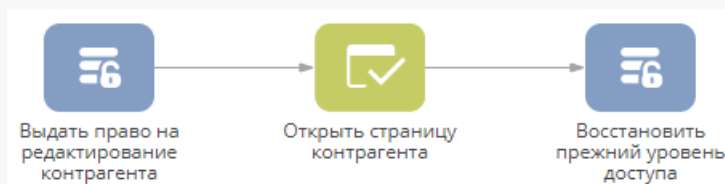
Значения параметров коллекции элемента [ *Вызвать веб-сервис* ] могут быть сопоставлены с вложенными параметрами другой коллекции элемента [ *Читать данные* ] или [ *Вызвать веб-сервис* ] ([Рис. 4](#), [Рис. 5](#)).

## Элемент процесса [Изменить права доступа]

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Элемент [ *Изменить права доступа* ] (Рис. 1) предназначен для установки или отмены прав доступа на записи Creatio.

Рис. 1 — Элементы [ *Изменить права доступа* ] на диаграмме процесса



С помощью этого элемента можно автоматизировать изменение прав доступа сотрудников при изменении должности или смене ответственных по контрагентам, контактам, продажам и т. д.

Данный элемент также используется для установки прав доступа, необходимых для выполнения пользователем действий по процессу. Например, перед выполнением действия [ *Открыть страницу редактирования* ], пользователю нужно выдать доступ на чтение записи, страница которой будет открыта.

**Важно.** Если у пользователя, который должен выполнить действие по процессу, нет прав доступа, необходимых для выполнения действия, то это действие выполнено не будет. Соответственно, если выполнение данного действия необходимо для завершения процесса, то процесс не сможет быть завершен.

Права могут быть добавлены или ограничены для:

- [ *Роли* ] — удаление прав доступа для выбранного элемента организационной структуры.
- [ *Сотруднику* ] — удаление прав доступа для определенного пользователя. Для корректной работы элемента укажите в поле контакт пользователя, которому необходимо ограничить права.
- [ *Для выборки сотрудников* ] — удаление прав доступа для всех пользователей, которые соответствуют условиям фильтра.

При **активации** элемент [ *Изменить права доступа* ] использует фильтр для получения списка записей указанного объекта, после чего добавляет или удаляет права на эти записи для указанных пользователей и ролей.

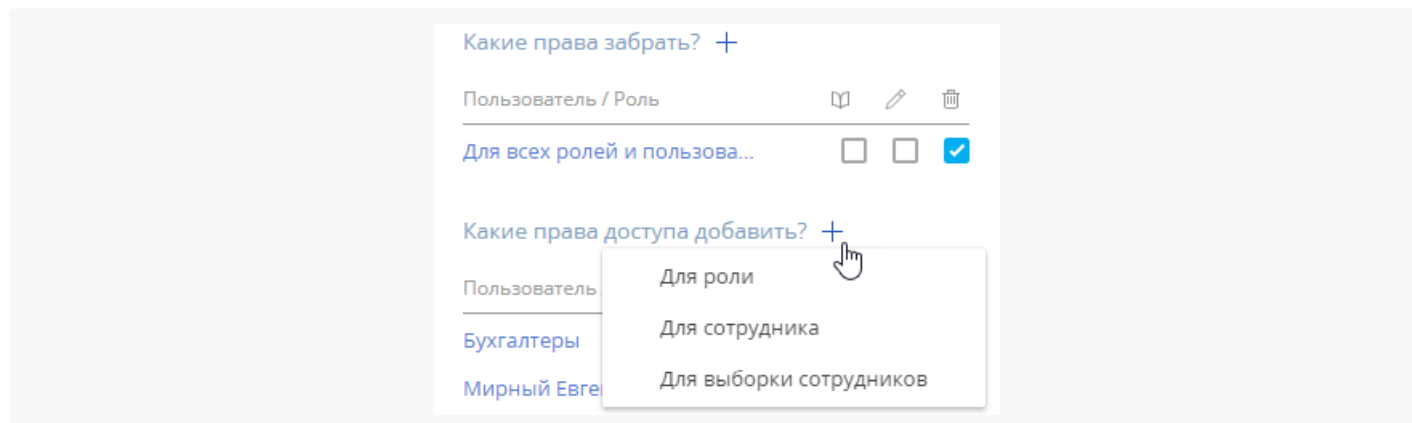
После **установки** прав доступа на записи элемент завершает выполнение и активирует свои исходящие потоки.

## Настроить свойства элемента [ *Изменить права доступа* ]




<b>Для записей какого объекта изменить права доступа?</b>	Объект, который содержит записи, на которые необходимо изменить права доступа. Например, чтобы изменить права доступа на определенные активности, выберите “Активность”, чтобы изменить права доступа к контактам — выберите “Контакт”, и т. д.
<b>Изменить доступ на все записи, соответствующие условию</b>	Установите фильтр для выбора записей, на которые будут изменены права доступа. Чтобы изменить права доступа к конкретной записи, настройте фильтр по колонке [ <i>Id</i> ] (уникальный идентификатор).
<b>Какие права забрать?</b>	Укажите права доступа, которые необходимо <b>забрать</b> при выполнении элемента. Нажмите [ + ], чтобы добавить новое право доступа. При выполнении элемента <b>указанные права доступа будут удалены</b> для указанных пользователей/ролей.
<b>Какие права доступа добавить?</b>	Укажите права доступа, которые необходимо <b>выдать</b> при выполнении элемента. Нажмите [ + ] (Рис. 2), чтобы добавить новое право доступа. При выполнении элемента <b>указанные права доступа будут добавлены</b> для указанных пользователей/ролей.

“Право доступа на запись” в Creatio — это определенный уровень доступа пользователя или роли на выполнение определенной операции с записью. Права доступа могут быть установлены как для отдельных пользователей (“**сотрудников**”), так и для групп пользователей (“**ролей**”).

