

# Веб-сервисы

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>Начало работы с low-code интеграцией веб-сервисов</b>	<b>4</b>
Общая процедура настройки интеграции с веб-сервисом	4
Изучение документации веб-сервиса	4
<b>Настроить интеграцию с веб-сервисом SOAP</b>	<b>5</b>
Автоматическая настройка свойств и методов веб-сервиса	6
Ручная настройка свойств и методов веб-сервиса	7
Проверить настройку интеграции с SOAP-сервисом	21
<b>Настроить интеграцию с веб-сервисом REST</b>	<b>22</b>
Настроить свойства и методы веб-сервиса	23
Настроить параметры запроса	29
Настроить параметры ответа	35
Настроить параметры запроса и ответа типа “коллекция”	41
Проверить настройку интеграции с REST-сервисом	43
<b>Настроить аутентификацию веб-сервиса</b>	<b>45</b>
Настроить базовую аутентификацию	45
Настроить OAuth-аутентификацию	46

# Начало работы с low-code интеграцией веб-сервисов

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Creatio поддерживает интеграцию с пользовательскими REST и SOAP сервисами с помощью инструментов low-code. На основе бизнес-логики пользователя Creatio сгенерирует и отправит запрос веб-сервису, получит ответ и предоставит необходимые данные. Данные, полученные от веб-сервиса, можно использовать для создания или обновления записей в базе данных Creatio, а также для реализации пользовательской бизнес-логики или автоматизации.

## Общая процедура настройки интеграции с веб-сервисом

Основные параметры подключения к веб-сервису настраиваются в разделе [ *Веб-сервисы* ] рабочего места [ *Студия* ]. Здесь вы можете настроить параметры интеграции с веб-сервисом, в том числе:

- Подключение Creatio к веб-сервису и аутентификацию в нем.
- Формирование запросов Creatio к веб-сервису.
- Чтение ответов от веб-сервиса.

**На заметку.** Сложность настройки во многом зависит от того, как реализован конкретный веб-сервис и документация к нему. Наиболее распространенные варианты интеграции с веб-сервисами не требуют специальных навыков разработки.

Общая процедура настройки интеграции с веб-сервисом включает следующее:

1. Добавление веб-сервиса и настройка его свойств и методов. Настройки отличаются для [REST](#) и [SOAP](#)-сервисов.
2. Настройка аутентификации веб-сервиса. Опциональный шаг. Настройка аутентификации идентична для REST и SOAP-сервисов и описана в статье [Аутентификация веб-сервиса](#).
3. Проверка интеграции с веб-сервисом. Настройки отличаются для [REST](#) и [SOAP](#)-сервисов.

После этого можно использовать интеграцию с веб-сервисом в пользовательских бизнес-процессах. Пример настройки процесса описан в статье [Использовать интеграцию с веб-сервисами для автоматического обновления курсов валют](#).

Подробнее процессы настройки [REST](#) и [SOAP](#) сервисов описаны в соответствующих главах.

## Изучение документации веб-сервиса

Перед тем как приступить к настройке интеграции с сервисом, необходимо изучить способы его вызова и структуру ответа, который будет получен. Эта информация, как правило, доступна в документации веб-сервиса.

Например, согласно <http://fixer.io/>, для вызова веб-сервиса “Fixer” можно использовать такие запросы типа GET:

- **<https://data.fixer.io/latest>** — для получения последних актуальных курсов обмена валют.
- **<https://data.fixer.io/2000-01-03>** — для получения курсов валют на конкретную дату (в данном конкретном случае — на 3 января 2000 г.).

# Настроить интеграцию с веб-сервисом SOAP

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Creatio поддерживает интеграцию с пользовательскими SOAP-сервисами с помощью инструментов low-code. Creatio сгенерирует и отправит запрос веб-сервису, получит ответ и предоставит необходимые данные. Данные, полученные от веб-сервиса, можно использовать для создания или обновления записей в Creatio, а также для реализации пользовательской бизнес-логики или автоматизации.

Общая последовательность настройки одинакова для всех SOAP-сервисов, детали во многом зависят от специфики веб-сервиса.

**Этапы** настройки интеграции с веб-сервисом:

1. [Добавление веб-сервиса и настройка](#) его свойств и методов.
2. Настройка аутентификации веб-сервиса (опциональный шаг). [Настройка аутентификации](#) идентична для REST и SOAP-сервисов.
3. [Проверка настройки интеграции с веб-сервисом](#).

После этого можно использовать интеграцию с веб-сервисом в пользовательских бизнес-процессах.

Пример настройки процесса описан в статье [Использовать интеграцию с веб-сервисами для автоматического обновления курсов валют](#).

**Пример.** Настроить интеграцию с SOAP-сервисом “PhoneVerify” (<https://ws.cdyne.com/phoneverify/phoneverify.asmx>) для получения информации по номеру телефона.

Необходимо реализовать поддержку следующих параметров:

- **“CheckPhoneNumber”** — метод, который возвращает информацию по номеру телефона.
- **“PhoneNumber”** — параметр запроса. В него необходимо передать номер телефона. Обязательный параметр.
- **“LicenseKey”** — параметр запроса. В него необходимо передать ключ.
- **“Company”** — параметр ответа. Содержит название компании.
- **“Valid”** — параметр ответа. Содержит информацию о корректности полученной информации.

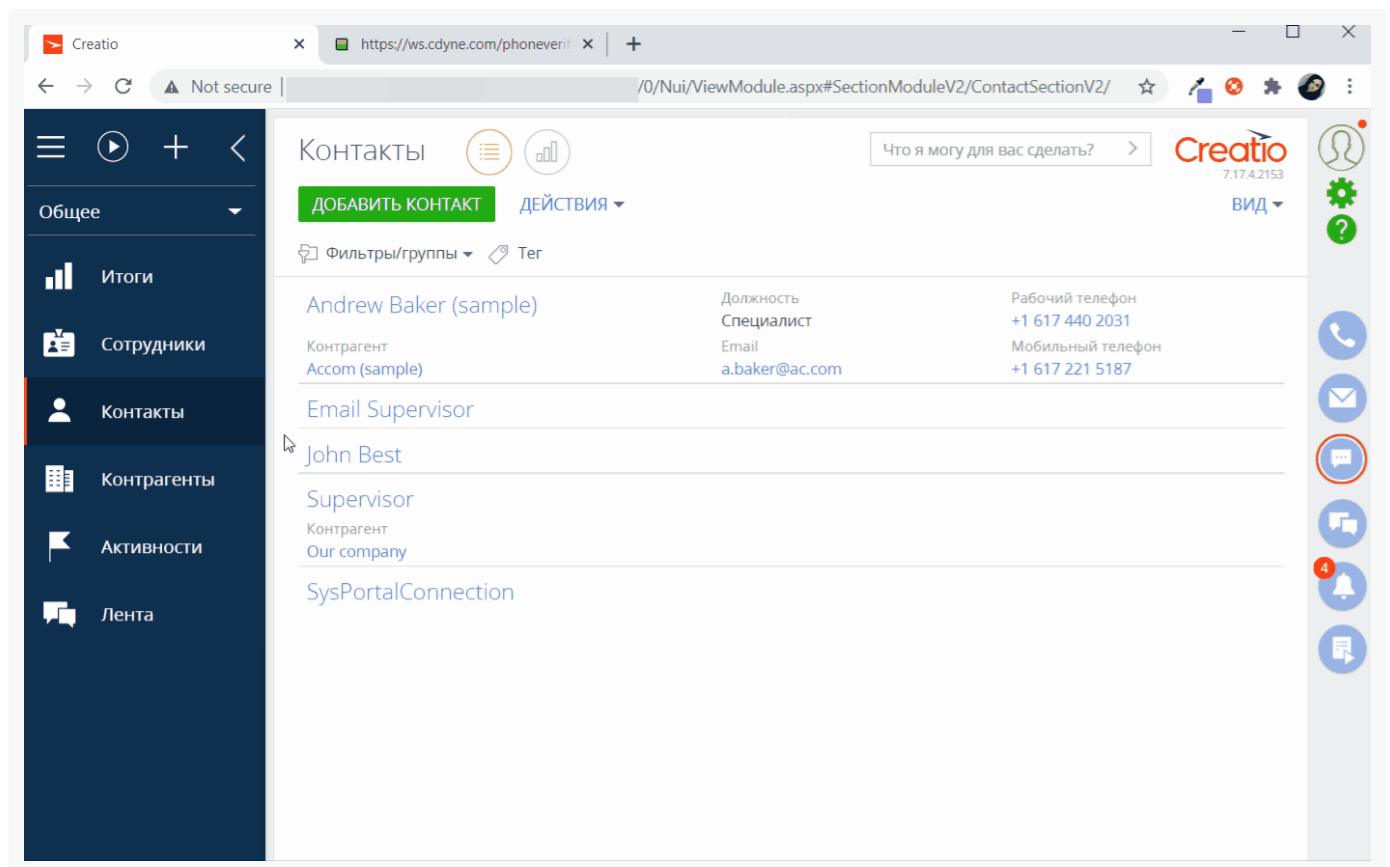
Creatio позволяет импортировать wsdl-файл или настроить интеграцию вручную.

**На заметку.** wsdl-файл содержит документацию SOAP-сервиса. wsdl-файл можно получить из открытых источников сервиса, с которым планируется интеграция.

## Автоматическая настройка свойств и методов веб-сервиса

1. Перейдите в рабочее место [ *Студия* ] и откройте раздел [ *Веб-сервисы* ].
2. Нажмите [ *Добавить веб-сервис* ] —> [ *SOAP сервис* ].
3. В окне “Быстрая настройка SOAP веб-сервиса” нажмите [ *Выбрать файл* ] или добавьте ссылку на wsdl-файл с описанием веб-сервиса, интеграцию с которым необходимо настроить.
4. В окне “Настройка SOAP веб-сервиса” выберите, какой сервис и версию SOAP использовать. Также настройте методы, которые необходимо вызывать, и параметры вызова методов. Нажмите [ *Далее* ].
5. В окне “Настройка SOAP веб-сервиса” выберите необходимые параметры ответов методов. Нажмите [ *Сохранить* ] (Рис. 1).

Рис. 1 — Автоматическая настройка веб-сервиса



Настройка веб-сервиса выполнена. Все свойства и методы на странице сервиса будут заполнены автоматически. Далее переходите к [проверке интеграции с SOAP-сервисом](#).

Ниже приведен альтернативный способ настройки свойств и методов веб-сервиса.

## Ручная настройка свойств и методов веб-сервиса

1. Перейдите в рабочее место [ *Студия* ] и откройте раздел [ *Веб-сервисы* ].
2. Нажмите [ *Добавить веб-сервис* ] —> [ *SOAP сервис* ].
3. В окне “Быстрая настройка SOAP веб-сервиса” нажмите [ *Пропустить быструю настройку* ].
4. Изучите документацию веб-сервиса. В нашем примере используется следующий код.

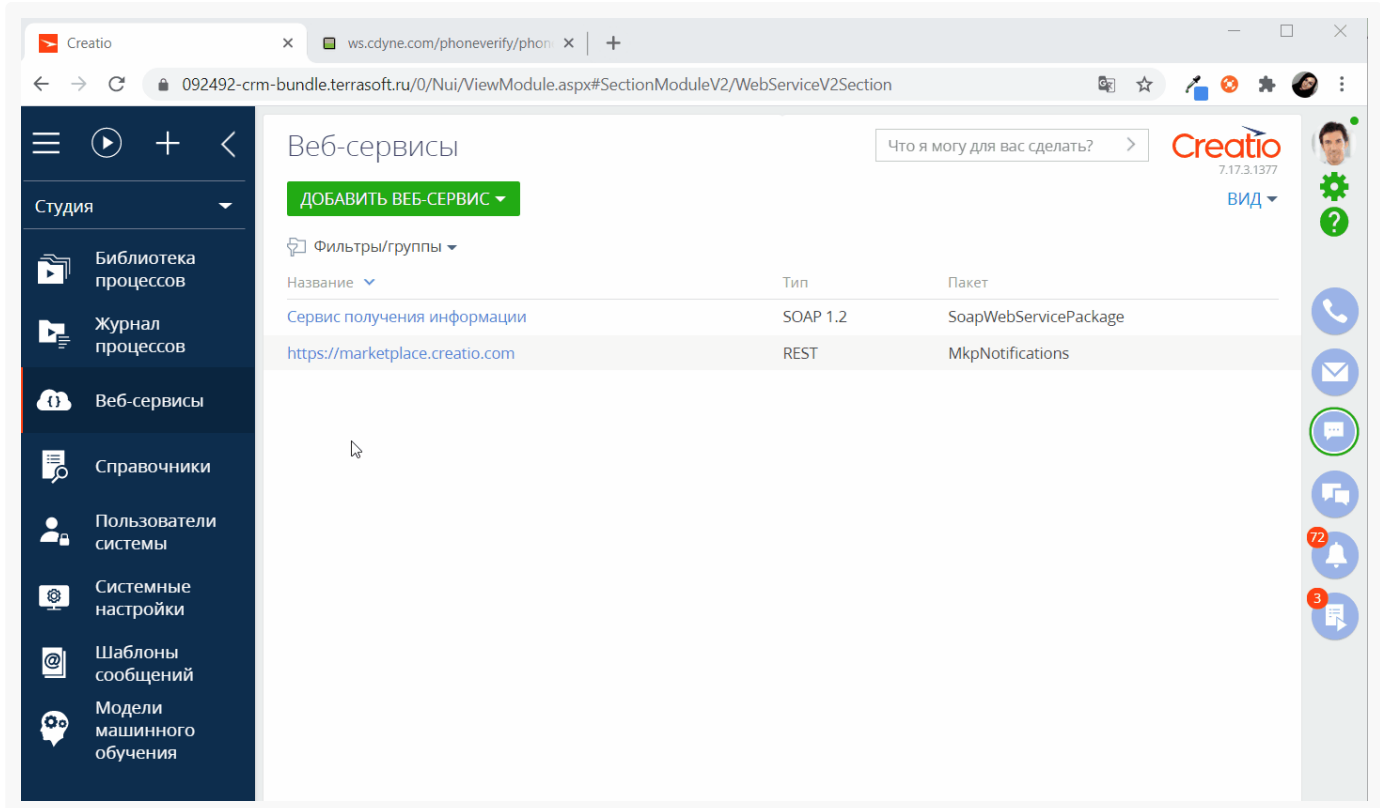
```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xml
...
</wsdl:definitions>
```

5. Заполните **поля страницы свойств** веб-сервиса (Рис. 2).

