

Email-сообщения

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

Общие принципы работы с email-сообщениями	4
Пользовательская фильтрация нежелательных обращений	4
Работа с цепочками email-сообщений	4
Обработчик макроса в шаблоне email-сообщения	8
Добавить свойство для фильтрации email-сообщений	8
Описание кейса	8
Алгоритм реализации кейса	8
Добавить обработчик макроса шаблона email-сообщения	9
Описание примера	9
Исходный код	10
Пример реализации примера	10
Отправка email-сообщений	12
Алгоритм отправки email-сообщения с существующей учетной записи	12
Алгоритм отправки email-сообщения с явным указанием учетных данных	14
Отправить email-сообщение с существующей учетной записи	16
Алгоритм реализации примера	17
Отправить email-сообщение с явным указанием учетных данных	28
Алгоритм реализации примера	28

Общие принципы работы с email-сообщениями



Пользовательская фильтрация нежелательных обращений

Creatio предоставляет возможность настроить фильтр нежелательных обращений. Пользователи имеют возможность фиксировать список email адресов и доменов, письма из которых могут автоматически помечаться как спам. При этом обращения по ним могут либо не регистрироваться, либо регистрироваться в отмененном состоянии (статус указывается в системной настройке [*Состояние нежелательных обращений по умолчанию*]).

Также фильтр нежелательных обращений позволяет анализировать заголовок письма на предмет наличия в нем определенных флагов, например `Auto-Submitted`. При наличии таких флагов фильтр позволяет не создавать обращения или создавать обращения в состоянии, указанном в системной настройке.

Для того чтобы анализировать электронный адрес, домен или часть адреса, достаточно просто добавить необходимое значение в справочник [*Черный список email адресов и доменов для регистрации обращений*]. После добавления значения (тип будет установлен автоматически), во время анализа входящего письма адреса будут помечены как входящие в черный список.

Анализ заголовка письма происходит с использованием справочника [*Управление свойствами заголовков email*]. Для того чтобы добавить новое свойство для анализа, необходимо выполнить следующие шаги:

1. Добавить новый класс, реализующий интерфейс `IHeaderPropertyHandler`. Этот интерфейс содержит в себе единственный метод `Check()`, возвращающий значение типа `bool`. В реализации метода `Check()` необходимо выполнить проверку значения свойства и вернуть результат. Если метод возвращает `true`, обращение будет создано по стандартному механизму, если `false` — результат проверки отрицательный и письмо будет обработано как письмо из черного списка.
2. Добавить значение в справочник [Управление свойствами заголовков email]. В колонке `Name` указать имя свойства, анализ которого должен проводиться при регистрации. В колонке `handler` должно содержаться имя класса, который был добавлен в предыдущем пункте.

Работа с цепочками email-сообщений

Начиная с версии 7.10.0 в приложении добавлен механизм формирования цепочек Email-сообщений. Целью данного решения является возможность улучшить пользовательский интерфейс работы с Email-сообщениями. Также использование этого механизма позволит упростить поиск связей для писем, например, копированием связей предыдущего письма.

Разработанный механизм построения цепочек опирается на данные из заголовков Email-сообщений `In-Reply-To`. В этом заголовке по общепринятым соглашениям должен содержаться идентификатор

письма `Message-ID`, на который отвечает текущее письмо. Приложение Creatio получает эти заголовки из Email-сервиса, с которым настроена синхронизация (IMAP/Exchange).

Условно механизм работы с цепочками можно поделить на две логические части — создание цепочки и заполнение связей активности.

Создание цепочек email-сообщений

В детали `EmailMessageData` добавлены три поля, отвечающие за цепочки Email-сообщений:

- `MessageId` — строка длиной 500 символов;
- `InReplyTo` — строка длиной 500 символов;
- `ParentMessageId` — справочное поле, ссылающееся на деталь `EmailMessageData`.

При синхронизации писем поля `MessageId` и `InReplyTo` заполняются соответствующими значениями из заголовков письма.

Поле `ParentMessageId` заполняется следующим образом:

1. В таблице `EmailMessageData` выполняется поиск записей, у которых `MessageId` равен текущему `InReplyTo`. Если записи найдены, первая из них устанавливается как текущий `ParentMessageId`.
2. Во всех записях `EmailMessageData`, у которых `InReplyTo` равен текущему `MessageId` и поле `ParentMessageId` не заполнено, полю `ParentMessageId` присваивается значение, равное текущему `Id`.

Таким образом, для каждого письма актуализируются связи с соседними письмами в цепочках.

Копирование связей предыдущего email-сообщения в цепочке

После добавления письма необходимо распространить связи активности по цепочке. Этот процесс будет рассмотрен на примере поля [*Обращение*].