Рис. 2 — Добавление прав доступа

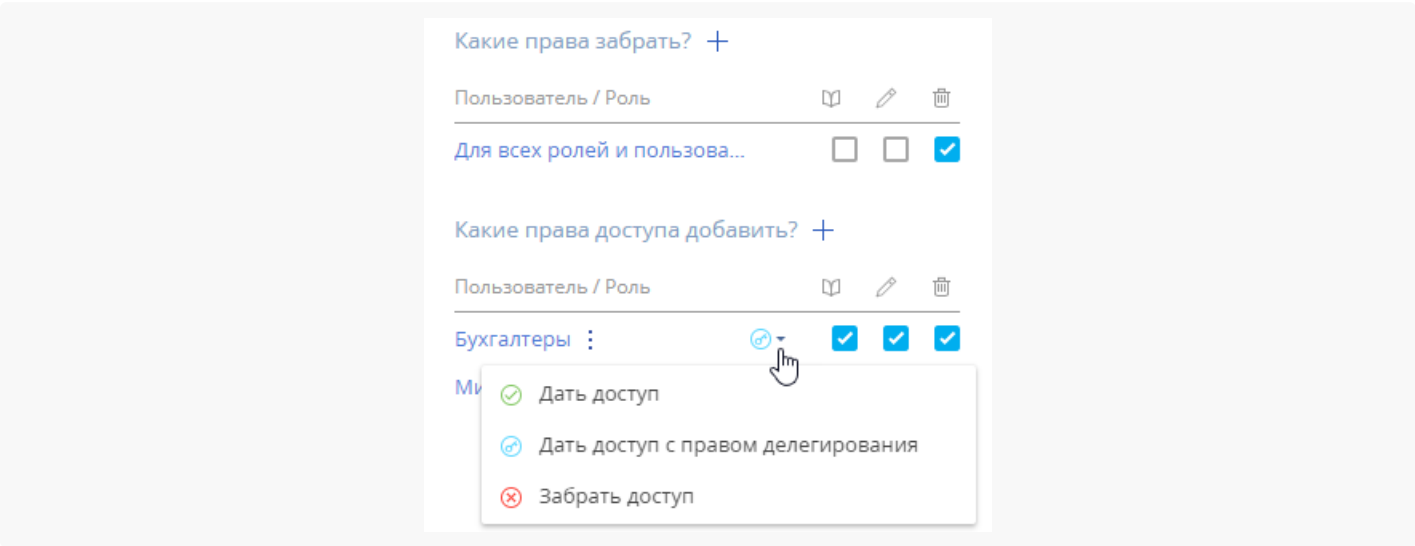




Права доступа устанавливаются на следующие **операции**:

 <b>Чтение</b>	Позволяет просматривать записи без возможности их редактировать и удалять. Без права на чтение пользователь не увидит запись в реестре, а также значения полей этой записи в других реестрах и на страницах. То есть для того, чтобы открыть страницу записи, нужно право на чтение.
 <b>Редактирование</b>	Позволяет редактировать значения полей записи. При этом у пользователя должно быть право на чтение, без которого запись недоступна для просмотра. Так, чтобы изменить значение любого поля на странице записи, нужны права на чтение и редактирование.
 <b>Удалить</b>	Позволяет удалять записи. При этом у пользователя должно быть право на чтение, без которого запись недоступна для просмотра. Так, чтобы удалить запись, нужно право на чтение и удаление.

При добавлении прав доступа на операции можно установить **уровень доступа** (Рис. 3):

Рис. 3 — Установка уровня доступа



-  [ *Дать доступ с правом делегирования* ] — пользователь может выполнять операции с записью самостоятельно, а также предоставлять доступ на выполнение операций другим пользователям, используя стандартную функциональность по управлению доступом к записям.
-  [ *Забрать доступ* ] — пользователь не сможет выполнять указанные операции.

## Примеры использования

- [Изменить права доступа в рамках бизнес-процесса](#)

# Элемент процесса [Прогнозировать данные]

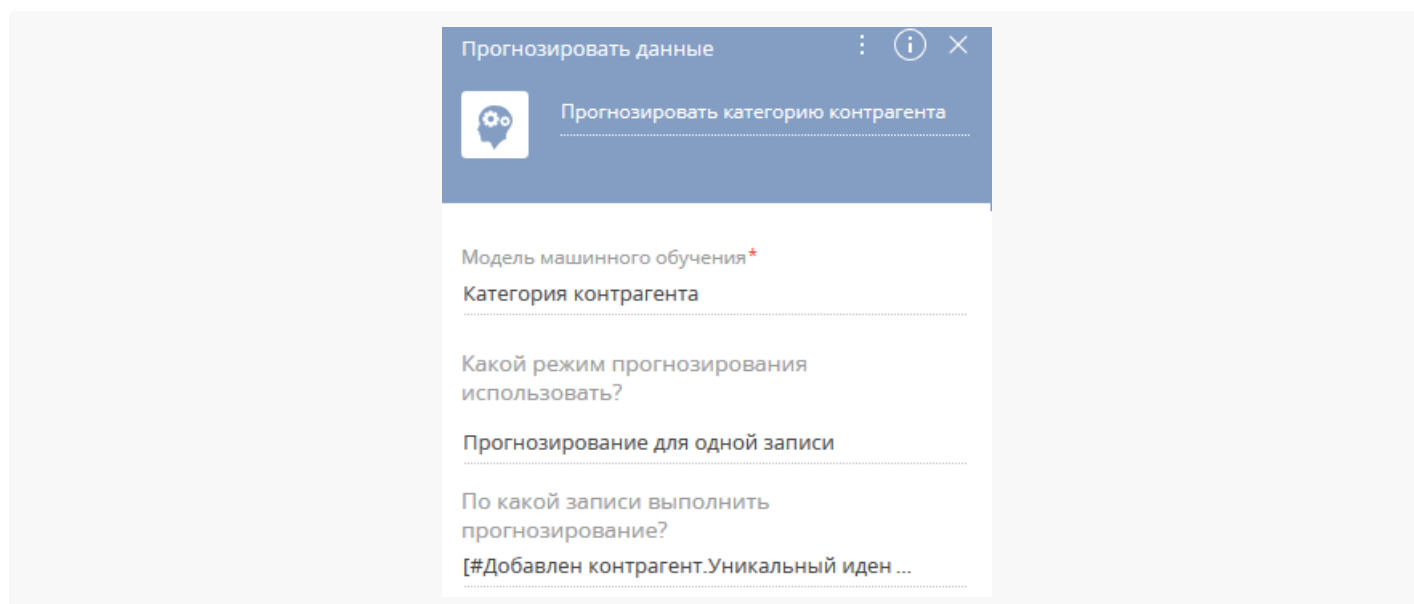
ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Элемент [ *Прогнозировать данные* ] использует обученные модели машинного обучения для прогнозирования данных на страницах записей при выполнении бизнес-процесса.