Поле	Комментарий	Пример
<b>Название</b>	Название будет отображаться в поле [ <i>Какой сервис вызывать?</i> ] области свойств элемента процесса [ <i>Вызвать веб-сервис</i> ].	Сервис получения информации по телефонному номеру
<b>Код</b>	Используется разработчиками для взаимодействия с веб-сервисом в программном коде Creatio. В данном случае уникальное имя интеграции с веб-сервисом состоит из его названия и префикса “Usr”.	UsrPhoneVerifyService
<b>Пространства имен</b>	Необходимо указывать только пространства имен, которые используются для методов и их параметров. Набор пространств имен задается в формате “Префикс пространства имен”: “Пространство имен”, разделенных символом “;” или переводом строки. Если пространство имен только одно, то его можно задать без префикса. Это пространство имен будет применено ко всему запросу.	http://ws.cdyne.com/ PhoneVerify/query
<b>URI сервиса</b>	Полный адрес вызова веб-сервиса будет состоять из этого URI и настроек, указанных на странице настройки метода. Используйте такой же протокол (http/https), как и у сайта вашего приложения Creatio. Если веб-сервис содержится в недоступном для редактирования пакете, то его URI будет доступен для редактирования.	http://ws.cdyne.com/ phoneverify/ phoneverify.asmx?WSDL
<b>Повторов вызова при ошибках</b>	Если ответ от веб-сервиса пришел с кодом ошибки или истек тайм-аут ответа, то запрос будет повторен указанное количество раз. При заполнении этого поля учитывайте тайм-аут ответа, который будет указан для методов веб-сервиса.	По умолчанию — 0
<b>Пакет</b>	Пакет, в котором будет сохранена данная интеграция с веб-сервисом. В списке отображаются пакеты, которые доступны для изменения текущим пользователем.	SoapWebServicePackage

Рис. 2 — Заполнение страницы свойств веб-сервиса





**На заметку.** Интеграции с веб-сервисами сохраняются в виде элементов конфигурации. Если такой элемент конфигурации находится в пакете, который недоступен для редактирования, то вы сможете изменить только URI веб-сервиса. Для внесения других изменений (например, для добавления методов) в такие интеграции с веб-сервисами скопируйте соответствующие элементы конфигурации в доступный для редактирования пакет.

## Настроить вызов метода

Для каждого веб-сервиса необходимо настроить вызов методов. При этом для одного веб-сервиса можно настроить несколько методов.

**На заметку.** В рамках одного процесса можно вызывать любое количество методов, используя несколько [элементов процесса \[ Вызвать веб-сервис \]](#), с передачей значений параметров между ними.

1. На детали [ *Методы* ] страницы настройки интеграции с веб-сервисом при помощи кнопки **+** добавьте методы.
2. Изучите документацию веб-сервиса. В нашем примере используется следующий код.

```
<wsdl:types>
  <s:schema elementFormDefault="qualified" targetNamespace="http://ws.cdyne.com/PhoneVerify"
    <s:element name="CheckPhoneNumber">
```

```
...
</s:element>
</wsdl:types>
```

### 3. Заполните **свойства метода** (Рис. 3).

Поле	Комментарий	Пример
<b>Название</b>	Название будет отображаться в поле [ <i>Какой сервис вызывать?</i> ] области свойств элемента процесса [ <i>Вызвать веб-сервис</i> ].	Проверить номер телефона
<b>Код (на английском)</b>	Используется разработчиками для взаимодействия с веб-сервисом в программном коде Creatio. В данном случае уникальное имя интеграции с веб-сервисом состоит из его названия и префикса “Usr”.	UsrCheck PhoneNumber
<b>Код элемента сообщения</b>	<p>Название узла XML, который является корневым внутри &lt;soap:Body&gt;. Часто совпадает с названием операции в WSDL. Например, это будет “MessageElementName” в следующем теле запроса:</p> <pre>&lt;soap:Body&gt;   &lt;MessageElementName&gt;     &lt;Parameters&gt;       ...     &lt;/Parameters&gt;   &lt;/MessageElementName&gt; &lt;/soap:Body&gt;</pre>	CheckPhoneNumber
<b>SOAP Action</b>	Для определения данного значения используется документация веб-сервиса. Например, у веб-сервиса “PhoneVerify” есть “конечная точка” “CheckPhoneNumber”, которая возвращает информацию по номеру телефона.	http://ws.cdyne.com/ PhoneVerify/query/ CheckPhoneNumber
<b>Тайм-аут ответа, мс</b>	Время ожидания ответа от веб-сервиса. Если после отправки запроса не был получен ответ, либо был получен код ошибки, то по истечении этого времени Creatio повторит запрос (если еще остались неиспользованные попытки повторного вызова).	По умолчанию — 5 000

<b>Использовать аутентификацию</b>	<b>Комментарий</b> аутентификацию для доступа к веб-сервису. Необходимо настроить аутентификацию на детали [ Аутентификация ] страницы настройки интеграции с веб-сервисом. Подробнее: <a href="#">Аутентификация веб-сервиса</a> .	<b>Пример</b> наванию — false
------------------------------------	---	-------------------------------

Рис. 3 — Свойства метода веб-сервиса

ОКОТМЕНАОТПРАВИТЬ ТЕСТОВЫЙ ЗАПРОС

Название\* Проверить номер телефона

Тайм-аут ответа, мс\* 5 000

Код (на английском)\*

UsrCheckPhoneNumber

Использовать аутентификацию

☐

Код элемента сообщения\*

CheckPhoneNumber

SOAP Action

http://ws.cdyne.com/PhoneVerify/query/CheckPhoneNumber

ПАРАМЕТРЫ ВЫЗОВА

ОБРАБОТКА ОТВЕТА

ДОБАВИТЬ ПАРАМЕТР

:

### Настроить параметры запроса

Параметры запроса используются для генерации URL-адреса конечной точки, который Creatio будет использовать при вызове веб-сервиса.

Количество и типы параметров запроса определяются спецификой веб-сервиса. Параметры необходимо добавлять относительно метода.

Доступны следующие **типы параметров запроса**:

<b>Параметр тела</b>	Этот тип параметра используется для отправки любых типов данных (включая коллекции) в теле запроса. Подробнее о методе POST запроса читайте в <a href="#">Википедии</a> .
<b>Параметр заголовка</b>	Этот тип параметра используется для генерации заголовка запроса. Подробнее об HTTP-заголовках читайте в <a href="#">Википедии</a> .
<b>Параметр cookies</b>	Параметры этого типа используются для передачи файлов cookie в запросах к сервису. Например, можно передать полученный ранее cookie для аутентификации. Подробнее о cookie читайте в <a href="#">Википедии</a> .

- Для настройки параметров запроса:
- Изучите документацию веб-сервиса. В нашем примере используется следующий код.

```
<s:element name="CheckPhoneNumber">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="PhoneNumber" type="s:string" />
      <s:element minOccurs="0" maxOccurs="1" name="LicenseKey" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
```

## 2. Добавьте **параметры запроса**:

- a. На вкладке [ *Параметры вызова* ] нажмите кнопку [ *Добавить параметр* ].
- b. Добавьте параметр “Номер телефона” и заполните его свойства (Рис. 4).

Поле	Комментарий	Пример
<b>Название</b>	Название параметра сервиса.	Номер телефона
<b>Тип параметра</b>	Тип параметра сервиса.	Параметр тела
<b>Путь к элементу</b>	Если для параметра используется пространство имен, то оно указывается в формате “Префикс пространства имен”: “Путь к параметру”.	PhoneNumber
<b>Код в Creatio</b>	Используется разработчиками для взаимодействия с веб-сервисом в программном коде Creatio. В данном случае уникальное имя интеграции с веб-сервисом состоит из его названия и префикса “Usr”.	UsrPhoneNumber
<b>Тип данных</b>	Тип данных параметров сервиса. Параметр с вложенными элементами должен иметь тип данных “Объект”.	Текст
<b>Является массивом</b>	Значение параметра-массива нельзя задать в элементе “Вызвать веб-сервис” в дизайнера процессов. Необходимо использовать элемент “Задание-сценарий”. Параметр с типом данных “Объект” обязательно должен быть массивом.	По умолчанию признак снят
<b>Обязательный</b>	При установленном признаке параметр будет обязательным в дизайнера процессов. Признак недоступен для редактирования при выборе значения по умолчанию.	По умолчанию признак установлен
<b>Значение по умолчанию</b>	Значение параметра по умолчанию.	Константа

Рис. 4 — Настройка параметра “Номер телефона”

ПАРАМЕТРЫ ВЫЗОВА

ОБРАБОТКА ОТВЕТА

ДОБАВИТЬ ПАРАМЕТР

⋮

Название	Тип	Значение по умолчан...
T Номер телефона	Параметр тела	
T Ключ	Параметр тела	?

Название\* Номер телефона

Тип параметра\* Параметр тела

Путь к элементу\* PhoneNumber

Код в Creatio\* UspPhoneNumber

Тип данных\* Текст

Является массивом ☐

Обязательный ☒

Значение по умолчанию Константа ▾

с. Добавьте параметр “Ключ” и заполните его свойства (Рис. 5).

Поле	Комментарий	Пример
<b>Название</b>	Название параметра сервиса.	Ключ
<b>Тип параметра</b>	Тип параметра сервиса.	Параметр тела
<b>Путь к элементу</b>	Если для параметра используется пространство имен, то оно указывается в формате “Префикс пространства имен”：“Путь к параметру”.	LicenseKey
<b>Код в Creatio</b>	Используется разработчиками для взаимодействия с веб-сервисом в программном коде Creatio. В данном случае уникальное имя интеграции с веб-сервисом состоит из его названия и префикса “Usr”.	UsrLicenseKey
<b>Тип данных</b>	Тип данных параметров сервиса. Параметр с вложенными элементами должен иметь тип данных “Объект”.	Текст
<b>Является массивом</b>	Значение параметра-массива нельзя задать в элементе “Вызвать веб-сервис” в дизайнера процессов. Необходимо использовать элемент “Задание-сценарий”. Параметр с типом данных “Объект” обязательно должен быть массивом.	По умолчанию признак снят
<b>Обязательный</b>	При установленном признаке параметр будет обязательным в дизайнера процессов. Признак недоступен для редактирования при выборе значения по умолчанию.	По умолчанию признак установлен
<b>Значение по умолчанию</b>	Значение параметра по умолчанию.	Константа

Рис. 5 — Настройка параметра “Ключ”

Название	Тип	Значение по умолчанию...
Номер телефона	Параметр тела	
Ключ	Параметр тела	?

Название *	Ключ
Тип параметра *	Параметр тела
Путь к элементу *	LicenseKey
Код в Creatio *	UsrLicenseKey
Тип данных *	Текст
Является массивом	<input type="checkbox"/>
Обязательный	<input type="checkbox"/>
Значение по умолчанию	Константа ▼ ?

## Настроить параметры ответа

Для настройки параметров ответа:

1. Изучите документацию веб-сервиса. В нашем примере используется следующий код.

```
<wsdl:types>
  <s:schema elementFormDefault="qualified" targetNamespace="http://ws.cdyne.com/PhoneVerify"
    <s:element name="CheckPhoneNumberResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="CheckPhoneNumberResult" type="
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:complexType name="PhoneReturn">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="Company" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="Valid" type="s:boolean" />
        ...
      </s:sequence>
    </s:complexType>
    ...
  </wsdl:types>
```

2. Добавьте **параметры обработки ответа**:

- а. На вкладке [ *Обработка ответа* ] нажмите кнопку [ *Добавить параметр* ].
- б. Добавьте параметр “Компания” и заполните его свойства (Рис. 6).



Поле	Комментарий	Пример
<b>Название</b>	Название параметра сервиса.	Компания
<b>Тип параметра</b>	Тип параметра сервиса.	Параметр тела
<b>Путь к элементу</b>	Если для параметра используется пространство имен, то оно указывается в формате “Префикс пространства имен”: “Путь к параметру”.	CheckPhoneNumberResponse/ CheckPhoneNumberResult/ Company
<b>Код в Creatio</b>	Используется разработчиками для взаимодействия с веб-сервисом в программном коде Creatio. В данном случае уникальное имя интеграции с веб-сервисом состоит из его названия и префикса “Usr”.	UsrCompany
<b>Тип данных</b>	Тип данных параметров сервиса. Параметр с вложенными элементами должен иметь тип данных “Объект”.	Текст
<b>Является массивом</b>	Значение параметра-массива нельзя задать в элементе “Вызвать веб-сервис” в дизайнера процессов. Необходимо использовать элемент “Задание-сценарий”. Параметр с типом данных “Объект” обязательно должен быть массивом.	По умолчанию признак снят
<b>Значение по умолчанию</b>	Значение параметра по умолчанию.	Константа

Рис. 6 — Настройка параметра “Компания”

ПАРАМЕТРЫ ВЫЗОВА

ОБРАБОТКА ОТВЕТА

ДОБАВИТЬ ПАРАМЕТР

⋮

Название	Тип	Значение по умолчан...
T Компания	Параметр тела	
Валидно	Параметр тела	

Название\* Компания

Тип параметра\* Параметр тела

Путь к элементу\* CheckPhoneNumberResponse/CheckPhoneNumb...

Код в Creatio\* UsrCompany

Тип данных\* Текст

Является массивом ☐

Значение по умолчанию Константа ▾

с. Добавьте параметр “Валидно” и заполните его свойства (Рис. 7).

Поле	Комментарий	Пример
Название	Название параметра сервиса.	Валидно
Тип параметра	Тип параметра сервиса.	Параметр тела
Путь к элементу	Если для параметра используется пространство имен, то оно указывается в формате “Префикс пространства имен”：“Путь к параметру”.	CheckPhoneNumberResponse/CheckPhoneNumberResult/Valid
Код в Creatio	Используется разработчиками для взаимодействия с веб-сервисом в программном коде Creatio. В данном случае уникальное имя интеграции с веб-сервисом состоит из его названия и префикса “Usr”.	UsrValid
Тип данных	Тип данных параметров сервиса. Параметр с вложенными элементами должен иметь тип данных “Объект”.	Логическое
Является массивом	Значение параметра-массива нельзя задать в элементе “Вызвать веб-сервис” в дизайнера процессов. Необходимо использовать элемент “Задание-сценарий”. Параметр с типом данных “Объект” обязательно должен быть массивом.	По умолчанию признак снят
Значение по умолчанию	Значение параметра по умолчанию.	Константа

Рис. 7 — Настройка параметра “Валидно”

Нажмите кнопку [ *Ок* ] для сохранения настроек.

## Настроить параметры запроса и ответа типа “коллекция”

Коллекцией (или массивом) является набор элементов. Creatio может передавать коллекции данных в веб-сервис и обрабатывать его ответы, содержащие коллекции. Если веб-сервис поддерживает получение и/или отправку массивов данных, то параметры типа “коллекция” можно использовать как для вызова веб-сервиса, так и для обработки его ответа.

### Типы параметров коллекции:

- **Простая коллекция.** Любой параметр можно представить в виде коллекции, установив в свойствах параметров признак “Является массивом”. Простые коллекции являются массивами значений одного типа данных. Каждое значение является отдельным элементом коллекции. Например, “1, 2, 3” — это простой массив значений целых чисел, а “Бостон, Нью Йорк, Чикаго” — простой массив текстовых значений.
- **Коллекция объекта.** Коллекция представляет собой корневой параметр (т. н. объект), который содержит вложенные параметры.

Для настройки коллекции:

1. Изучите документацию веб-сервиса. В нашем примере используется следующий код.

```
<s:element name="CheckPhoneNumbers">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="PhoneNumbers" type="tns:ArrayOfString" />
      <s:element minOccurs="0" maxOccurs="1" name="LicenseKey" type="s:string" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:complexType name="ArrayOfString">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="string" nillable="true" type="s:string" />
  </s:sequence>
</s:complexType>
```

```
</s:complexType>
```

- Добавьте метод “CheckPhoneNumbers”. Поля методов описаны на шаге [Настроить вызов метода](#).
- Настройте **параметры запроса**. Поля параметров описаны на шаге [Настроить параметры запроса](#).

Обратите внимание, что в соответствии с документацией веб-сервиса параметр запроса “ArrayOfString” — комплексный тип, который состоит из массива элементов “string”. Для настройки параметров запроса, который является коллекцией:

- В поле [ *Тип параметра* ] выберите “Параметр тела”.
- Установите признак [ *Является массивом* ].
- В поле [ *Название элемента массива* ] укажите “string” (Рис. 8). По умолчанию — “item” (для простых массивов).