Для текущей записи `EmailMessageData` выполняется поиск такой родительской записи, у которой указана активность с заполненным полем [*Обращение*]. Значение поля [*Обращение*] из этой активности будет распространяться по цепочке. Начиная с найденной записи `EmailMessageData` выполняется поиск всех дочерних записей `EmailMessageData` и обновляется значение поля [*Обращение*] в найденных активностях.

Например, существует цепочка из трех Email-сообщений:

ActivityId	Title	CaseId	EmailMessageId	EmailMessageParentId
28BD6D59-B9D7-4FF9-89F5-FEE1DD003912	Re: relation	NULL	66812FBF-411B-4FE0-94C9-1E70FBBEB2D3	F05B529D-C98C-4E26-BE00-21F8721AEF58
DC0A40D4-700A-40EB-B394-90E0376C3B5D	Re: relation	1C6E18E3-48B1-495E-8EF9-ACA35DB9DE0B	F05B529D-C98C-4E26-BE00-21F8721AEF58	E1A0120E-B7C0-4261-9DE0-C63341BF1E0B
D7A9B82C-ED46-437C-980A-B2650D4FF3DA	relation	906909E8-4D64-47FD-AF92-B65B0826AEC3	E1A0120E-B7C0-4261-9DE0-C63341BF1E0B	NULL

Затем в цепочку приходит еще одно Email-сообщение:

ActivityId	Title	CaseId	EmailMessageId	EmailMessageParentId
6623B052-73AD-4AE5-AE61-A6F9BCD930A0	Re: relation	NULL	60C00B01-D0BF-40F6-923E-1830E433AEA1	66812FBF-411B-4FE0-94C9-1E70FBBEB2D3
28BD6D59-B9D7-4FF9-89F5-FEE1DD003912	Re: relation	NULL	66812FBF-411B-4FE0-94C9-1E70FBBEB2D3	F05B529D-C98C-4E26-BE00-21F8721AEF58
DC0A40D4-700A-40EB-B394-90E0376C3B5D	Re: relation	1C6E18E3-48B1-495E-8EF9-ACA35DB9DE0B	F05B529D-C98C-4E26-BE00-21F8721AEF58	E1A0120E-B7C0-4261-9DE0-C63341BF1E0B
D7A9B82C-ED46-437C-980A-B2650D4FF3DA	relation	906909E8-4D64-47FD-AF92-B65B0826AEC3	E1A0120E-B7C0-4261-9DE0-C63341BF1E0B	NULL

Начиная с записи, у которой `EmailMessageId` = "60C00B01-D0BF-40F6-923E-1830E433AEA1" выполняется поиск записи, у которой колонка `CaseId` заполнена (не содержит `NULL`). Это запись, у которой `EmailMessageId` = "F05B529D-C98C-4E26-BE00-21F8721AEF58", а `CaseId` = "1C6E18E3-48B1-495E-8EF9-ACA35DB9DE0B".

Теперь начиная с записи, у которой `EmailMessageId` = "F05B529D-C98C-4E26-BE00-21F8721AEF58",

необходимо обновить значение поля `CaseId` для дочерних записей по всей цепочке.

ActivityId	Title	CaseId	EmailMessageId	EmailMessageParentId
6623B052-73AD-4AE5-AE61-A6F9BCD930A0	Re: relation	1C6E18E3-48B1-495E-8EF9-ACA35DB9DE0B	60C00B01-D0BF-40F6-923E-1830E433AEA1	66812FBF-411B-4FE0-94C9-1E70FBBEB2D3
28BD6D59-B9D7-4FF9-89F5-FEE1DD003912	Re: relation	1C6E18E3-48B1-495E-8EF9-ACA35DB9DE0B	66812FBF-411B-4FE0-94C9-1E70FBBEB2D3	F05B529D-C98C-4E26-BE00-21F8721AEF58
DC0A40D4-700A-40EB-B394-90E0376C3B5D	Re: relation	1C6E18E3-48B1-495E-8EF9-ACA35DB9DE0B	F05B529D-C98C-4E26-BE00-21F8721AEF58	E1A0120E-B7C0-4261-9DE0-C63341BF1E0B
D7A9B82C-ED46-437C-980A-B2650D4FF3DA	relation	906909E8-4D64-47FD-AF92-B65B0826AEC3	E1A0120E-B7C0-4261-9DE0-C63341BF1E0B	NULL

На заметку.

Сложности данного механизма обусловлены тем, что очень часто почтовый сервер отдает письма не в той последовательности, в которой они были написаны в цепочке.

Самый простой и частый пример — в ходе сессии синхронизации сначала приходит почта из папки "Входящие", а затем — из папки "Исходящие". Если же письма были получены не последовательно, то не будет возможности построить цепочку на момент загрузки каждого письма. Но такая возможность появляется к завершению цикла синхронизации всех писем почтового ящика. Безусловно, логика поиска цепочек может не сработать, если настроена загрузка не всех папок почтового ящика, или цепочка была прервана другими участниками переписки. Тем не менее, в большом количестве случаев цепочку можно построить на момент завершения загрузки всех писем из почтового ящика.