При настройке элемента необходимо указать заранее созданную модель машинного обучения, которая будет использоваться для прогнозирования данных, а также определить записи для прогнозирования.

**На заметку.** Перед использованием в бизнес-процессах модель необходимо обучить. Модели, не прошедшие обучение, недоступны для выбора в поле [ *Модель машинного обучения* ] элемента [ *Прогнозировать данные* ]. Подробнее: [Инструменты ИИ](#).

Рис. 1 — Настройка элемента [ *Прогнозировать данные* ]



## Примеры использования

- [Настроить бизнес-процесс с прогнозированием значения справочного поля](#)
- [Настроить бизнес-процесс с рекомендательным прогнозированием](#)

## Элемент процесса [Задание-сценарий]

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Элемент процесса [ *Задание-сценарий* ] является системным действием, которое выполняет программный код C# сценария и обеспечивает его взаимодействие с другими элементами и данными бизнес-процесса. При помощи элемента реализуется расширенная логика, не предусмотренная базовыми элементами Creatio, например:

- выполнение сложных математических расчетов;

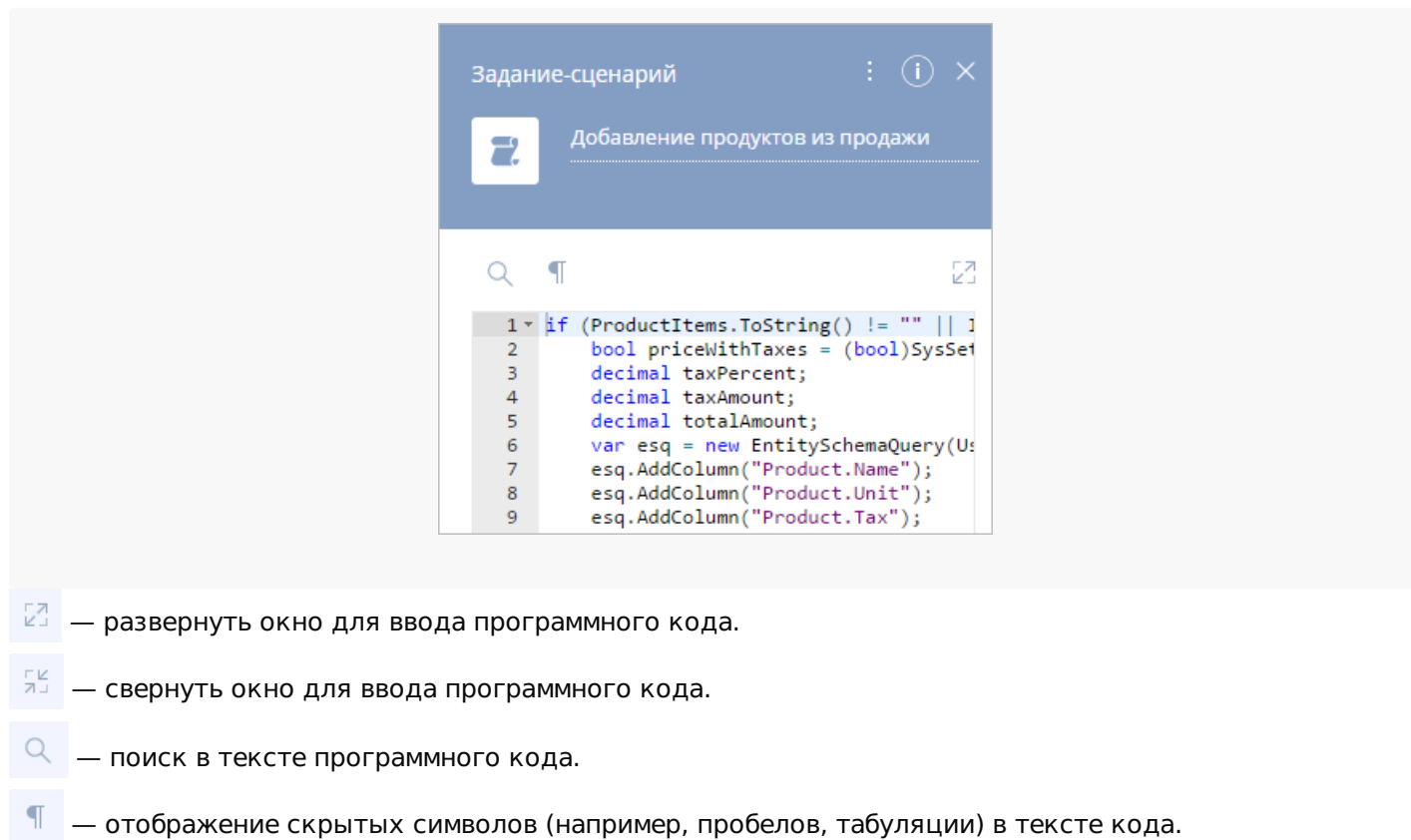
- выполнение массовой обработки записей;
- выполнение циклов;
- выполнение задач, которые затруднительно реализовать с помощью других элементов системы;
- замена нескольких последовательных элементов [ *Формула* ].

Работая с элементом [ *Задание-сценарий* ], придерживайтесь следующих рекомендаций.

- Соблюдайте структуру кода. После сохранения протестируйте работу элемента, выполнив процесс. Корректность и скорость отработки элемента зависят от качества кода и квалификации разработчика.
- Выделяйте комментариями основные блоки и задачи, выполняемые в задании-сценарии, для пояснения функций и предназначения той или иной части кода, ее влияния в процессе. Это позволит пользователю лучше понимать, что происходит в процессе, а разработчику комментарии помогут быстрее разобраться в ранее созданном процессе.
- При использовании сторонних схем указывайте точные названия таких схем. Руководствуйтесь данными рекомендациями при написании сторонних схем.
- Избегайте временных решений, поскольку нестандартный код ухудшает возможности поддержки кода. При выявлении ошибок в процессах контролируйте корректное исправление ошибок.