Рис. 8 — Настройка параметров запроса

The screenshot shows the 'PARAMETERS CALL' (ПАРАМЕТРЫ ВЫЗОВА) tab. A table lists parameters with columns: Name, Type, and Default Value. The parameter 'Номера телефонов' (Phone Numbers) is selected. A modal dialog is open for editing this parameter. The dialog contains the following fields:

- Название (Name): Номера телефонов
- Тип параметра (Parameter Type): Параметр тела
- Путь к элементу (Path to element): PhoneNumbers
- Код в Creatio (Code in Creatio): UsrPhoneNumbers
- Тип данных (Data Type): Текст
- Является массивом (Is Array): ☒
- Название элемента массива (Array Element Name): string
- Обязательный (Required): ☒

The 'Является массивом' and 'Название элемента массива' fields are highlighted with a red rectangle. At the bottom right of the dialog are 'OK' and 'ОТМЕНА' (Cancel) buttons.

- Изучите документацию веб-сервиса. В нашем примере используется следующий код.

```
<s:element name="CheckPhoneNumbersResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1" name="CheckPhoneNumbersResult" type="tns:A" />
    </s:sequence>
  </s:complexType>
</s:element>
<s:complexType name="ArrayOfPhoneReturn">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="unbounded" name="PhoneReturn" type="tns:PhoneReturn" />
  </s:sequence>
</s:complexType>
<s:element name="PhoneReturn" type="tns:PhoneReturn" />
<s:element name="ArrayOfPhoneReturn" nillable="true" type="tns:ArrayOfPhoneReturn" />
<s:complexType name="PhoneReturn">
  <s:sequence>
```

```

<s:element minOccurs="0" maxOccurs="1" name="Company" type="s:string" />
<s:element minOccurs="1" maxOccurs="1" name="Valid" type="s:boolean" />
...
</s:sequence>
</s:complexType>

```

5. Настройте **параметры ответа**. Поля методов описаны на шаге [Настроить параметры ответа](#).

В поле [ *Тип данных* ] выберите “Объект”. После этого станут доступны вложенные параметры (Рис. 9).

Рис. 9 — Настройка типа данных

Параметры ответа веб-сервиса типа “коллекция” могут использоваться в качестве входящих параметров элемента бизнес-процесса [ *Вызвать веб-сервис* ]. Подробнее: [Элемент процесса \[ Вызвать веб-сервис \]](#).

## Проверить настройку интеграции с SOAP-сервисом

Для дальнейшего использования данных от сервиса в пользовательских бизнес-задачах необходимо проверить настройку интеграции.

Начиная с версии 7.18.0 появилась возможность проверить настройку интеграции с SOAP-сервисом из интерфейса приложения, без создания [тестового бизнес-процесса](#).

Чтобы **проверить настройку интеграции с SOAP-сервисом**:

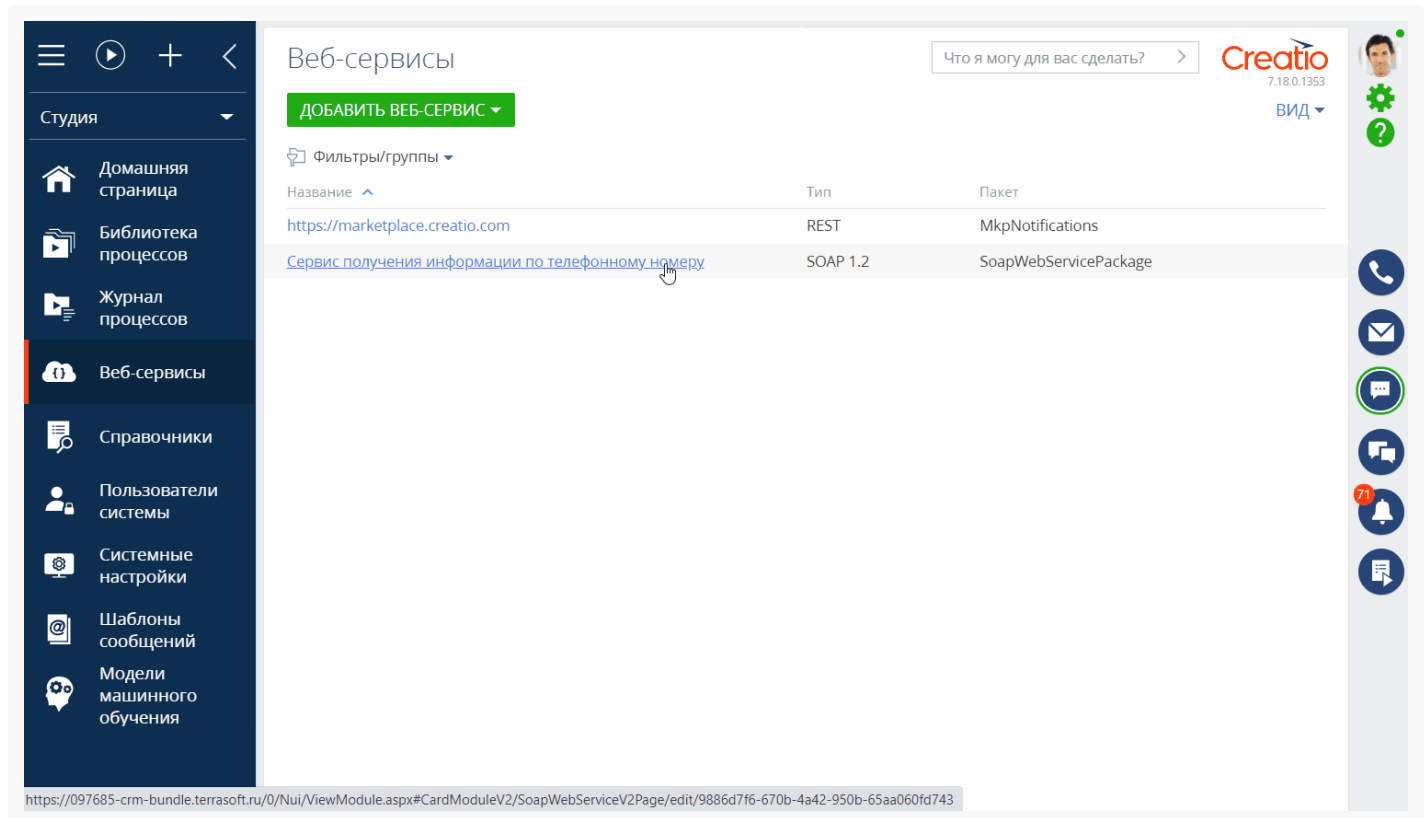
1. Перейдите в рабочее место [ *Студия* ] и откройте раздел [ *Веб-сервисы* ].
2. Откройте страницу веб-сервиса, интеграцию с которым необходимо проверить. В нашем примере это “Сервис получения информации по телефонному номеру”.
3. На детали [ *Методы* ] страницы настройки интеграции с веб-сервисом откройте метод “Проверить номер телефона”.
4. На странице настройки метода нажмите [ *Отправить тестовый запрос* ].
5. Задайте значения параметров вызова метода.

Параметр	Значение
Номер телефона	+1 617 765 7997

6. Нажмите [ *Отправить запрос* ] (Рис. 10).

В результате выполнения тестового запроса будет получен ответ от веб-сервиса. Значения параметров ответа отображены на вкладке [ *Параметры ответа* ] в соответствующих полях (Рис. 10). Другие вкладки позволяют посмотреть запрос и ответ на запрос на языках XML и HTTP (формат данных raw).

Рис. 10 — Проверка интеграции с SOAP-сервисом



Если при настройке интеграции с веб-сервисом была допущена ошибка, то в результате тестового запроса будет получен ответ с пустыми значениями параметров. В этом случае проверьте все настройки сервиса и повторите попытку.

После проверки вы можете использовать интеграцию с веб-сервисом в пользовательских бизнес-процессах. Подробнее данный пример описан в статье [Использовать интеграцию с веб-сервисами для автоматического обновления курсов валют](#).

В зависимости от особенностей веб-сервиса, прежде чем он станет доступным для использования, может потребоваться выполнить процедуру аутентификации. Подробнее: [Настроить аутентификацию веб-сервиса](#).

## Настроить интеграцию с веб-сервисом REST

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

Creatio поддерживает интеграцию с пользовательскими REST-сервисами с помощью инструментов low-

code. Creatio сгенерирует и отправит запрос веб-сервису, получит ответ и предоставит необходимые данные. Данные, полученные от веб-сервиса, можно использовать для создания или обновления записей в Creatio, а также для реализации пользовательской бизнес-логики или автоматизации.

Общая последовательность настройки одинакова для всех REST-сервисов, а детали во многом зависят от специфики веб-сервиса.

**Этапы** настройки интеграции с веб-сервисом:

1. [Добавление веб-сервиса и настройка](#) его свойств и методов.
2. Настройка аутентификации веб-сервиса (опциональный шаг). [Настройка аутентификации](#) идентична для REST и SOAP-сервисов.
3. [Проверка настройки интеграции с веб-сервисом](#).

После этого можно использовать интеграцию с веб-сервисом в пользовательских бизнес-процессах. Например, можно настроить интеграцию с веб-сервисом “Fixer”, чтобы данные в справочнике [ *Валюты* ] обновлялись в рамках бизнес-процесса. Подробнее данный пример описан в статье [Использовать интеграцию с веб-сервисами для автоматического обновления курсов валют](#).

**Пример.** Настроить интеграцию с REST-сервисом “Fixer” (<https://fixer.io>) для получения актуальных курсов обмена валют по отношению к евро.

В этом примере необходимо реализовать поддержку следующих параметров:

- **“endpoint”** — параметр в адресе метода. Возвращает фактические значения курса валют в режиме реального времени.
- **“access\_key”** — параметр запроса. В него необходимо передать ключ доступа к сервису.
- **“symbols”** — параметр запроса. В него необходимо передать коды валют. Допускается использование нескольких кодов, указанных через запятую.
- **“\$.base”** — параметр ответа. Содержит значение базовой валюты, в отношении которой были рассчитаны курсы валют. Базовой валютой является евро.
- **“\$.date”** — параметр ответа. Содержит информацию о дате актуальности курсов валют.
- **“rates”** — параметр ответа. Содержит коллекцию курсов обмена валют по отношению к базовой валюте (каждый курс обмена представлен отдельным параметром, название которого соответствует коду валюты).

## Настроить свойства и методы веб-сервиса

Перед началом работы с REST-сервисом “Fixer” необходимо зарегистрироваться на сайте <https://fixer.io/> и получить API ключ для доступа к сервису.

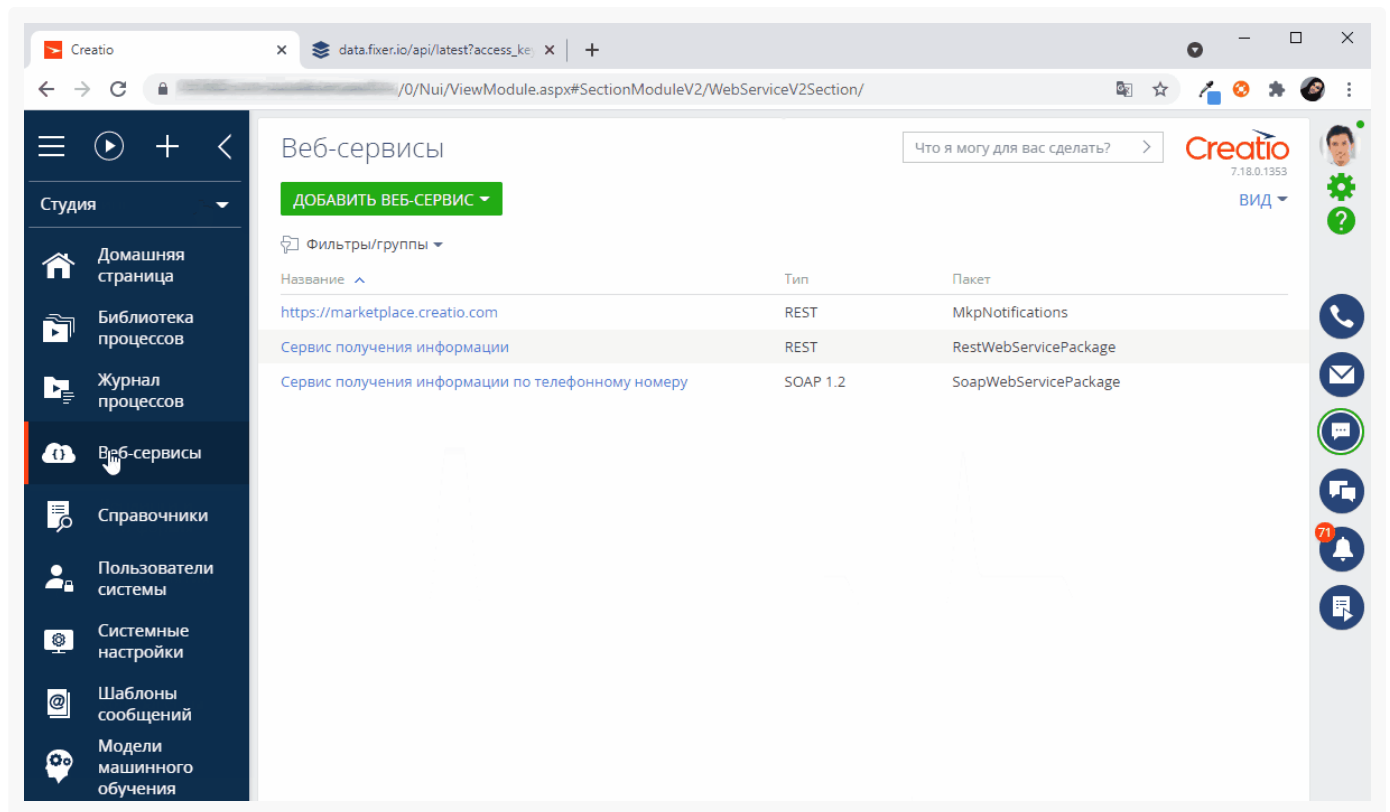
Creatio предоставляет возможность настроить свойства и методы веб-сервиса [автоматически](#) или [вручную](#).

## Автоматическая настройка свойств и методов веб-сервиса

1. Перейдите в рабочее место [ *Студия* ] и откройте раздел [ *Веб-сервисы* ].

2. Нажмите [ *Добавить веб-сервис* ] —> [ *REST сервис* ].
3. Укажите адрес веб-сервиса и нажмите [ *OK* ] (Рис. 1).

Рис. 1 — Автоматическая настройка методов веб-сервиса



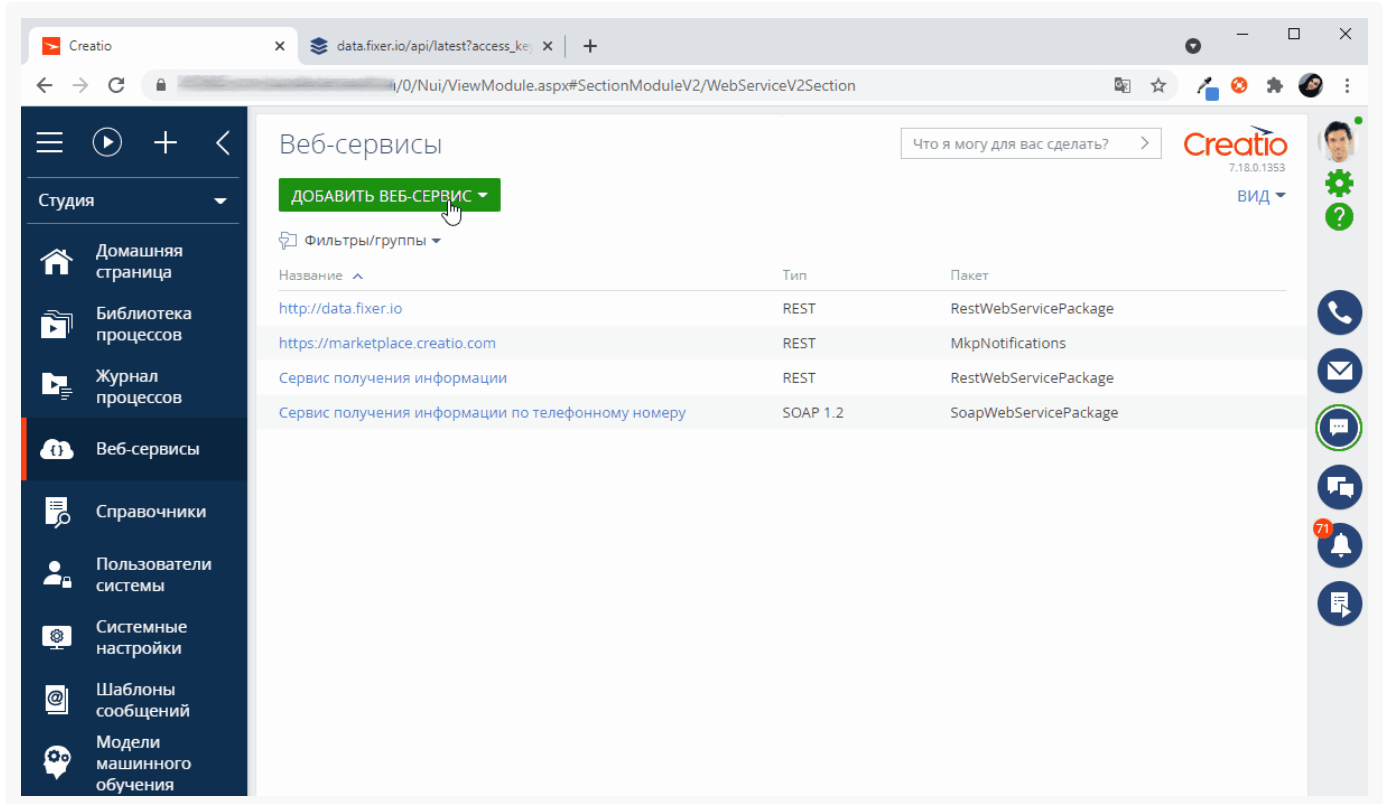
## Ручная настройка свойств и методов веб-сервиса

1. Перейдите в рабочее место [ *Студия* ] и откройте раздел [ *Веб-сервисы* ].
2. Нажмите [ *Добавить веб-сервис* ] —> [ *REST сервис* ].
3. Заполните **поля страницы свойств** веб-сервиса (Рис. 2).



Поле	Комментарий	Пример
<b>Название</b>	Название будет отображаться в поле [ <i>Какой сервис вызывать?</i> ] области свойств элемента процесса [ <i>Вызвать веб-сервис</i> ].	Курсы обмена валют (Fixer)
<b>Код</b>	Используется разработчиками для взаимодействия с веб-сервисом в программном коде Creatio. В данном случае уникальное имя интеграции с веб-сервисом состоит из его названия и префикса “Usr”.	UsrFixer
<b>URI сервиса</b>	Полный адрес вызова веб-сервиса будет состоять из этого URI и настроек, указанных на странице настройки метода. Используйте такой же протокол (http/https), как и у сайта вашего приложения Creatio. Если веб-сервис содержится в недоступном для редактирования пакете, то его URI будет доступен для редактирования.	http://data.fixer.io
<b>Повторов вызова при ошибках</b>	Если ответ от веб-сервиса пришел с кодом ошибки или истек тайм-аут ответа, то запрос будет повторен указанное количество раз. При заполнении этого поля учитывайте тайм-аут ответа, который будет указан для методов веб-сервиса.	10
<b>Тип</b>	Тип веб-сервиса.	Значение по умолчанию — REST
<b>Пакет</b>	Пакет, в котором будет сохранена данная интеграция с веб-сервисом. В списке отображаются пакеты, которые доступны для изменения текущим пользователем.	RestWebServicePackage

Рис. 2 — Заполнение страницы свойств веб-сервиса



**На заметку.** Интеграции с веб-сервисами сохраняются в виде элементов конфигурации. Если такой элемент конфигурации находится в пакете, который недоступен для редактирования, то вы сможете изменить только URI веб-сервиса. Для внесения других изменений (например, для добавления методов) в такие интеграции с веб-сервисами скопируйте соответствующие элементы конфигурации в доступный для редактирования пакет.

Для каждого веб-сервиса необходимо настроить вызов методов. При этом для одного веб-сервиса можно настроить несколько методов.

**На заметку.** В рамках одного процесса можно вызывать любое количество методов, используя несколько [элементов процесса](#) [ [Вызвать веб-сервис](#) ], с передачей значений параметров между ними.

- На детали [ *Методы* ] страницы настройки интеграции с веб-сервисом при помощи кнопки **+** добавьте методы.
- Изучите [документацию веб-сервиса](#) и заполните **свойства метода** (Рис. 3).

Поле	Комментарий	Пример
<b>Название</b>	Название будет отображаться в поле [ <i>Какой сервис вызывать?</i> ] области свойств элемента процесса [ <i>Вызвать веб-сервис</i> ].	Получить курсы обмена

Код (на Поле Английском)	Используется разработчиками для взаимодействия с веб-сервисом в Комментарий	UsrLatest Пример
	программном коде Creatio. В данном случае уникальное имя интеграции с веб-сервисом состоит из его названия и префикса "Usr".	
<b>Адрес метода</b>	<p>Для определения данного значения используется документация веб-сервиса.</p> <p>Вы можете указать статическое значение либо использовать "Параметры в адресе метода" в качестве "переменных" для формирования динамического адреса метода, который будет зависеть от свойств конкретного элемента процесса [ <i>Вызвать веб-сервис</i> ]. Например, у веб-сервиса "Fixer" есть конечная точка "latest", которая возвращает актуальные курсы валют.</p> <p>Для более гибкой интеграции можно передать конечную точку в качестве параметра адреса метода, указав имя параметра в фигурных скобках. В данном примере мы используем переменную "{endpoint}". Можно использовать несколько переменных, например: "{parameter1}/{parameter2}".</p>	<pre>/{endpoint}?access_key= 6f*****53 &amp;symbols=USD,GBP,UAH,RUB</pre>
<b>Полный адрес</b>	<p>Полный адрес генерируется автоматически. Он состоит из URI веб-сервиса и адреса метода с указанием параметров и значений параметров. Используется формат "?paramCode1=value1&amp;paramCode2=value2".</p> <p>Полный адрес отображается для проверки настроек метода.</p> <p>Структура адреса: "Web service URI" + "Адрес метода" + "?" + "Параметры вызова, разделенные символом &amp;". Например: http://data.fixer.io/latest?access_key=*****&amp;symbols=GBP</p>	<pre>http://data.fixer.io/api/ {endpoint}?access_key= 6f*****53 &amp;symbols=USD,GBP,UAH,RUB</pre>

Поле	Комментарий	Пример
	<p><code>symbols=GBP</code>.            Параметр <code>access_key</code></p> <p>используется для указания ключа доступа к веб-сервису.</p> <p>Параметр <code>"symbols"</code> используется для указания валют, курсы обмена которых следует вернуть.</p> <p>Например, для получения курсов обмена доллара США и британского фунта используется следующий запрос: <code>https://data.fixer.io/latest?symbols=USD,GBP</code>. Если параметр <code>"symbols"</code> не был передан в запросе, то будут возвращены курсы обмена всех валют, поддерживаемых веб-сервисом.</p>	
<b>Тип запроса</b>	<p>Тип HTTP-метода, используемый в запросе. Поддерживаются стандартные типы HTTP-методов. Тип запроса, как правило, указан в документации веб-сервиса.</p> <p>Например, для получения таких данных как курс валют, используются запросы типа GET.</p>	GET
<b>Тип содержимого</b>	Поддерживается только тип <a href="#">JSON</a> .	JSON
<b>Тайм-аут ответа, мс</b>	<p>Время ожидания ответа от веб-сервиса. Если после отправки запроса не был получен ответ, либо был получен код ошибки, то по истечении этого времени Creatio повторит запрос (если еще остались неиспользованные попытки повторного вызова).</p>	По умолчанию — 5000
<b>Использовать аутентификацию</b>	<p>Использовать аутентификацию для доступа к веб-сервису. Необходимо настроить аутентификацию на детали [ Аутентификация ] страницы настройки интеграции с веб-сервисом. Подробнее: <a href="#">Аутентификация веб-сервиса</a>.</p>	По умолчанию — false

Рис. 3 — Свойства метода веб-сервиса

ОК
ОТМЕНА
⚙️ БЫСТРАЯ НАСТРОЙКА ▾
ОТПРАВИТЬ ТЕСТОВЫЙ ЗАПРОС

Название*	Получить курсы обмена	Тип запроса*	GET
Код (на английском)*	UsrLatest	Тип содержимого	JSON
Адрес метода*	/[endpoint]?access_key=6f[redacted]53&symbols=USD,⚙️	Тайм-аут ответа, мс*	5 000
Полный адрес	http://data.fixer.io/[endpoint]? access_key=6f[redacted]53&symbols=USD,GBP,UAH,RUB	Использовать аутентификацию	<input type="checkbox"/>

< ПАРАМЕТРЫ ВЫЗОВА
ОБРАБОТКА ОТВЕТА
>

ДОБАВИТЬ ПАРАМЕТР
:

## Настроить параметры запроса

Параметры запроса можно добавить вручную или импортировать из примеров вызова по кнопке [ *Быстрая настройка* ], выбрав формат примера (сURL, RAW or JSON) в выпадающем меню. Также параметры вызова можно добавить из поля [ *Адрес метода* ]. В нашем примере для получения актуальных курсов обмена валют можно использовать GET-запрос с конечной точкой “/latest”.

Параметры запроса метода используются для генерации URL-адреса конечной точки, которая будет использоваться Creatio при вызове веб-сервиса.

Количество и типы параметров запроса определяются спецификой веб-сервиса. Вы можете [импортировать](#) параметры вызова из примеров или добавить их [вручную](#).

Доступны следующие **типы параметров запроса**:

<b>Параметр в адресе метода</b>	<p>Этот тип параметра используется как переменные для формирования адреса метода запроса.</p> <p>Добавьте названия параметров в адресе метода, заключив их в фигурные скобки в поле [ <i>Адрес метода</i> ]. Например: {parameterName1}/{parameterName2}, и т. д.</p> <p>При вызове веб-сервиса эти переменные будут заменены фактическими значениями параметров адреса метода, указанными для соответствующего элемента [ <a href="#">Вызвать веб-сервис</a> ] в дизайнера процессов. Например: <code>http://web.service.uri/parameterValue1/parameterValue2</code>.</p> <p>Если в свойствах элемента значение параметра не указано, но поле [ <i>Значение по умолчанию</i> ] параметра заполнено, то будет отправлено значение по умолчанию.</p> <p>Для этого типа параметра признак [ <i>Обязательный</i> ] установлен и недоступен для редактирования.</p>
<b>Параметр тела</b>	<p>Этот тип параметра используется для отправки любых типов данных (включая <a href="#">коллекции</a>) в теле запроса. Подробнее о методе POST запроса читайте в <a href="#">Википедии</a>.</p> <p>Параметр недоступен для GET-метода.</p>
<b>Параметр запроса</b>	<p>Параметр будет добавлен в запрос после адреса метода и символа “?”. Подробнее о параметре читайте в <a href="#">Википедии</a> (статья на английском языке).</p>
<b>Параметр заголовка</b>	<p>Этот тип параметра используется для генерации заголовка запроса. Подробнее об HTTP-заголовках читайте в <a href="#">Википедии</a>.</p>
<b>Параметр cookies</b>	<p>Параметры этого типа используются для передачи файлов cookie в запросах к сервису. Например, можно передать полученный ранее cookie для аутентификации. Подробнее о cookie читайте в <a href="#">Википедии</a>.</p>

## Автоматическая настройка параметров запроса


1. В правой части поля [ *Адрес метода* ] в правой части поля нажмите кнопку быстрой настройки . Тот же результат можно получить, выбрав опцию [ *По полю “Адрес метода”* ] в меню кнопки [ *Быстрая настройка* ], расположенной в верхней части окна добавления метода (Рис. 4).

Рис. 4 — Добавление параметров из поля [ *Адрес метода* ]

Настройка вызова по полю "Адрес метода"

Название\* Получить курсы обмена

Код (на английском)\* UsrLatest

Адрес метода\* /{endpoint}?access\_key=6f...53&symbols=U...

Полный адрес http://data.fixer.io/{endpoint}?access\_key=6f...53&symbols=USD,GBP,UAH,RUB

- Из открывшегося списка параметров выберите те, которые необходимо добавить в запрос, и нажмите [ Сохранить ] (Рис. 5).

Рис. 5 — Список параметров адреса метода

Настройка вызова по полю "Адрес метода"

СОХРАНИТЬ НАЗАД

Проверьте настройки метода

Адрес метода /{endpoint}

Выберите, какие параметры добавить в вызов :

Название	Тип	Значение по умолчанию
<input checked="" type="checkbox"/> T Endpoint	Параметр в адресе метода	
<input checked="" type="checkbox"/> T Access_key	Параметр запроса	
<input checked="" type="checkbox"/> T Symbols	Параметр запроса	

В результате выбранные параметры будут добавлены в запрос. Вы можете изменить настройки параметров (например, можно установить признак [ *Обязательный* ] для параметра "Access\_key"), чтобы подготовить их для дальнейшего использования на шаге бизнес-процесса [ *Вызвать веб-сервис* ].

## Ручная настройка параметров запроса

- Изучите [документацию веб-сервиса](#).
- Добавьте **параметры запроса**:
  - На вкладке [ *Параметры вызова* ] нажмите кнопку [ *Добавить параметр* ].
  - Добавьте параметр "Endpoint" и заполните его свойства (Рис. 6).

Поле	Комментарий	Пример
Название	Название параметра сервиса.	Endpoint
Тип параметра	Тип параметра сервиса.	Параметр в адресе метода
Код в запросе	Код, который используется в запросе.	endpoint
Код в Creatio	Используется разработчиками для взаимодействия с веб-сервисом в программном коде Creatio. В данном случае уникальное имя интеграции с веб-сервисом состоит из его названия и префикса “Usr”.	UsrEndpoint
Значение по умолчанию	Значение параметра по умолчанию.	Константа. Значение — “latest”

Рис. 6 — Настройка параметра “Endpoint”

ПАНЕЛЬ НАСТРОЕК: ПАРАМЕТРЫ ВЫЗОВА | ОБРАБОТКА ОТВЕТА

ДОБАВИТЬ ПАРАМЕТР

Название	Тип	Значение по умолчанию
T Endpoint	Параметр в адресе метода	latest

Настройка параметра:

- Название\*: Endpoint
- Тип параметра\*: Параметр в адресе метода
- Код в адресе\*: endpoint
- Код в Creatio\*: UsrEndpoint
- Обязательный: ☐
- Значение по умолчанию: Константа ▼ latest

**На заметку.** Системные настройки, используемые при настройке параметров метода, привязываются к пакету веб-сервиса. Это упрощает перенос веб-сервиса в другую [рабочую среду](#).