Чтобы редактировать код сценария, дважды щелкните по элементу на диаграмме. На панели настройки элемента откроется окно для ввода и редактирования программного кода (Рис. 1).

Рис. 1 — Вкладка для редактирования кода задания-сценария



**На заметку.** Платформа Creatio позволяет выполнять процессы без необходимости их публикации.



Однако использование методов и элементов [ *Задание-сценарий* ] требует публикации схемы. Подробнее о вызове методов процесса читайте в статьях: [“Настроить интеграцию с веб-сервисом SOAP”](#) и [“Настроить интеграцию с веб-сервисом REST”](#).

## Методы Get и Set

Все создаваемые бизнес-процессы в Creatio являются **интерпретируемыми**. Для обращения к значениям параметра процесса следует использовать методы Get и Set (Рис. 2).

Метод **Get** возвращает значение параметра элемента или процесса.

Сигнатура метода:

```
Get<T>(string path)
```

T — тип значения параметра;

path — строка, определяющая путь к параметру или свойству. Путь формируется согласно правилам:

- “имя параметра”,
- “имя свойства”,
- “имя элемента.имя параметра”,
- “имя элемента.имя свойства”.

Метод **Set** указывает значение параметру элемента или процесса.

Сигнатура метода:

```
Set(string path, T value)
```

value — указываемое значение,

path — строка, определяющая путь к параметру или свойству. Путь формируется согласно правилам, описанным выше для метода Get.

"T" — универсальный параметр, который принимает значение типа соответствующего параметра в терминах C#. Соответствие типов параметров для Creatio и C# приведено в таблице ниже.

## Соответствие типов параметров Creatio и C#

Тип параметра Creatio	Тип параметра C#
Целое число	int
Дробное число (0.00000001)	decimal
Дробное число (0.0001)	
Дробное число (0.001)	
Дробное число (0.01)	
Дробное число (0.1)	
Деньги	
Дата/Время	DateTime
Дата	
Время	
Уникальный идентификатор	Guid
Справочник	
Логическое	bool
Строка (50 символов)	string
Строка (250 символов)	
Строка (500 символов)	
Строка неограниченной длины	
Не локализуемая строка	
Коллекция значений	ICollection и любые классы, которые реализуют эти интерфейсы
Коллекция записей	ICompositeObjectList<ICompositeObject> и любые классы, которые реализуют эти интерфейсы

## Примеры работы с различными типами параметров

**Параметр типа "Целое число"**

```
int integerValue = Get<int>("IntegerParameter");
integerValue += 5;
Set<int>("IntegerParameter", integerValue);
```

**Параметры типа "Деньги" и "Дробное число" любой точности**

```
decimal decimalValue = Get<decimal>("DecimalParameter");
decimalValue += 5.5m;
Set<decimal>("DecimalParameter", decimalValue);
```

**Параметр типа "Уникальный идентификатор"**

```
Guid uniqueIdentifierValue = Get<Guid>("UniqueIdentifierParameter");
if (uniqueIdentifierValue != Guid.Empty) {
    uniqueIdentifierValue = Guid.Empty;
    Set<Guid>("UniqueIdentifierParameter", uniqueIdentifierValue);
}
```

**Параметры типов "Дата", "Время" и "Дата/Время"**

```
DateTime dateTimeValue = Get<DateTime>("DateTimeParameter");
dateTimeValue = dateTimeValue.AddDays(1);
Set<DateTime>("DateTimeParameter", dateTimeValue);
```

**Параметр типа "Справочник"**

```
Guid lookupValue = Get<Guid>("LookupParameter");
if (lookupValue.IsEmpty()) {
    lookupValue = (Guid)UserConnection.SystemValueManager.GetValue(UserConnection, "CurrentUserC
    Set<Guid>("LookupParameter", lookupValue);
}
```

**Параметры типа "Не локализуемая строка" и "Строка" любой длины**

```
string textValue = Get<string>("TextParameter");
textValue += " and something else";
Set<string>("TextParameter", textValue);
```

**Параметры для локализуемых строк**

```

LocalizableString localizableStringValue = Get<LocalizableString>("LocalizableStringParameter");
CultureInfo cultureRu = CultureInfo.GetCultureInfo("ru-RU");
CultureInfo cultureEn = CultureInfo.GetCultureInfo("en-US");
localizableStringValue.SetCultureValue(cultureRu, "Здравствуйте!");
localizableStringValue.SetCultureValue(cultureEn, "Hello!");
Set<LocalizableString>("LocalizableStringParameter", localizableStringValue);

```

**Параметр типа "Логический"**

```

bool booleanValue = Get<bool>("BooleanParameter");
booleanValue = !booleanValue;
Set<bool>("BooleanParameter", booleanValue);

```

**Параметр типа "Коллекция значений"**

```

ObjectList<int> numbers = ObjectList.Create(1, 2, 3, 4);
Set<ObjectList<int>>("IntegerValuesParameter", numbers);
var items = Get<ObjectList<int>>("IntegerValuesParameter");
items.Add(5);
Set<ObjectList<int>>("IntegerValuesParameter", items);

ObjectList<bool> booleanValues = ObjectList.Create(false, true, true, false);
Set<ObjectList<bool>>("BooleanValuesParameter", booleanValues);
booleanValues = Get<ObjectList<bool>>("BooleanValuesParameter");
if (booleanValues.Count == 4) {
    booleanValues.Clear();
}
Set<ObjectList<bool>>("BooleanValuesParameter", booleanValues);
ObjectList<DateTime> dateTimeValues = ObjectList.Create(new DateTime(2020, 08, 03, 13, 15, 14),
Set<ObjectList<DateTime>>("DateTimeValuesParameter", dateTimeValues);

ObjectList<Guid> guidValues = ObjectList.Create(Guid.NewGuid(), Guid.NewGuid());
Set<ObjectList<Guid>>("GuidValuesParameter", guidValues);
guidValues = Get<ObjectList<Guid>>("GuidValuesParameter");
if (!guidValues.Contains(Guid.Empty)) {
    guidValues.Add(Guid.Empty);
}
Set<ObjectList<Guid>>("GuidValuesParameter", guidValues);

ObjectList<decimal> decimalValues = ObjectList.Create(3.14m, 432434.00032m);