с. Добавьте параметр “API ключ” и заполните его свойства (Рис. 7).



Поле	Комментарий	Пример
<b>Название</b>	Название параметра сервиса.	API ключ
<b>Тип параметра</b>	Тип параметра сервиса.	Параметр запроса
<b>Код в запросе</b>	Код, который используется в запросе.	access_key
<b>Код в Creatio</b>	Используется разработчиками для взаимодействия с веб-сервисом в программном коде Creatio. В данном случае уникальное имя интеграции с веб-сервисом состоит из его названия и префикса "Usr".	UsrAccessKey
<b>Обязательный</b>	При установленном признаке параметр будет обязательным в дизайнера процессов. Признак недоступен для редактирования при выборе значения по умолчанию.	Признак установлен
<b>Значение по умолчанию</b>	Значение параметра по умолчанию.	Константа

Рис. 7 — Настройка параметра "API ключ"

ПАНЕЛЬ НАВИГАЦИИ: ПАРАМЕТРЫ ВЫЗОВА (активно), ОБРАБОТКА ОТВЕТА

ДОБАВИТЬ ПАРАМЕТР

Название	Тип	Значение по умолчанию
Endpoint	Параметр в адресе метода	latest
API ключ	Параметр запроса	

Настройка параметра "API ключ":

- Название\*: API ключ
- Тип параметра\*: Параметр запроса
- Код в запросе\*: access\_key
- Код в Creatio\*: UsrAccesskey
- Обязательный: ☒
- Значение по умолчанию: Константа

d. Добавьте параметр "Валюты" и заполните его свойства (Рис. 8).

Поле	Комментарий	Пример
<b>Название</b>	Название параметра сервиса.	Валюты
<b>Тип параметра</b>	Тип параметра сервиса.	Параметр запроса
<b>Код в запросе</b>	Код, который используется в запросе.	symbols
<b>Код в Creatio</b>	Используется разработчиками для взаимодействия с веб-сервисом в программном коде Creatio. В данном случае уникальное имя интеграции с веб-сервисом состоит из его названия и префикса "Usr".	UsrSymbols
<b>Обязательный</b>	При установленном признаке параметр будет обязательным в дизайнерах процессов. Признак недоступен для редактирования при выборе значения по умолчанию.	По умолчанию признак снят
<b>Значение по умолчанию</b>	Значение параметра по умолчанию.	Константа

Рис. 8 — Настройка параметра "Валюты"

ПАРАМЕТРЫ ВЫЗОВА    ОБРАБОТКА ОТВЕТА

**ДОБАВИТЬ ПАРАМЕТР** :

Название	Тип	Значение по умолчанию
Endpoint	Параметр в адресе метода	latest
API ключ	Параметр запроса	
<b>Валюты</b>	Параметр запроса	

Название\* Валюты

Тип параметра\* Параметр запроса

Код в запросе\* symbols

Код в Creatio\* UsrSymbols

Обязательный ☐

Значение по умолчанию Константа

В результате Creatio сможет вызывать веб-сервис, используя запросы, автоматически генерируемые бизнес-процессами на основании шаблона:

```
http://data.fixer.io/latest?access_key={значение}&symbols={значение}
```

Значения в фигурных скобках будут определяться значениями параметров запроса, например:

```
http://data.fixer.io/latest?access_key=6f*****53&symbols=USD
```

## Настроить параметры ответа

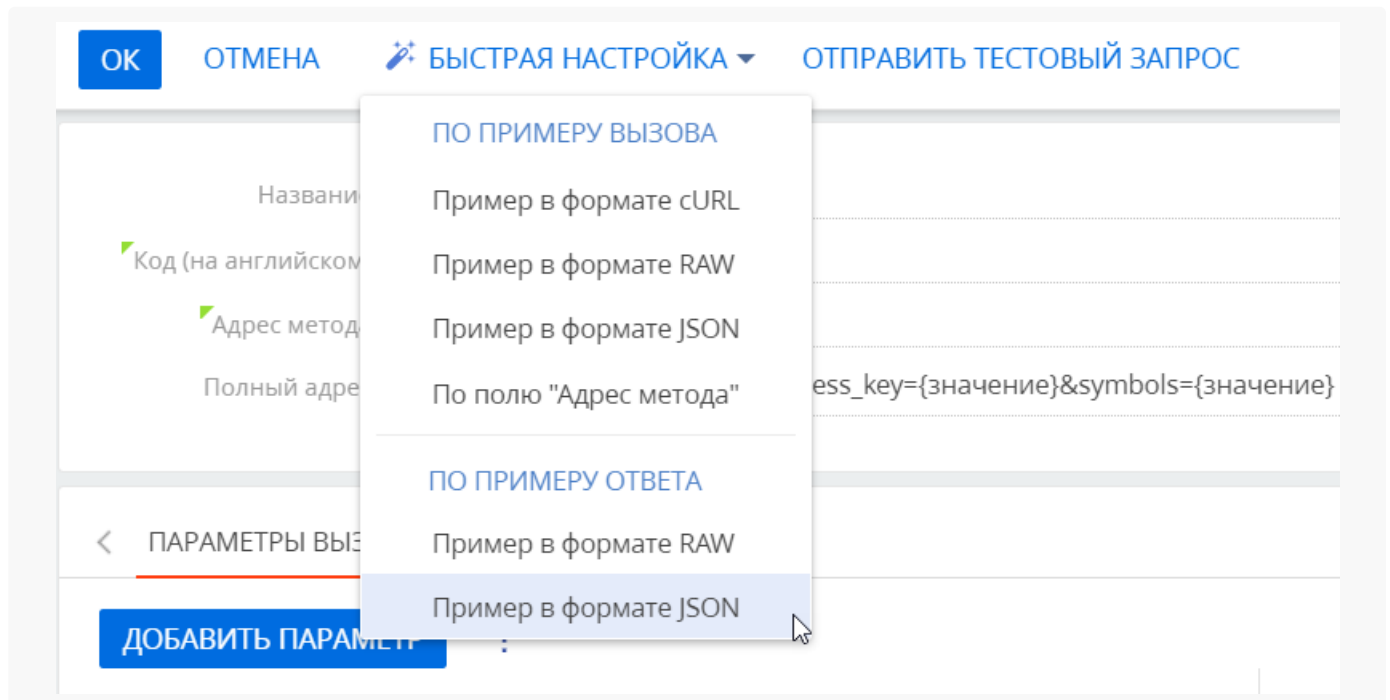
Ответ от веб-сервиса может приходить в форматах cURL, RAW, JSON. Для использования данных, полученных в ответе веб-сервиса, необходимо настроить обработку и передачу данных из ответа в параметры обработки ответа. Вы можете добавить параметры ответа [автоматически](#) или [вручную](#).

### Автоматическая настройка параметров ответа

Если ответ от сервера приходит в формате JSON или RAW, то параметры ответа можно добавить автоматически по кнопке [ *Быстрая настройка* ].

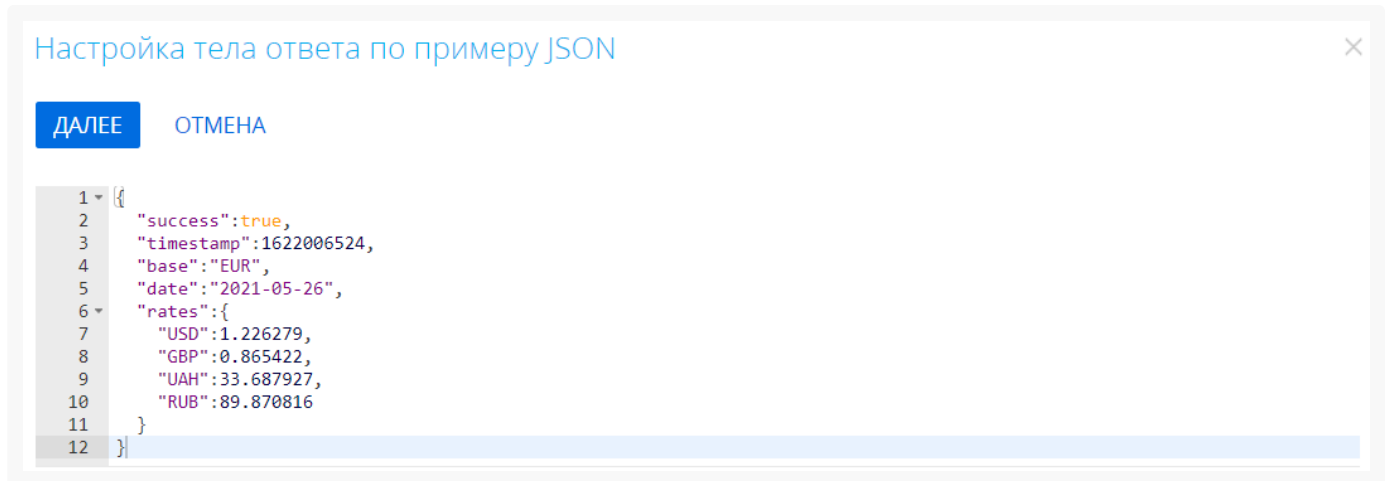
1. Нажмите кнопку [ *Быстрая настройка* ]. В блоке меню “По примеру ответа” выберите [ *Пример в формате JSON* ] (Рис. 9).

Рис. 9 — Настройка параметров ответа по примеру в формате JSON



2. Введите JSON-код ответа сервера (Рис. 10). Код отобразится в браузере, если вы попытаетесь перейти по URL-адресу вызова веб-сервиса.

Рис. 10 — Ответ в формате JSON



Нажмите кнопку [ *Далее* ].

3. Откроется список параметров:

- **"Success"** — указывает, был ли запрос успешным.
- **"Timestamp"** — объект, содержащий стандартную для UNIX отметку времени, которая указывает на время сбора данных, используемых в работе веб-сервиса.
- **"Base"** — указывает базовую валюту, по отношению к которой были рассчитаны курсы обмена валют. Например, **"base":"EUR"** в ответе указывает на то, что курсы валют были рассчитаны по отношению к евро.
- **"Date"** — содержит дату актуальности возвращенных курсов. Например, **"date":"2021-05-26"** в ответе указывает на то, что курсы валют были рассчитаны на 26 мая 2021 года.
- **"Rates"** — содержит массив вложенных параметров, каждый из которых содержит курс обмена определенной валюты по отношению к базовой валюте.  
Например, **"rates":{"USD":1.222845, "GBP":0.862279, "UAH":33.550002, "RUB":89.876838}** в ответе указывает на то, что сервисом были возвращены курсы обмена доллара США (**"USD":1.222845**), британского фунта (**"GBP":0.862279**), украинской гривны (**"UAH":33.550002**) и русского рубля (**"RUB":89.876838**).

Выберите параметры, которые необходимо добавить в ответ, и нажмите [ *Сохранить* ] (Рис. 11).

Рис. 11 — Список параметров из примера в формате JSON

### Настройка тела ответа по примеру JSON

СОХРАНИТЬ
НАЗАД

Выберите, какие параметры добавить в ответ :

Название	Тип	Значение по умолчанию
<input type="checkbox"/> <input checked="" type="checkbox"/> Success	Параметр тела	
<input type="checkbox"/> 123 Timestamp	Параметр тела	
<input checked="" type="checkbox"/> T Base	Параметр тела	
<input checked="" type="checkbox"/> T Date	Параметр тела	
<input checked="" type="checkbox"/> 123 Rates USD	Параметр тела	
<input checked="" type="checkbox"/> 123 Rates GBP	Параметр тела	
<input checked="" type="checkbox"/> 123 Rates UAH	Параметр тела	
<input checked="" type="checkbox"/> 123 Rates RUB	Параметр тела	

В результате выбранные параметры будут добавлены в ответ. Вы можете изменить настройки параметров (например, можно изменить тип данных для курса валют на “Дробное число”), чтобы подготовить их для дальнейшего использования.

## Ручная настройка параметров ответа

- Изучите [документацию веб-сервиса](#).
- Добавьте **параметры обработки ответа**:
  - На вкладке [ *Обработка ответа* ] нажмите кнопку [ *Добавить параметр* ].
  - Добавьте параметр “Базовая валюта” и заполните его свойства (Рис. 12).

Поле	Комментарий	Пример
<b>Название</b>	Название параметра сервиса.	Базовая валюта
<b>Тип параметра</b>	Тип параметра сервиса.	Параметр тела
<b>Путь к элементу (JSONPath)</b>	Значение в формате JSON. Например, JSONPath "\$.base" получает значение параметра "base" в ответе от веб-сервиса.	\$.base
<b>Код в Creatio</b>	Используется разработчиками для взаимодействия с веб-сервисом в программном коде Creatio. В данном случае уникальное имя интеграции с веб-сервисом состоит из его названия и префикса "Usr".	UsrBaseCurrency
<b>Тип данных</b>	Тип данных параметров сервиса. Параметр с вложенными элементами должен иметь тип данных "Объект".	Текст
<b>Является массивом</b>	Значение параметра-массива нельзя задать в элементе "Вызвать веб-сервис" в дизайнера процессов. Необходимо использовать элемент "Задание-сценарий". Параметр с типом данных "Объект" обязательно должен быть массивом.	По умолчанию признак снят
<b>Значение по умолчанию</b>	Значение параметра по умолчанию.	Константа. Значение — "EUR"

Рис. 12 — Настройка параметра обработки "Базовая валюта"

ПАНЕЛЬ НАСТРОЕК ПАРАМЕТРА

ПАРАМЕТРЫ ВЫЗОВА    ОБРАБОТКА ОТВЕТА

ДОБАВИТЬ ПАРАМЕТР

Название	Тип	Значение по умолчанию
T Базовая валюта	Параметр тела	EUR

Название\* Базовая валюта

Тип параметра\* Параметр тела

Путь к элементу (JSONPath)\* \$.base

Код в Creatio\* UsrBaseCurrency

Тип данных\* Текст

Является массивом ☐

Значение по умолчанию Константа ▼ EUR

с. Добавьте параметр "Дата" и укажите его свойства (Рис. 13).

Поле	Комментарий	Пример
<b>Название</b>	Название параметра сервиса.	Дата
<b>Тип параметра</b>	Тип параметра сервиса.	Параметр тела
<b>Путь к элементу (JSONPath)</b>	Значение в формате JSON. Например, JSONPath “\$.date” получает значение параметра “ <b>date</b> ” в ответе от веб-сервиса.	\$.date
<b>Код в Creatio</b>	Используется разработчиками для взаимодействия с веб-сервисом в программном коде Creatio. В данном случае уникальное имя интеграции с веб-сервисом состоит из его названия и префикса “Usr”.	UsrDate
<b>Тип данных</b>	Тип данных параметров сервиса. Параметр с вложенными элементами должен иметь тип данных “Объект”.	Дата
<b>Является массивом</b>	Значение параметра-массива нельзя задать в элементе “Вызвать веб-сервис” в дизайнера процессов. Необходимо использовать элемент “Задание-сценарий”. Параметр с типом данных “Объект” обязательно должен быть массивом.	По умолчанию признак снят
<b>Значение по умолчанию</b>	Значение параметра по умолчанию.	Константа

Рис. 13 — Настройка параметра “Дата”

ПАРАМЕТРЫ ВЫЗОВА

ОБРАБОТКА ОТВЕТА

ДОБАВИТЬ ПАРАМЕТР

Название	Тип	Значение по умолчанию
T Базовая валюта	Параметр тела	EUR
Дата	Параметр тела	

Название\*

Дата

Тип параметра\*

Параметр тела

Путь к элементу (JSONPath)\*

\$.date

Код в Creatio\*

UsrDate

Тип данных\*

Дата

Является массивом

☐

Значение по умолчанию

Константа ▼

d. Добавьте параметр “RatesUSD” и заполните его свойства (Рис. 14).

Поле	Комментарий	Пример
<b>Название</b>	Название параметра сервиса.	Rates USD
<b>Тип параметра</b>	Тип параметра сервиса.	Параметр тела
<b>Путь к элементу (JSONPath)</b>	Значение в формате JSON. Например, JSONPath "\$.rates.USD" получает значение параметра "USD" в ответе от веб-сервиса.	\$.rates.USD
<b>Код в Creatio</b>	Используется разработчиками для взаимодействия с веб-сервисом в программном коде Creatio. В данном случае уникальное имя интеграции с веб-сервисом состоит из его названия и префикса "Usr".	UsrRatesUSD
<b>Тип данных</b>	Тип данных параметров сервиса. Параметр с вложенными элементами должен иметь тип данных "Объект".	Дробное число
<b>Является массивом</b>	Значение параметра-массива нельзя задать в элементе "Вызвать веб-сервис" в дизайнера процессов. Необходимо использовать элемент "Задание-сценарий". Параметр с типом данных "Объект" обязательно должен быть массивом.	По умолчанию признак снят
<b>Значение по умолчанию</b>	Значение параметра по умолчанию.	Константа

Рис. 14 — Настройка параметра "Rates USD"

ПАРАМЕТРЫ ВЫЗОВА

ОБРАБОТКА ОТВЕТА

ДОБАВИТЬ ПАРАМЕТР

⋮

Название	Тип	Значение по умолчанию
Базовая валюта	Параметр тела	EUR
Дата	Параметр тела	
Rates USD	Параметр тела	

Название\* Rates USD

Тип параметра\* Параметр тела

Путь к элементу (JSONPath)\* \$.rates.USD

Код в Creatio\* UsrRatesUSD

Тип данных\* Дробное число

Является массивом ☐

Значение по умолчанию Константа ▼

**На заметку.** JSONpath "\$.rates.USD" получает значение параметра "USD". "USD" — вложенный



параметр параметра “rates” ответа веб-сервиса.

- е. Аналогичным образом настройте параметры обработки ответа для курсов остальных необходимых валют (Рис. 15).

Рис. 15 — Настройка параметров обработки ответа веб-сервиса

Название	Тип	Значение по умолчанию
Базовая валюта	Параметр тела	EUR
Дата	Параметр тела	
0.5 Rates USD	Параметр тела	
0.5 Rates GBP	Параметр тела	
0.5 Rates UAH	Параметр тела	
0.5 Rates RUB	Параметр тела	

Название*	Базовая валюта
Тип параметра*	Параметр тела
Путь к элементу (JSONPath)*	\$.base
Код в Creatio*	UsrBaseCurrency
Тип данных*	Текст
Является массивом	<input type="checkbox"/>
Значение по умолчанию	Константа ▼ EUR

3. Нажмите кнопку [ *Ок* ] для сохранения настроек.

## Настроить параметры запроса и ответа типа “коллекция”

Коллекцией (или массивом) является набор элементов. Creatio может передавать коллекции данных в веб-сервис и обрабатывать его ответы, содержащие коллекции. Если веб-сервис поддерживает получение и/или отправку массивов данных, то параметры типа “коллекция” можно использовать как для вызова веб-сервиса, так и для обработки его ответа.

### Типы параметров коллекции:

- **Простая коллекция.** Любой параметр можно представить в виде коллекции, установив в свойствах параметров признак “Является массивом”. Простые коллекции являются массивами значений одного типа данных. Каждое значение является отдельным элементом коллекции. Например, “1, 2, 3” — это простой массив значений целых чисел, а “Бостон, Нью-Йорк, Чикаго” — простой массив текстовых значений.
- **Коллекция объекта.** Коллекция представляет собой корневой параметр (т. н. объект), который содержит вложенные параметры. Каждый элемент коллекции объекта может иметь разные типы параметров. Такие параметры элемента коллекции являются вложенными параметрами корневого параметра. Например, коллекция контактов может содержать вложенные параметры для передачи имени контакта, даты рождения и возраста (Рис. 16).

Рис. 16 — Пример коллекции контактов

```

graph TD
    Root[Коллекция контактов] --> Child1[ФИО контакта]
    Root --> Child2[Дата рождения]
    Root --> Child3[Возраст]
  
```

Коллекция такой структуры выглядит так:

ФИО контакта	Дата рождения	Возраст
Евгений Мирный	12.04.1991	30
Сергей Авдоров	10.04.1985	36
Ольга Варенская	12.05.1989	31

Creatio может передавать коллекции данных в веб-сервис и обрабатывать ответы веб-сервиса, содержащие коллекции. Если веб-сервис поддерживает получение и/или отправку массивов данных, то параметры типа “коллекция” можно использовать как для вызова веб-сервиса, так и для обработки его ответа.

**На заметку.** Для параметров типа “Параметр тела” вы можете выбрать только тип данных “коллекция”. Этот тип параметров недоступен для методов с запросами типа GET. Тем не менее, он доступен для ответов типа GET.

Для настройки коллекции:

1. Изучите [документацию веб-сервиса](#).
2. Добавьте метод “Получение контактов”.
3. Для настройки параметров запроса, который является коллекцией, в поле [ *Тип данных* ] укажите “Объект”. Признак [ *Является массивом* ] будет установлен автоматически (Рис. 17).

Рис. 17 — Добавление параметра в коллекцию

The screenshot shows the 'ПАРАМЕТРЫ ВЫЗОВА' (Parameters of Call) tab in the Creatio interface. A modal window is open for adding a parameter. The parameter name is 'Записи' (Records), type is 'Параметр тела' (Body parameter). The JSONPath is '\$.records', the Creatio code is 'UsrRecords', and the data type is 'Объект' (Object). The 'Is array' checkbox is checked. The modal has 'OK' and 'ОТМЕНА' (Cancel) buttons.

В результате параметр ответа веб-сервиса типа “коллекция”, который получает записи контактов с сервера, будет иметь такой вид (Рис. 18).

Рис. 18 — Настройка параметров запроса

ОК
ОТМЕНА
⚙️ БЫСТРАЯ НАСТРОЙКА ▾
ОТПРАВИТЬ ТЕСТОВЫЙ ЗАПРОС

Название\*

Код (на английском)\*

Адрес метода\*

Полный адрес

Тип запроса\*




Тип содержимого

Тайм-аут ответа, мс\*

Использовать аутентификацию ☐

< ПАРАМЕТРЫ ВЫЗОВА
ОБРАБОТКА ОТВЕТА

ДОБАВИТЬ ПАРАМЕТР ▾
:

Название	Тип	Значение по умолчанию
—  Записи	Параметр тела	
 Имя	Параметр тела	
 Email	Параметр тела	

Название\*

Тип параметра\*

Путь к элементу (JSONPath)\*

Код в Creatio\*

Тип данных\*

Является массивом ☒

**На заметку.** Значение JSONPath для вложенных параметров указывается как суффикс JSONPath корневого параметра. Например, если в поле [ *Путь к элементу (JSONPath)* ] корневого параметра указано значение “\$.records”, а значение “\$.records.name” должно быть указано для вложенного параметра “Имя”, то в поле [ *Путь к элементу (JSONPath)* ] вложенного параметра достаточно указать “\$.records”.

Параметры ответа веб-сервиса типа “коллекция” могут использоваться в качестве входящих параметров элемента бизнес-процесса [ *Вызвать веб-сервис* ]. Подробнее: [Элемент процесса \[ Вызвать веб-сервис \]](#).

## Проверить настройку интеграции с REST-сервисом

Для дальнейшего использования данных от сервиса в пользовательских бизнес-задачах необходимо проверить настройку интеграции.

Начиная с версии 7.18.0 появилась возможность проверить настройку интеграции с SOAP-сервисом из интерфейса приложения, без создания тестового бизнес-процесса.

Чтобы **проверить настройку интеграции с REST-сервисом**:

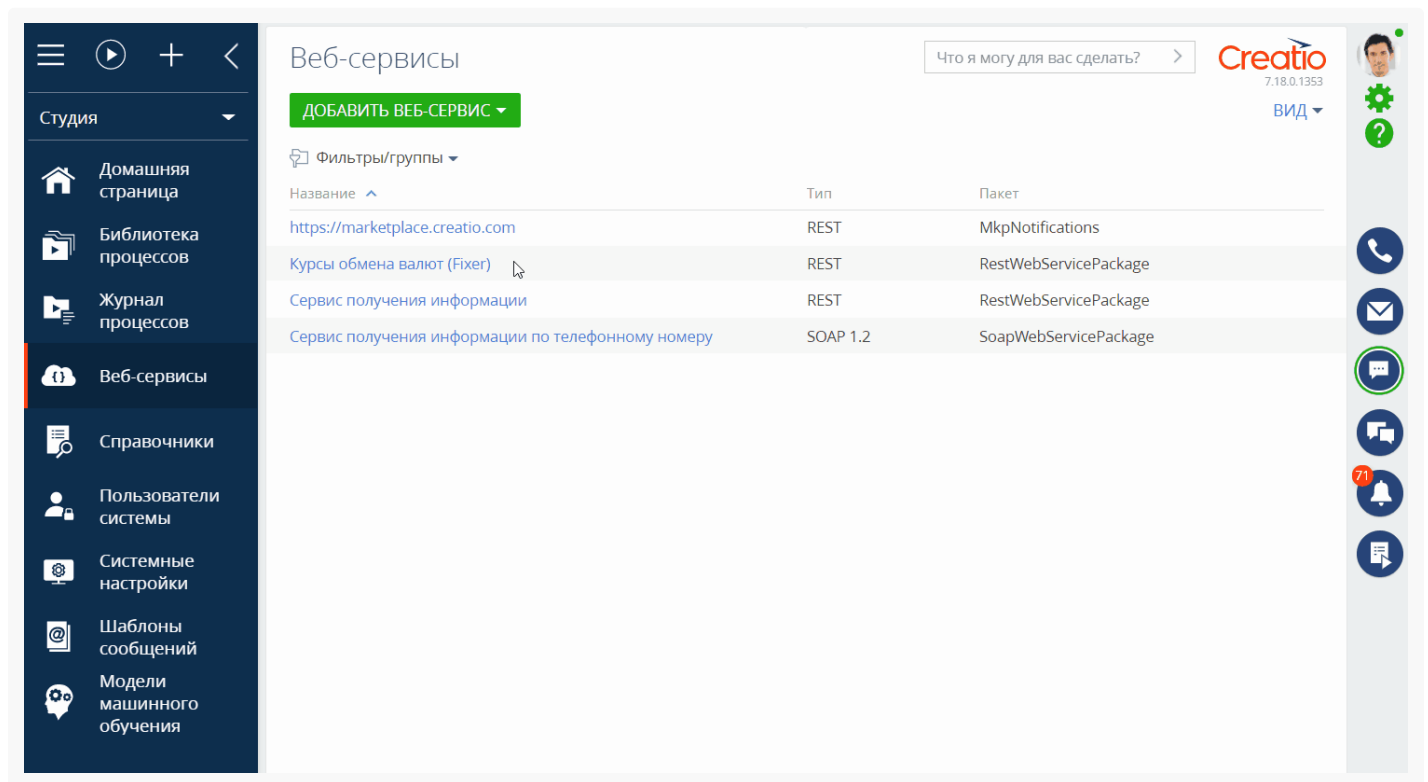
1. Перейдите в рабочее место [ *Студия* ] и откройте раздел [ *Веб-сервисы* ].
2. Откройте страницу веб-сервиса, интеграцию с которым необходимо проверить. В нашем примере это “Курсы обмена валют (Fixer)”.
3. На детали [ *Методы* ] страницы настройки интеграции с веб-сервисом откройте метод “Получить курсы обмена”.
4. На странице настройки метода нажмите [ *Отправить тестовый запрос* ].
5. Задайте значения параметров вызова метода.

Параметр	Значение
Endpoint	latest
API ключ	6f*****53
Валюты	USD,GBP,UAH,RUB

6. Нажмите [ *Отправить запрос* ] (Рис. 19).

В результате выполнения тестового запроса будет получен ответ от веб-сервиса. Значения параметров ответа отображены на вкладке [ *Параметры ответа* ] в соответствующих полях (Рис. 19). Другие вкладки позволяют посмотреть запрос и ответ на запрос на языках JSON и HTTP (формат данных raw).

Рис. 19 — Проверка интеграции с REST-сервисом



Если при настройке интеграции с веб-сервисом была допущена ошибка, то в результате тестового запроса будет получен ответ с пустыми значениями параметров. В этом случае проверьте все настройки сервиса и повторите попытку.

После проверки вы можете использовать интеграцию с веб-сервисом в пользовательских бизнес-процессах. Подробнее данный пример описан в статье [Использовать интеграцию с веб-сервисами для автоматического обновления курсов валют](#).

В зависимости от особенностей веб-сервиса, прежде чем он станет доступным для использования, может потребоваться выполнить процедуру аутентификации. Подробнее: [Настроить аутентификацию веб-сервиса](#).

# Настроить аутентификацию веб-сервиса

ПРОДУКТЫ: [ВСЕ ПРОДУКТЫ](#)

В зависимости от особенностей веб-сервиса, прежде чем он станет доступным для использования, может потребоваться выполнить процедуру аутентификации. При интеграции Creatio с веб-сервисами поддерживается **два типа аутентификации**:

- [Аутентификация OAuth 2.0](#) — безопасный способ аутентификации, поддерживается большинством популярных провайдеров, включая Facebook, Google и Amazon.
- [Базовая HTTP-аутентификация](#) — заключается в передаче логина и пароля в рамках вызова веб-сервиса. Выполняется с целью проверки учетных данных, переданных из Creatio в ходе интеграции с веб-сервисом. Это более простая форма аутентификации, но менее надежная.

## Настроить базовую аутентификацию

Для включения базовой аутентификации:

1. Получите учетные данные для выполнения базовой аутентификации. Способ получения зависит от специфики интегрируемого веб-сервиса.
2. Перейдите в раздел [ *Системные настройки* ] и добавьте следующие системные настройки:
  - a. Системную настройку типа “Строка” для хранения логина доступа в веб-сервис.
  - b. Системную настройку типа “Зашифрованная строка” для хранения пароля доступа в веб-сервис.

Системные настройки, используемые в процессе аутентификации, [привязываются к пакету](#) с настройками интеграции веб-сервиса.

**Важно.** Поскольку системные настройки привязываются к пакету незаполненными, то логин и пароль не сохраняются при переносе на другую [рабочую среду](#). Поэтому после установки пакета, который содержит функциональность интеграции с веб-сервисом в рабочей среде, необходимо повторно указать значения системных настроек логина и пароля.

3. В качестве значений системных настроек укажите логин и пароль для аутентификации в веб-сервисе.
4. Откройте страницу веб-сервиса и перейдите на вкладку [ *Аутентификация* ].
5. В поле [ *Аутентификация* ] укажите “Basic”.
6. В полях [ *Имя пользователя* ] и [ *Пароль* ] выберите системные настройки, содержащие логин и пароль доступа к веб-сервису (Рис. 1).