```

```

Set<ObjectList<decimal>>("DecimalValuesParameter", decimalValues);
decimalValues = Get<ObjectList<decimal>>("DecimalValuesParameter");
decimalValues.RemoveAt(1);
Set<ObjectList<decimal>>("DecimalValuesParameter", decimalValues);

ObjectList<string> stringValues = ObjectList.Create("string value 1", "string value 2");
Set<ObjectList<string>>("StringValuesParameter", stringValues);
stringValues = Get<ObjectList<string>>("StringValuesParameter");
stringValues.Remove("string value 1");
Set<ObjectList<string>>("StringValuesParameter", stringValues);

```

### Параметр типа "Коллекция записей"

```

var list = Get<ICompositeObjectList<ICompositeObject>>("ReadDataUserTask1.ResultCompositeObjectList");
var sb = new StringBuilder();
foreach (ICompositeObject item in list) {
    if (item.TryGetValue<string>("Name", out string value)) {
        sb.Append(value).Append( " | ");
    }
}
Set<string>("FieldsOfCompositeObjectListParameter", sb.ToString());

```

### Параметр типа "Коллекция записей", создающий новый перечень записей

```

var list = new CompositeObjectList<CompositeObject>();
var item1 = new CompositeObject();
item1["Id"] = Guid.NewGuid();
item1["Name"] = "Name1";
list.Add(item1);
var item2 = new CompositeObject();
item2["Id"] = Guid.NewGuid();
item2["Name"] = "Name2";
list.Add(item2);
Set<CompositeObjectList<CompositeObject>>("CompositeObjectListParameter", list);

```

## Элемент процесса [Привязать процесс к объекту]

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Каждый бизнес-процесс в Creatio может быть запущен множество раз, и каждый экземпляр процесса может быть связан с различными записями в системе: как теми, которые уже существуют в системе, так

и теми, которые создаются в рамках процесса.

С целью отслеживания корректности выполнения бизнес-процесса, по некоторым записям, например, контрагентам или контактам, важно сохранять список бизнес-процессов, выполнение которых связано с этими записями.

Список записей, с которыми связан экземпляр процесса, отображается в блоке [ *Связанные объекты* ] раздела [ *Журнал процессов* ].

Для создания связи между экземпляром процесса и определенной записью системы используется элемент [ *Привязать процесс к объекту* ] ([Рис. 1](#)).

Рис. 1 — Пример использования элемента [ *Привязать процесс к объекту* ]



На панели настройки элемента [ *Привязать процесс к объекту* ] укажите настройки для создания связи ([Рис. 2](#)):

Рис. 2 — Панель настройки элемента [ *Привязать процесс к объекту* ]

1. [ *Объект привязки* ] — укажите объект, с которым необходимо связать процесс, например “Контакт”. Поле заполняется при помощи [окна определения значения параметра](#).
2. [ *Запись объекта привязки* ] — укажите запись объекта, с которой должен быть связан процесс. Поле заполняется при помощи [окна определения значения параметра](#). Заполняя параметры элемента, вы можете указывать конкретные значения либо определять значения динамически на основании информации в других элементах процесса.

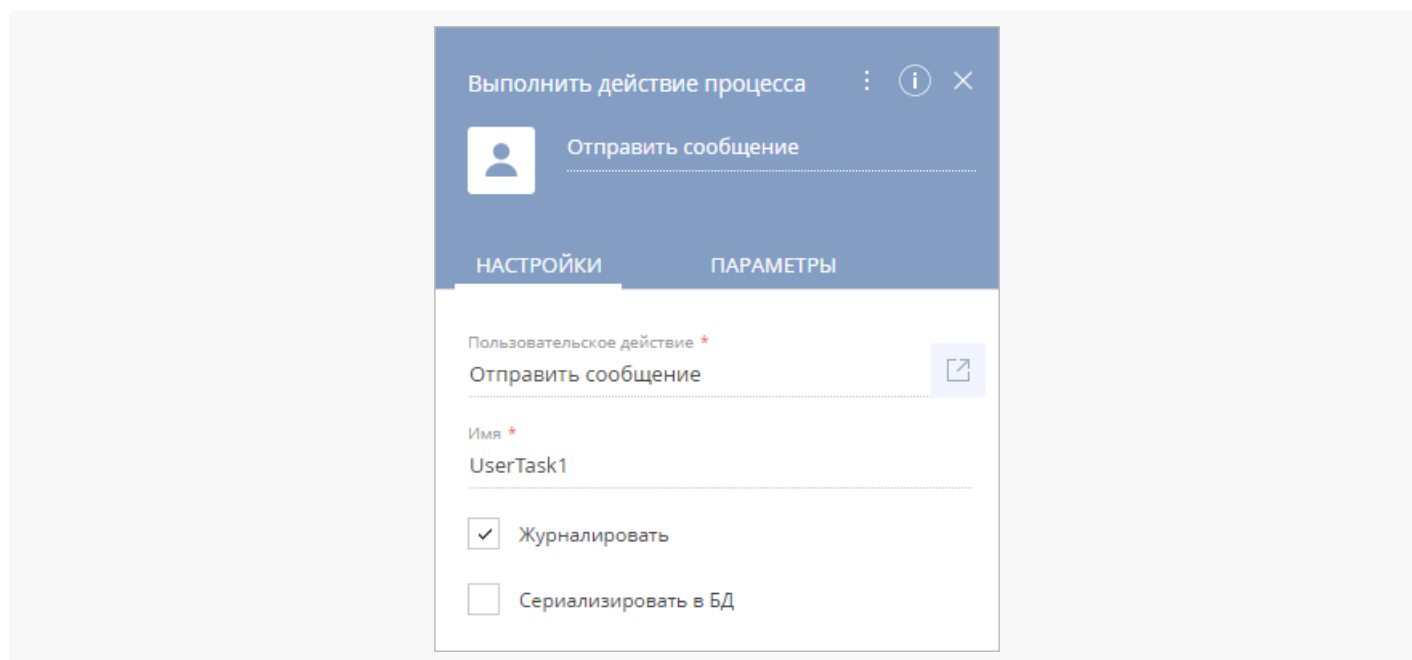
## Элемент процесса [Выполнить действие]

# процесса]

ПРОДУКТЫ: **ВСЕ ПРОДУКТЫ**

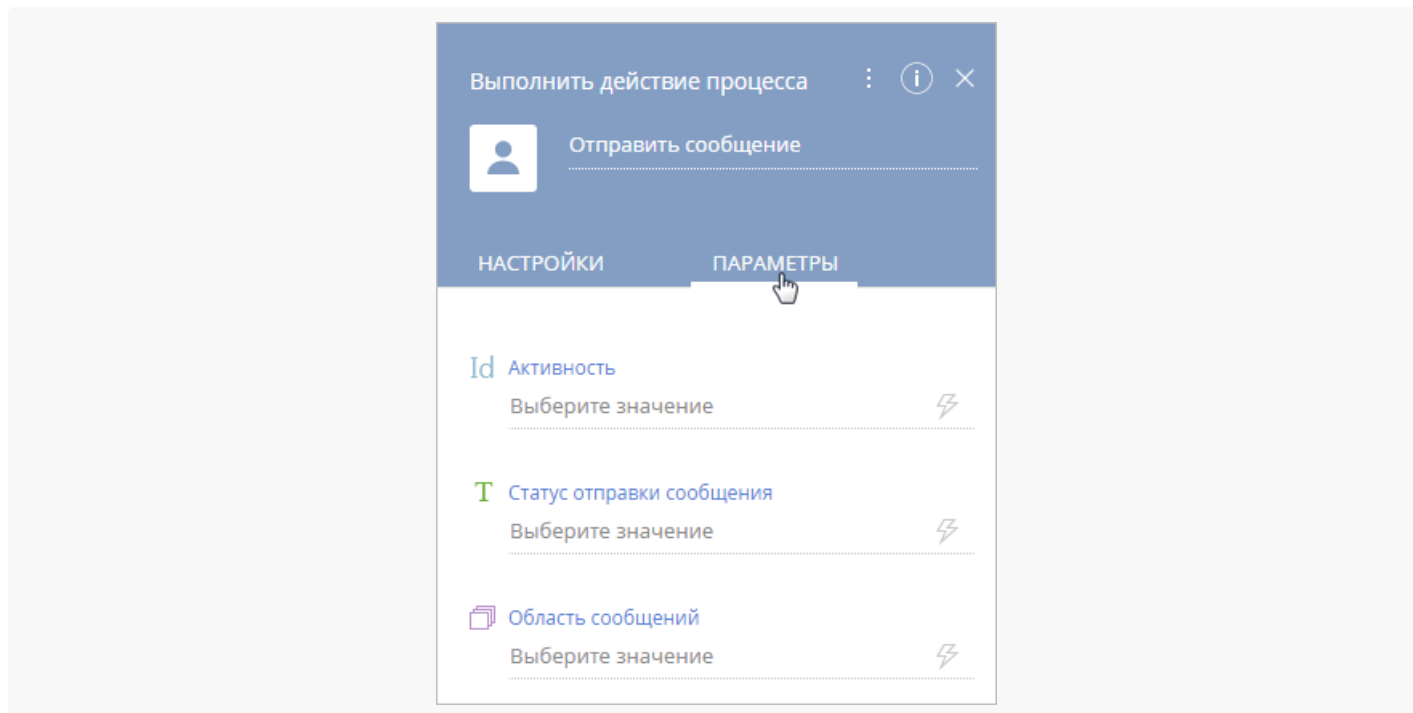
Выполняемые шаги процесса определяются элементами [ *Выполнить действие процесса* ]. Логика работы элемента [ *Выполнить действие процесса* ], а также его параметры зависят от пользовательского действия, которое выбирается на панели настройки элемента (Рис. 1).

Рис. 1 — Панель настройки элемента [ *Выполнить действие процесса* ]




[ *Пользовательское действие* ] — тип действия. После заполнения этого свойства на вкладке [ *Параметры* ] страницы настройки отобразится список параметров выбранного действия процесса (Рис. 2).

Рис. 2 — Список параметров выбранного действия процесса



Поля заполняются при помощи [меню значений параметра](#).

**На заметку.** Заполняя параметры элемента, вы можете указывать конкретные значения либо определять значения динамически на основании информации в других элементах процесса.

Кнопка  используется для перехода в дизайнер пользовательских действий, в котором осуществляется работа с параметрами действия.

Элемент поддерживает [коллекции параметров](#), содержащие сложные значения, каждое из которых включает несколько записей. Например, коллекцию контактов с именем, адресом и номером телефона для каждого из них. Чтобы создать коллекцию для выполнения действия процесса, откройте дизайнер пользовательских действий. Добавьте параметр “Сериализуемый список составных значений”. Далее средствами разработки необходимо настроить нужную бизнес-логику.

Меню [ *Элементы* ] включает основные элементы, которые вы можете использовать при построении бизнес-процессов, например, “Отправить сообщение”.

В списке действий процесса также доступны системные действия, при помощи которых реализована базовая логика разделов Creatio, например, инициализация деталей в разделе. Системные действия используются, как правило, в процессах базовых страниц и объектов и могут наследоваться другими страницами и объектами.

## Элемент процесса [Обработать файл]

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Используйте элемент процесса [ *Обработать файл* ] для автоматизации действий с файлами в бизнес-процессах (Рис. 1).



Рис. 1 — Элемент процесса [ Обработать файл ]



Элемент [ *Обработать файл* ] позволяет следующее:

- Читать и копировать файлы на деталях [ *Файлы и ссылки* ] записей Creatio.
- Получать файлы из параметров процесса.
- Генерировать отчеты Word и Fast Report.

С прочитанными или сгенерированными файлами вы можете выполнять следующие действия:

- Использовать файлы в ходе выполнения процесса, например, отправлять их виде вложений в email-сообщениях.
- Сохранять файлы на деталях [ *Файлы и ссылки* ] других записей (Рис. 2).

Рис. 2 — Настройка элемента [ Обработать файл ]

Отправить отчет

СОХРАНИТЬ ЗАПУСТИТЬ ОТМЕНА ДЕЙСТВИЯ

Введите название или код элемента

Из какого источника получить файл?