Рис. 1— Вкладка [ *Аутентификация* ] страницы веб-сервиса

https://data.fixer.io

СОХРАНИТЬ ОТМЕНА

Название\* https://data.fixer.io

Код\* UsrService\_8129830

Описание

URI сервиса\* https://data.fixer.io

Повторов вызова при ошибках 0

Тип REST

Пакет\* Custom

< МЕТОДЫ АУТЕНТИФИКАЦИЯ >

Аутентификация\* Basic

Имя пользователя Сист. настройка Имя пользователя для...

Пароль Сист. настройка Пароль пользователя ...

7. Примените изменения по кнопке [ Сохранить ].

## Настроить OAuth-аутентификацию

### Настроить интеграцию с существующим приложением OAuth 2.0

Для выполнения OAuth-аутентификации при условии, что приложение было настроено ранее (например, при установке пакета с настройками интеграции с веб-сервисом):

1. Перейдите в рабочее место [ Студия ] —> [ Веб-сервисы ]. Откройте страницу веб-сервиса и перейдите на вкладку [ Аутентификация ].
2. В поле [ Аутентификация ] выберите “OAuth 2.0”.
3. В поле [ Приложение ] выберите предварительно настроенное OAuth-приложение (Рис. 2).

Рис. 2 — Выбор ранее настроенного OAuth-приложения

< МЕТОДЫ АУТЕНТИФИКАЦИЯ >

Аутентификация\* OAuth 2.0

Приложение G

Google API

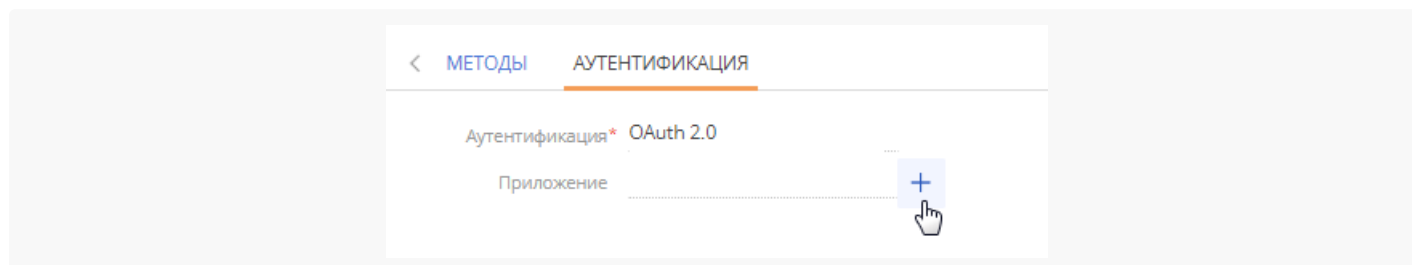
4. Примените изменения по кнопке [ Сохранить ].

## Настроить новое приложение OAuth 2.0

Для настройки нового OAuth-приложения и выполнения аутентификации с веб-сервисом:

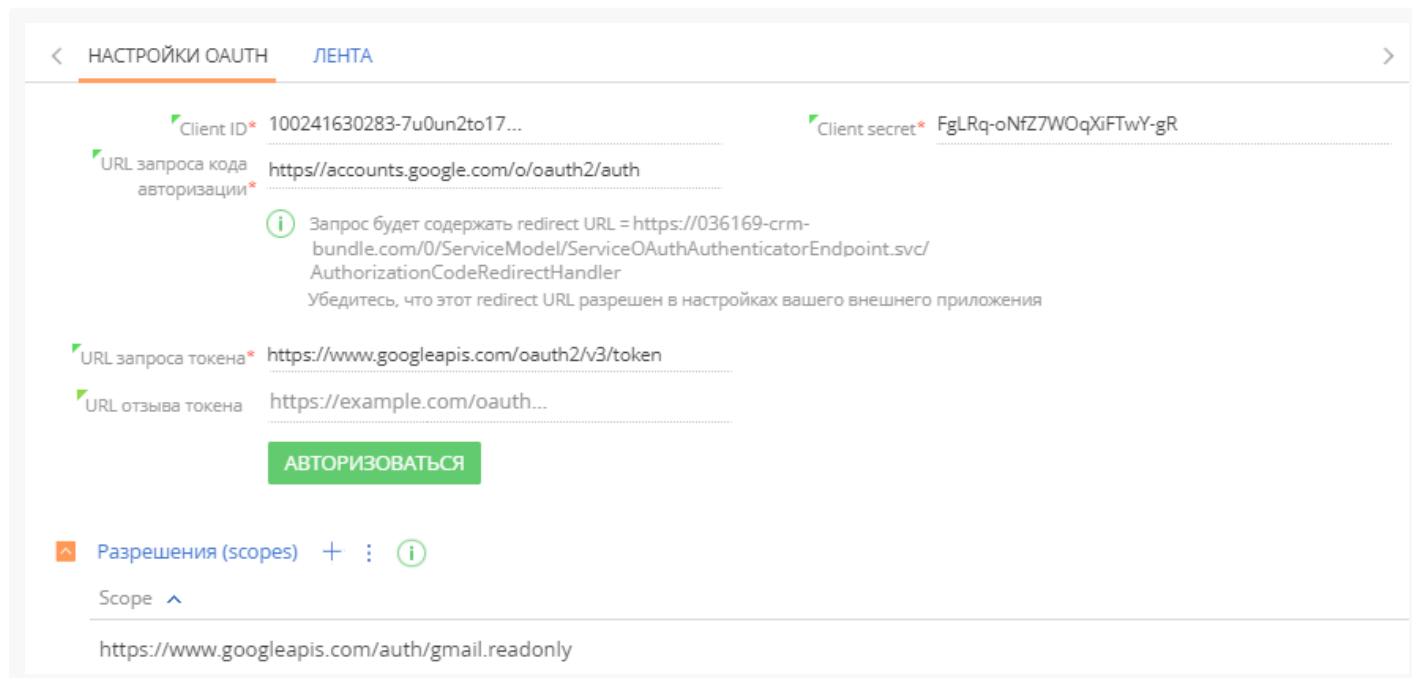
1. Перейдите в рабочее место [ *Студия* ] —> [ *Веб-сервисы* ]. Откройте страницу веб-сервиса и перейдите на вкладку [ *Аутентификация* ].
2. В поле [ *Аутентификация* ] выберите “OAuth 2.0”.
3. В поле [ *Приложение* ] нажмите [ + ] (Рис. 3).

Рис. 3 — Добавление нового OAuth-приложения



Заполните страницу настройки OAuth-приложения (Рис. 4). Необходимые значения параметров обычно можно найти в документации веб-сервиса или получить через API такого веб-сервиса. Например, данные для доступа к Google API можно найти в разделе [ *Учетные данные* ] [консоли разработчика Google](#), а также в [документации API](#).

Рис. 4 — Страница настройки OAuth-приложения



**Важно.** Настройка OAuth-аутентификации выполняется на стороне Creatio и во внешнем приложении.

Если вы используете **Creatio on-site**, то для корректной работы аутентификации необходимо, чтобы адрес веб-сервиса, вызов которого вы настраиваете, был доступен на вашем сервере.

Ниже приведены общие рекомендации по заполнению параметров настройки OAuth, а также подсказки, как получить значения таких параметров:

1. [ *Название* ] — укажите название нового приложения. Название отобразится на вкладке [ *Аутентификация* ] в поле [ *Приложение* ]. Также можно выбрать изображение, которое будет использоваться как иконка этого OAuth-приложения.
2. [ *Client ID* ] — укажите идентификатор клиента, выданный сервером авторизации интегрированного веб-сервиса. В документации и в API разных веб-сервисов идентификатор клиента также может называться:
  - Application (идентификатор приложения).
  - Consumer ID (идентификатор заказчика).
  - Public key (публичный ключ).

Указанный идентификатор будет значением параметра вызова `client_id`. Подробнее читайте в документации по [IETF Tools](#) (на английском языке). Например, Client ID для Google API можно найти в разделе [ *Учетные данные* ] консоли Google API (<https://console.developers.google.com>), в поле “Client ID”. Подробнее читайте в документации [Google](#) (на английском языке).

3. [ *Client secret* ] — укажите секретный ключ, предоставленный сервером авторизации. Также в документации и в API разных веб-сервисов секретный ключ может называться:
  - Application secret (секретный ключ приложения).
  - Consumer secret.
  - Secret key.

Указанный секретный ключ будет значением параметра вызова `client_secret`. Подробнее читайте в документации по [IETF Tools](#) (на английском языке). Например, Client secret для Google API можно найти в разделе [ *Учетные данные* ] консоли Google API (<https://console.developers.google.com>), в поле “Client secret”. Подробнее читайте в документации [Google](#) (на английском языке).

4. [ *URL запроса кода авторизации* ] — укажите URL, который будет служить конечной точкой (endpoint) для запроса доступа у пользователя, который его может предоставить. Например, когда Google информирует вас, что приложение пытается получить доступ к определенным данным, вы выступаете в роли “пользователя, который может предоставить доступ”.  
Указанный URL является значением параметра вызова `auth_url`. Подробнее читайте в документации по [IETF Tools](#) (на английском языке). Например, URL запроса кода авторизации для Google API — “<https://accounts.google.com/o/oauth2/auth>”. Его можно найти в документации, описывающей пример [интеграции OAuth-приложения с Google](#). Также URL доступен в JSON-файле с настройками, который можно загрузить в разделе “Учетные данные” консоли разработчика Google.
5. [ *URL запроса токена* ] — укажите URL-адрес, который будет использоваться в качестве конечной точки (endpoint) для запроса токена доступа при вызове веб-сервиса. Подробнее читайте в документации по [IETF Tools](#) (на английском языке). Например, URL запроса токена для Google API — “<https://www.googleapis.com/oauth2/v3/token>”. Его можно найти в документации, описывающей пример [интеграции OAuth-приложения с Google](#). Также URL доступен в JSON-файле с настройками, который можно загрузить в разделе “Учетные данные” консоли разработчика Google.
6. [ *Redirect URL* ] — укажите URL-адрес, по которому сервер аутентификации будет перенаправлять пользователей после выполнения аутентификации. “Redirect URL” отображается на странице



настройки OAuth-аутентификации (Рис. 5).

Рис. 5 — Redirect URL на странице настройки OAuth-аутентификации в Creatio

НАСТРОЙКИ OAUTH ЛЕНТА

Client ID\* 932498208625-crqvq4ss79ho7e7f... Client secret\* E7ATbJM4AuIAH6-EqbnbR\_MP

URL запроса кода авторизации\* https://accounts.google.com/o/oauth2/auth

Запрос будет содержать redirect URL = https://036905-crm-bundle.com/0/ServiceModel/ServiceOAuthAuthenticatorEndpoint.svc/AuthorizationCodeRedirectHandler. Убедитесь, что этот redirect URL разрешен в настройках вашего внешнего приложения

URL запроса токена\* https://www.googleapis.com/oauth2/token Отправлять client id и secret в запросе токена\* Как заголовок Basic авторизации

URL отзыва токена\* https://example.com/oauth2/revoke

Убедитесь, что в настройках интегрированного сервиса указан корректный URL-адрес для переадресации.

7. [ Отправлять client id и secret в запросе токена ] — выберите часть запроса токена, которая будет содержать токен доступа. Авторизация предусматривает несколько способов передачи токена. В поле [ Отправлять client id и secret в запросе токена ] доступно для выбора несколько вариантов. Вариант передачи зависит от особенностей сторонней системы и обычно рассматривается в ее документации.

- **“В теле запроса”**. Многие популярные сервисы, например, Google, LinkedIn, JIRA и т. д. обрабатывают запросы, когда токен доступа передается в теле запроса. Например:

```
POST /token HTTP/1.1
Host: server.example.com
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token&refresh_token=tGzv3J0kF0XG5Qx2TlKWIA
&client_id=s6BhdRkqt3&client_secret=7Fjfp0ZBr1KtDRbnfVdmIw
```

- **“В виде заголовка Basic-авторизации”**. Некоторые сервисы, например, такие, как [DocuSign](#), [GoToWebinar](#), требуют, чтобы идентификатор и секретный ключ клиента передавались в формате заголовка и не примут их в теле запроса. Подробнее читайте в документации [IETF](#). Например:

```
curl -X POST "https://api.getgo.com/oauth/v2/token" \
-H "Authorization: Basic {Base64 Encoded consumerKey and consumerSecret}" \
-H "Accept:application/json" \
-H "Content-Type: application/x-www-form-urlencoded" \
-d "grant_type=authorization_code&code={responseKey}&redirect_uri=http%3A%2F%2Fcode.example.com"
```

- **“В параметрах GET-запроса”**. Некоторые сервисы выполняют аутентификацию другим способом.

Например, вместо POST-запроса с параметрами, содержащимися в теле и заголовках, Facebook использует GET-запрос токена со всеми параметрами в URL запроса. Например:

```
https://developers.facebook.com/docs/facebook-login/manually-build-a-login-flow/#confirm\
GET https://graph.facebook.com/v3.2/oauth/access_token?\
client_id={app-id}\
&redirect_uri={redirect-uri}\
&client_secret={app-secret}\
&code={code-parameter}
```

8. [ *URL отзыва токена* ] — укажите URL, который будет использоваться в качестве конечной точки (endpoint) для отзыва токена доступа при отклонении соответствующих вызовов веб-сервиса. Поле не является обязательным для заполнения. Обычно пользователь, имеющий доступ к интегрированному веб-сервису, может отменить доступ к определенным приложениям OAuth. Например, вы можете ограничить доступ к определенным приложениям в настройках своей учетной записи Google: URL-адрес для запроса отзыва токена Google API — “https://accounts.google.com/o/oauth2/revoke”. Получить его можно в [документации](#) Google API.
9. [ *Авторизоваться* ] — войдите в веб-сервис по этой кнопке, используя общие учетные данные, и разрешите доступ к стороннему приложению для Creatio.
10. [ *Разрешения (scopes)* ] — настройте на детали разные уровни доступа приложения к данным от имени конечного пользователя. URL-адреса для указания на детали [ *Разрешения (scopes)* ] обычно доступны в документации веб-сервиса. Каждое API может объявлять одно или несколько разрешений. Например, [разрешения Google API](#), в частности, “https://www.googleapis.com/auth/gmail.readonly” (разрешение на просмотр email-сообщений и настроек) доступны в документации API.

Поскольку в разных сервисах используется различная терминология, а также есть отличия в структуре API и документации, то настройка аутентификации также имеет свои особенности. [Возможные проблемы настройки OAuth](#) и способы их решения описаны далее.

## Возможные проблемы настройки OAuth-приложений

Ниже рассмотрены типичные проблемы, с которыми может столкнуться пользователь при настройке интеграции с веб-сервисом с использованием аутентификации OAuth 2.0.

### Указанные настройки соединения некорректны или неактуальны

Настройки отличаются от настроек соединения в интегрированном приложении. Ошибка может быть в значениях Client ID, Client secret, а также в URL запроса кода авторизации, URL запроса токена или в URL отзыва токена.

Ошибка может возникать в следующих случаях:

- При добавлении пользователя на странице OAuth-приложения.
- При вызове веб-сервиса.

Проверьте, что все поля на странице настройки OAuth-приложения заполнены корректно. Убедитесь, что указанные значения совпадают с соответствующими настройками во внешнем приложении. После этого попробуйте добавить нового пользователя повторно.

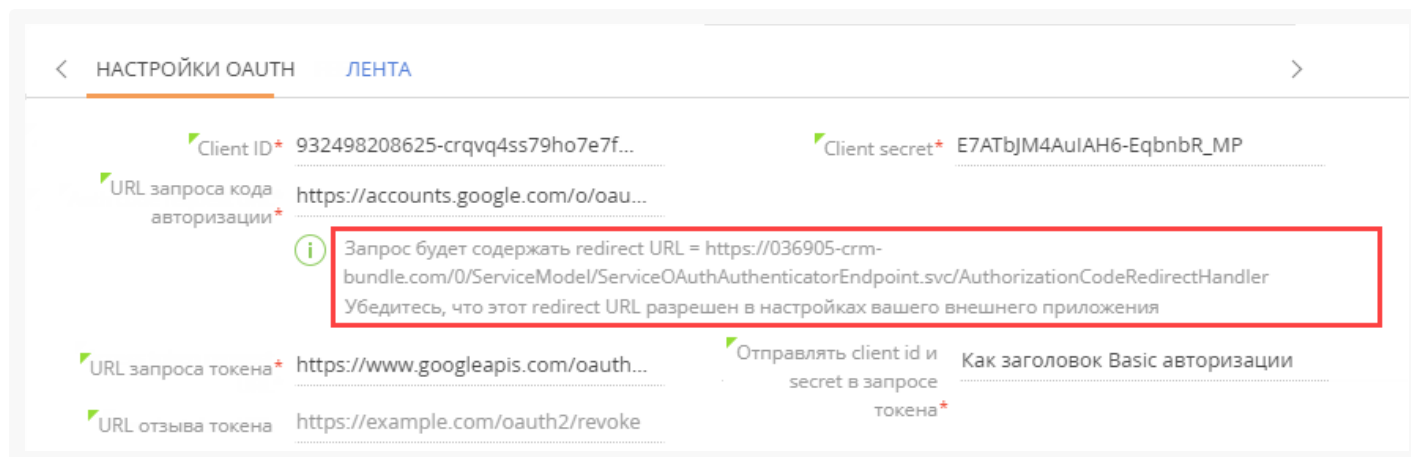
## Во внешнем приложении указан некорректный адрес перенаправления (redirect URL)

Настройка OAuth-аутентификации выполняется на стороне Creatio и во внешнем приложении. По требованиям безопасности используется специальный адрес (redirect URL) для перенаправления пользователей после получения токена доступа от сервера аутентификации. Домен, на который выполняется перенаправление, часто требует отдельной проверки.

Ошибка может возникать при добавлении пользователя на странице OAuth-приложения.

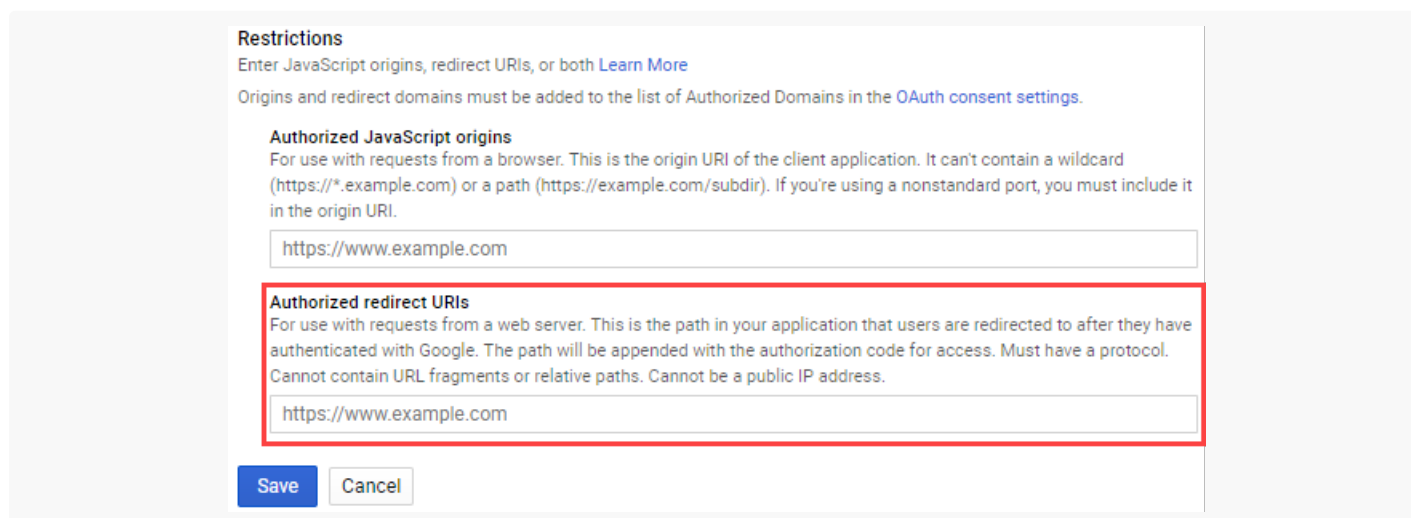
Корректный redirect URL отображается на странице настройки OAuth-приложения в Creatio (Рис. 6).

Рис. 6 — Redirect URL на странице настройки OAuth-аутентификации в Creatio



Этот же URL-адрес необходимо скопировать в настройки внешнего приложения (Рис. 7).

Рис. 7 — Добавление redirect URL в Google API



Если ссылка во внешней системе не совпадает со ссылкой, указанной на странице настройки OAuth-приложения в Creatio, то токен доступа не будет получен.

Таким образом, для решения проблемы скопируйте корректную ссылку (redirect URL) со страницы настройки OAuth-приложения в Creatio и укажите ее в соответствующем поле в настройках внешнего приложения.

## Недостаточно разрешений (scopes) на выполнение нужной операции

В OAuth 2.0 токен доступа часто выдается с определенными ограничениями к функциональности вызова внешнего приложения. Например, приложения могут требовать отдельные разрешения на чтение и изменение данных. В таком случае список необходимых разрешений необходимо указать в Creatio, во внешнем приложении или при авторизации пользователя.

В Creatio необходимые разрешения указываются на детали [ *Разрешения (scopes)* ] страницы настройки OAuth-приложения. Creatio будет запрашивать эти разрешения при выполнении пользователем OAuth-аутентификации. Если необходимое разрешение не было добавлено на деталь [ *Разрешения (scopes)* ], то, скорее всего, пользователь не сможет авторизоваться.

Ошибка может возникать, если при попытке вызвать веб-сервис, например, для изменения его данных, нет разрешения на изменение. В этом случае внешнее приложение выдаст ошибку доступа на изменение.

Для решения проблемы выполните следующее:

1. Добавьте нужный scope на деталь.
2. Удалите из Creatio пользователя веб-сервиса, чтобы ранее выданный токен доступа был отозван.
3. Отмените доступ к приложению Creatio в настройках веб-сервиса (например, вы можете отменить доступ к определенным приложениям в настройках своей учетной записи Google).
4. Добавьте пользователя повторно.

## Разрешения в Creatio не совпадают с разрешениями в настройках веб-сервиса

Иногда разрешения (scopes) должны быть указаны в интегрированном приложении. Например, уровень доступа можно установить в процессе настройки приложения в интегрированном веб-сервисе. В этом случае разрешения в Creatio должны совпадать с разрешениями, установленными в настройках приложения в интегрированном веб-сервисе.

В зависимости от архитектуры приложения, возможны следующие ошибки, если разрешения в Creatio превышают разрешения в настройках веб-сервиса:

- Пользователь не сможет авторизоваться.
- При попытке вызвать веб-сервис с целью изменения его данных внешнее приложение выдаст ошибку.

Если **пользователь не может авторизоваться**, то выполните следующие действия:

1. Добавьте необходимое разрешение в настройках внешнего приложения.
2. Повторите попытку авторизации.

Если при попытке вызвать веб-сервис для изменения данных **внешнее приложение выдает ошибку**, то выполните следующие действия:

1. Добавьте необходимое разрешение в настройках внешнего приложения.
2. Удалите пользователя веб-сервиса из Creatio, чтобы отменить ранее выданный токен доступа.
3. Отмените доступ к приложению Creatio в настройках веб-сервиса (например, вы можете отменить доступ к определенным приложениям в настройках своей учетной записи Google).
4. Добавьте пользователя повторно.

## Разрешения, запрошенные Creatio, отклонены при входе в приложение

Корректно указанные разрешения в Creatio могут быть не приняты при авторизации. В некоторых приложениях, например, в [Facebook](#), список выданных разрешений может быть изменен пользователем веб-сервиса во время логина.

Например, чтение и изменение данных по умолчанию разрешены. Если после авторизации пользователь в настройках снимает признак (чекбокс), разрешающий изменение данных, то когда Creatio запросит доступ на чтение и изменение данных, приложение выдаст только токен доступа на чтение данных.

При попытке вызвать веб-сервис с целью изменения его данных внешнее приложение выдаст ошибку.

Чтобы решить проблему, выполните следующее:

1. Удалите из Creatio пользователя веб-сервиса, чтобы ранее выданный токен доступа был отозван.
2. Отмените доступ к приложению Creatio в настройках веб-сервиса (например, вы можете отменить доступ к определенным приложениям в настройках своей учетной записи Google).
3. Добавьте пользователя в Creatio еще раз, отметив все нужные разрешения (scopes) во время авторизации.

## Не установлен признак [ *Использовать аутентификацию* ] в методе веб-сервиса или способ аутентификации выбран неверно

Этот случай в большей степени относится ко всему процессу настройки интеграции с веб-сервисами, чем к OAuth-аутентификации. В подобной ситуации веб-сервис возвращает ошибку авторизации 403.

Ошибка происходит при вызове веб-сервиса, поскольку Creatio пытается подключиться без передачи токена доступа.

Чтобы решить проблему, выберите корректный вид аутентификации на вкладке [ *Аутентификация* ] страницы настройки веб-сервиса. Или установите признак [ *Использовать аутентификацию* ] в методах веб-сервиса.

## Установлены другие ограничения доступа в настройках веб-сервиса

Некоторые сервисы могут ограничивать доступ по IP или доменам, с которых выполняется авторизация. Например, в Facebook есть ограничения по доменам.

Ошибка возникает при добавлении пользователя на странице OAuth-приложения.

Чтобы решить проблему, на стороне веб-сервиса добавьте и корректно настройте IP и/или домен вашего приложения Creatio.

## Приложение ожидает client ID и client secret в другой части запроса

При запросе или обновлении токена Creatio по умолчанию включает идентификатор клиента (client id) и секретный ключ (client secret) в тело запроса. Например:

```
POST /token HTTP/1.1
Host: server.example.com
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token&refresh_token=tGzv3J0kF0XG5Qx2TlKWIA
```

```
&client_id=s6BhdRkqt3&client_secret=7Fjfp0ZBr1KtDRbnfVdmIw
```

Большинство сервисов, например, Google, LinkedIn, Salesforce, JIRA и т. д. обрабатывают такие запросы. Однако, существуют сервисы, которые не принимают client id и client secret в теле запроса. Протокол OAuth допускает разные варианты.

Например, [Docusign](#), [GoToWebinar](#) передают client id и client secret в форме базовой аутентификации:

```
curl -X POST "https://api.getgo.com/oauth/v2/token" \:
-H "Authorization: Basic {Base64 Encoded consumerKey and consumerSecret}" \
-H "Accept:application/json" \
-H "Content-Type: application/x-www-form-urlencoded" \
-d "grant_type=authorization_code&code={responseKey}&redirect_uri=http%3A%2F%2Fcode.example.com"
```

Этот способ описан в формате [RFC](#) (документация, содержащая технические спецификации и стандарты, которые применяются в интернете) как альтернатива подходу, используемому в Creatio.

Ошибка возникает при попытке пользователя авторизоваться в приложении. Авторизация не происходит. На сегодняшний день нет других способов отправить идентификатор клиента (client id) и секретный ключ (client secret) в Creatio.

## При авторизации в приложении отсутствует токен обновления (refresh token)

Refresh token нужен для автоматического обновления токена доступа. В результате интеграция будет работать без необходимости подтверждать доступ до тех пор, пока токен доступа обновляется автоматически. Токен обновления используется не всегда. В зависимости от архитектуры внешнего приложения такие случаи не всегда являются ошибкой интеграции.

Ниже приведены типовые примеры, когда токен обновления не используется.

- Токен доступа не имеет срока действия.** Внешнее приложение выдает токен доступа (access token), который не требует обновления, т. к. не имеет срока действия. Такие случаи достаточно редки. Несмотря на предупреждающее уведомление, интеграция может работать в течение длительного времени.

Чтобы решить проблему, изучите документацию веб-сервиса. Убедитесь, что приложение не выдает токен обновления, а токен доступа к приложению не имеет срока действия. В таком случае можно проигнорировать предупреждение о необходимости обновления токена доступа и продолжать работать с веб-сервисом.
- В приложении не предусмотрено обновление токенов доступа.** Внешнее приложение выдает токен доступа с определенным сроком действия. Но такой токен нельзя обновить в связи с ограничениями в настройках приложения (требования безопасности). Такие случаи возможны при интеграции с платежными приложениями, предусматривающими лишь кратковременный доступ пользователя.

Подобные приложения не работают с общим пользователем, поскольку общий доступ — это возможность избежать необходимости авторизоваться в веб-сервисе для каждого отдельного пользователя Creatio.

Чтобы решить проблему, изучите документацию веб-сервиса. Убедитесь, что приложение не выдает токен обновления, а токен доступа к приложению имеет срок действия.

- **Обновление токена доступа необходимо включить в настройках приложения.** Внешнее приложение выдает токен доступа, который имеет срок действия. Токен доступа не может быть обновлен, т. к. в приложении не настроена возможность его обновления. Например, в приложении необходимо настроить отдельное разрешение на запрос токена обновления (refresh token). Для решения проблемы выполните следующее:
  - Изучите документацию веб-сервиса. Убедитесь, что приложение выдает токены обновления.
  - Определите, где в настройках приложения добавляются разрешения на выдачу токена обновления и заполните эти настройки.
  - Выполните отзыв текущего токена доступа.
  - Добавьте общего пользователя в Creatio.
- **Приложение выдает токен обновления (refresh token) однократно на каждого пользователя.** Внешнее приложение может выдавать токен обновления только один раз. Пользователь, получив токен обновления и войдя в приложение, повторно получить его не может.

**Пример.** Администратор добавил в Creatio токен этого же внешнего приложения.

Администратор удалил токен из Creatio, не отозвав его (например, удалил токен из базы данных, не заполнив поле [ *URL отзыва токена* ]).

Когда администратор добавляет новый токен в Creatio, внешнее приложение не выдаст токен обновления повторно, так как ранее такой токен уже был выдан, но не был отменен.

Также проблема может возникнуть, если администратор вошел во внешнее приложение из другой системы, а затем пытается войти в него же из Creatio.

Чтобы решить проблему, изучите документацию веб-сервиса. Убедитесь, что приложение выдает токены обновления. Определите, как отменить доступ к веб-сервису. Обычно в веб-сервисах есть форма или пользовательский интерфейс для управления доступом внешних систем. Удалите токен приложения, который пользователь добавляет в Creatio (например, удалите интеграцию с Creatio).

В результате при добавлении токена доступа в Creatio интегрированное приложение выдаст токен обновления.

- **Токен обновления (refresh token) был аннулирован во внешнем приложении, но все еще хранится в Creatio.** Это может произойти, если администратор вошел в Creatio, и в интегрированное приложение, а затем удалил токен доступа на странице управления токеном веб-сервиса. В результате в Creatio будут храниться устаревшие токены доступа и обновления (access token и refresh token). Во время вызова веб-сервиса в Creatio возникает ошибка авторизации. Чтобы ее решить, выполните следующее:
  - Удалите пользователя со страницы настройки OAuth-приложения в Creatio.
  - Добавьте пользователя в Creatio еще раз.