Файлы и ссылки объекта

Из какого объекта получить файл?

Необходимо указать значение

# Настроить обработку файлов в ходе процесса

## Пример настройки для файлов объекта

**Пример.** Необходимо скопировать 10 последних файлов, которые были добавлены на деталь [ *Файлы и ссылки* ] определенного контакта, на деталь другого контакта.

Для этого необходимо настроить элемент [ *Обработать файл* ] следующим образом (Рис. 3).

1. [ *Из какого источника получить файл?* ] — выберите “Файлы и ссылки объекта”, чтобы элемент мог читать файлы на детали [ *Файлы и ссылки* ].
2. [ *Из какого объекта получить файл?* ] — выберите объект Creatio, содержащий файлы, которые необходимо получить, в нашем примере — “Файл и ссылка контакта”.
3. [ *Как отфильтровать записи?* ] — установите необходимые фильтры и укажите количество записей для чтения. В нашем примере необходимо выбрать вложения Евгения Мирного и добавить условие чтения первых 10 записей. Таким образом, элемент будет использовать первые 10 записей, которые соответствуют установленному порядку сортировки. Подробнее о настройке фильтров читайте в статье [Фильтры](#).
4. [ *Как отсортировать записи?* ] — укажите порядок сортировки. Использование сортировки записей может быть полезным, например, когда у объекта Creatio на детали [ *Файлы и ссылки* ] хранится множество файлов, а вам необходимо использовать в процессе только последние из них. В нашем примере необходимо отсортировать записи по дате создания в порядке убывания.
5. [ *Какие действия выполнить с файлом?* ] — в поле можно выбрать один из следующих вариантов:
  - а. **“Использовать в процессе”**, если вам необходимо использовать записи файлов в текущем процессе, не сохраняя их в каком-либо объекте Creatio, или передать данные файлы в другой бизнес-процесс в качестве параметра.
  - б. **“Сохранить в “Файлы и ссылки” объекта”**, если вам необходимо сохранить файлы на детали [ *Файлы и ссылки* ] определенного объекта. Вы также можете использовать коллекцию файлов в текущем процессе или передать эти файлы в другой бизнес-процесс в качестве параметра. В нашем примере необходимо выбрать вариант “Сохранить в “Файлы и ссылки” объекта”.

**На заметку.** При выборе значения “Сохранить в “Файлы и ссылки” объекта” для элемента [ *Обработать файл* ] Creatio сохраняет идентификаторы (Id) копируемых файловых записей. При необходимости вы можете настроить в процессе маппинг других элементов с Id этих записей.

6. Поле [ *В какой объект сохранить файл?* ] отображается, если на предыдущем шаге вы выбрали вариант “Сохранить в “Файлы и ссылки” объекта”. Укажите деталь [ *Файлы и ссылки* ] объекта Creatio, на которой будут храниться скопированные файлы. Поскольку в нашем примере необходимо скопировать файлы одного контакта Creatio на деталь другого контакта, выберите значение “Файл и ссылка контакта”.

В зависимости от указанного объекта появляется новое поле для заполнения. В этом поле нужно указать запись, в которой будут храниться сохраненные файлы. В нашем примере в данном поле

указывается связь детали [ *Файлы и ссылки* ] с объектом “Контакт”. При заполнении поля вы можете использовать [параметры](#) либо значения. Например, выберите “Значение справочника” и укажите нужный контакт из списка “Файлы и примечания” —> нажмите [ *Выбрать* ].

Рис. 3— Пример настройки элемента [ *Обработать файл* ] для обработки файлов и ссылок

В результате выполнения элемента в ходе процесса первые 10 файлов детали [ *Файлы и ссылки* ] одного контакта (например, Евгения Мирного), выбранные согласно настроенному порядку сортировки, будут скопированы и сохранены на детали [ *Файлы и ссылки* ] другого контакта, (например, Сергея Авдорова).

## Пример настройки для файлов из параметров процесса

**Пример.** Необходимо получить файл из параметра процесса и добавить его на деталь [ *Файлы и ссылки* ] текущего контакта.

Для этого необходимо настроить элемент [ *Обработать файл* ] следующим образом (Рис. 4).

1. [ *Из какого источника получить файл?* ] — выберите “Параметр процесса”, чтобы элемент мог читать параметры процесса с типом данных “Файл”.
2. [ *Файлы* ] — укажите параметр процесса, из которого необходимо получить файл. Вы можете указать как один файл, так и коллекцию файлов. Нажмите кнопку ⚡ рядом с полем и выберите:
  - “**Параметр процесса**”, чтобы выбрать параметр, значение которого необходимо передать.
  - “**Формула**”, чтобы настроить условия генерации файла на основании параметров процесса.
3. [ *В какой объект сохранить файл?* ] — укажите деталь [ *Файлы и ссылки* ] объекта Creatio, на которой будут храниться скопированные файлы. Поскольку в нашем примере необходимо скопировать файл на деталь текущего контакта, выберите значение “Файл и ссылка контакта”.

В зависимости от указанного объекта появляется новое поле для заполнения. В этом поле нужно указать запись, в которой будут храниться сохраненные файлы. В нашем примере в данном поле указывается связь детали [ *Файлы и ссылки* ] с объектом “Контакт”. При заполнении поля вы можете использовать [параметры](#) либо значения. Например, выберите “Контакт текущего пользователя”.

Рис. 4 — Пример настройки элемента [ *Обработать файл* ] для обработки файлов из параметра процесса

В результате выполнения элемента файл, полученный из параметра процесса, будет сохранен на детали [ *Файлы и ссылки* ] текущего пользователя.

## Настроить генерацию отчетов в ходе процесса

Генерация отчетов в ходе бизнес-процессов доступна в Creatio версии 7.17.2 и выше.

**Пример.** Необходимо сгенерировать отчет по заказам и сохранить его на детали [ *Файлы и ссылки* ] партнерства.

**На заметку.** Раздел [ *Партнерства* ] доступен в продуктах Sales Creatio enterprise и в CRM Creatio.

Для этого необходимо настроить элемент [ *Обработать файл* ] следующим образом (Рис. 4).

1. [ *Из какого источника получить файл?* ] — выберите “Сгенерированный отчет”, чтобы отчет создавался непосредственно в ходе выполнения элемента.
2. [ *Какой отчет сгенерировать?* ] — выберите отчет из списка доступных. В нашем примере — “Заказ”. Для выбора доступны отчеты Word и Fast Report. Если вы выберете отчет Fast Report, то станет доступна опция генерации отчета по каждой записи. Она активируется установкой признака в поле [ *Сгенерировать отдельный отчет для каждой записи* ]. По умолчанию для **Fast Report** формируется один отчет, содержащий данные всех записей, соответствующих условиям фильтрации. Для **Word** всегда формируется отдельный отчет по каждой записи. Подробнее о создании отчетов: [Отчеты и печатные формы](#).
3. [ *Раздел* ] — в поле отображается раздел, из записей которого генерируется отчет. Поле заполняется автоматически и недоступно для редактирования.
4. [ *Как отфильтровать записи?* ] — установите необходимые фильтры. В нашем примере можно отфильтровать заказы, которые оформлялись в определенной валюте — долларах и евро. Подробнее о настройке фильтров читайте в статье [Фильтры](#).
5. [ *Название* ] — укажите условия формирования названия для сгенерированного файла. Название файла будет состоять из имени выбранного отчета и значения, указанного в данном параметре. Например, выберите [ *Колонка из выборки* ] — > [ *Номер* ], чтобы название сгенерированного файла имело формат “Заказ. ORD-30”.
6. [ *Какие действия выполнить с файлом?* ] — в поле можно выбрать один из следующих вариантов:
  - a. **“Использовать в процессе”**, если вам необходимо использовать записи файлов в текущем процессе, не сохраняя их в каком-либо объекте Creatio, или передать данные файлы в другой бизнес-процесс в качестве параметра.
  - b. **“Сохранить в “Файлы и ссылки” объекта”**, если вам необходимо сохранить файлы на детали [ *Файлы и ссылки* ] определенного объекта. Вы также можете использовать коллекцию файлов в текущем процессе или передать эти файлы в другой бизнес-процесс в качестве параметра. В нашем примере необходимо выбрать вариант “Сохранить в “Файлы и ссылки” объекта”.

**На заметку.** При выборе значения “Сохранить в “Файлы и ссылки” объекта” для элемента [ *Обработать файл* ] Creatio сохраняет идентификаторы (Id) копируемых файловых записей. При необходимости вы можете настроить в процессе маппинг других элементов с Id этих записей.

- c. Поле [ *В какой объект сохранить файл?* ] отображается, если на предыдущем шаге вы выбрали вариант “Сохранить в “Файлы и ссылки” объекта”. Укажите деталь [ *Файлы и ссылки* ] объекта Creatio, на которой будут храниться скопированные файлы. Поскольку в ашем примере необходимо

сохранить сгенерированный отчет в записи партнерства, то выберите значение “Файлы партнера”.

В зависимости от указанного объекта появляется новое поле для заполнения. В этом поле нужно указать запись, в которой будут храниться сохраненные файлы. В нашем примере в данном поле указывается связь детали [ *Файлы и ссылки* ] с объектом “Партнерство”. При заполнении поля вы можете использовать [параметры](#) либо значения. Например, выберите “Значение справочника” и укажите нужное партнерство из списка “Партнерство” —> нажмите [ *Выбрать* ].

Рис. 5 — Пример настройки элемента [ *Обработать файл* ] для генерации отчетов

В результате выполнения элемента в ходе процесса будет сформирован набор отчетов по всем отфильтрованным записям раздела [ *Заказы* ]. Отчеты будут сохранены на детали [ *Файлы и ссылки* ] партнерства (например, Виста Ltd).

## Элемент процесса [Поиск и объединение дублей]

ПРОДУКТЫ: **ВСЕ ПРОДУКТЫ**

Элемент процесса [ *Поиск и объединение дублей* ] используется для автоматического объединения дублирующихся записей в разделах Creatio.

**На заметку.** Для работы с элементом необходимо, чтобы в приложении были настроены [глобальный поиск](#) и [массовый поиск дублей](#) дублей. Также убедитесь, что в разделе, для которого вы планируете выполнять поиск и слияние дублей бизнес-процессом, есть хотя бы одно активное правило поиска дублей.

В качестве результирующей записи будет использована запись с самой ранней датой создания. При этом:

- Заполненные поля записи с самой ранней датой создания остаются без изменений.
- Незаполненные поля записи с самой ранней датой создания заполняются данными самой поздней записи.
- Детали содержат все данные с деталей всех объединенных записей.

Параметры поиска дублей задаются на панели настройки элемента (Рис. 1):

Рис. 1 — Пример панели настройки элемента [ *Поиск и объединение дублей* ]

Поля заполняются при помощи меню значений параметра:

1. [ *Объект* ] – укажите раздел, для которого будет выполняться поиск дублей. В списке отображаются только те разделы, для которых существует хотя бы одно правило активное поиска дублей.
2. [ *Идентификатор записи* ] – укажите идентификатор записи, для которой необходимо выполнить поиск дублей.
3. [ *По каким правилам искать дубли?* ] – укажите правила, по которым будет выполняться поиск дублей:
  - [ *Все активные правила* ] – применить все активные правила поиска дублей для выбранного раздела. Используется по умолчанию.
  - [ *Выбранные правила* ] – нажмите [ *Добавить правило* ] и выберите необходимые правила. Если вы выбрали этот вариант, но не добавили ни одного правила, то после завершения настройки значение поля автоматически изменится на [ *Все активные правила* ].

Если вы деактивируете правило поиска дублей, которое указано в элементе, то при его выполнении это правило не будет использоваться. Если вы деактивируете все правила поиска дублей, указанных в элементе, то бизнес-процесс, который содержит данный элемент, не сможет завершиться. Описание причин ошибки вы найдете в [ *Журнале процессов* ].

**На заметку.** При деактивации правила поиска дублей Creatio предупредит вас, что оно может использоваться в бизнес-процессах. Однако Creatio не проверяет, использует ли его какой-либо элемент [ *Поиск и объединение дублей* ].