Рекомендации по добавлению пользовательского процесса поиска всех связей в цепочках после загрузки email-сообщения

Для того чтобы начать работу по новому полученному Email-сообщению после завершения формирования цепочек, необходимо знать следующее:

1. В таблице `EmailMessageData` появляется поле `Id` сессии синхронизации, уникальное для одного запуска процесса синхронизации. Заполняется только для синхронизированных Email-сообщений.
2. Появляется таблица `FinishedSyncSession`, в которую добавляется запись, если в рамках синхронизации

пришло хоть одно Email-сообщение.

3. Процессы, ранее слушавшие сигнал сохранения активности, в новой реализации слушают сигнал добавления (`Inserted`) объекта `FinishedSyncSession`. По значению `Id` сессии синхронизации выбираются Email-сообщения из `EmailMessageData`. Также в `EmailMessageData` существует поле `MailboxSyncSettings`, которое можно использовать для отбора Email-сообщений из сервисного ящика.

Обработчик макроса в шаблоне email-сообщения

Шаблоны email-сообщений используются, например, при общении со службой поддержки. Шаблоны доступны в справочнике [*Шаблоны email-сообщений*] ([*Email templates*]). Подробнее об их настройке можно узнать из статьи "[Настройка автоматической отправки email-уведомлений](#)".

Например, в письме, уведомляющем клиента о закрытии его обращения, используется шаблон "Сообщение о закрытии обращения". Для подстановки некоторых значений колонок объектов системы в шаблоны email-сообщений используются преднастроенные макросы, например, обращение к контакту или должность ответственного.

Creatio позволяет реализовать собственную логику заполнения значения, которое возвращает обработчик макросов. При этом во время обработки макроса система выполнит реализованный разработчиком алгоритм, а не базовую логику.

На специализированный обработчик класса указывает тег `@Invoke`. Далее через точку должно быть указано имя класса реализующего интерфейс `IMacrosInvokable`, который включает в себя метод `GetMacrosValue()`. Этот метод должен вернуть строку, которая будет подставлена вместо строки макроса.

Для реализации пользовательского обработчика макроса необходимо:

1. Создать класс, реализующий интерфейс `IMacrosInvokable`.
2. Зарегистрировать макрос в таблице `EmailTemplateMacros`, указав для него `ParentId` (базовый шаблон с тегом `@Invoke`) и `ColumnPath` (имя класса).
3. Добавить макрос в шаблон email-сообщения.

Добавить свойство для фильтрации email-сообщений

 Средний

Описание кейса

Добавить для анализа обращения новое свойство `No-reply`. В случае, если данное свойство присутствует в заголовке письма и его значение отлично от "No", обращение обработать, как обращение из черного списка.

Алгоритм реализации кейса

1. Добавление нового класса, реализующего интерфейс IHeaderPropertyHandler

В пользовательском пакете (например, `Custom`) необходимо добавить новую схему типа "Исходный код". В этой схеме необходимо определить класс, реализующий интерфейс `IHeaderPropertyHandler`, например:

```
namespace Terrasoft.Configuration
{
    using System;
    public class NoreplyHandler: IHeaderPropertyHandler
    {
        public bool Check(object value) {
            return string.Equals(value.ToString(), "No", StringComparison.OrdinalIgnoreCase);
        }
    }
}
```

После сохранения созданную схему необходимо опубликовать.

Интерфейс `IHeaderPropertyHandler` определен в пакете `JunkFilter`, поэтому он обязательно должен быть добавлен в зависимости пользовательского пакета.

2. Добавление значения в справочник

Для корректной работы нового фильтра необходимо добавить в справочник [*Управление свойствами заголовков email*] новое свойство `No-reply`. В качестве класса обработчика (свойство `handler`) необходимо указать созданный на предыдущем шаге класс `NoreplyHandler`.

В результате выполнения перечисленных действий, после получении входящего письма, содержащего в своем заголовке флаг `No-reply` со значением, отличным от "No", возможны следующие варианты:

- обращение не будет создано в случае, если значение системной настройки [Создавать обращения по нежелательным письмам] равно `false`;
- обращение будет создано со статусом, указанным в системной настройке [Состояние нежелательных обращений по умолчанию], если значение системной настройки [Создавать обращения по нежелательным письмам] равно `true`.

Добавить обработчик макроса шаблона email-сообщения



Пример реализован для продуктов линейки Service Creatio.

Описание примера

Добавить обработчик макроса шаблона email-сообщения, который будет возвращать строку "Test".

Исходный код

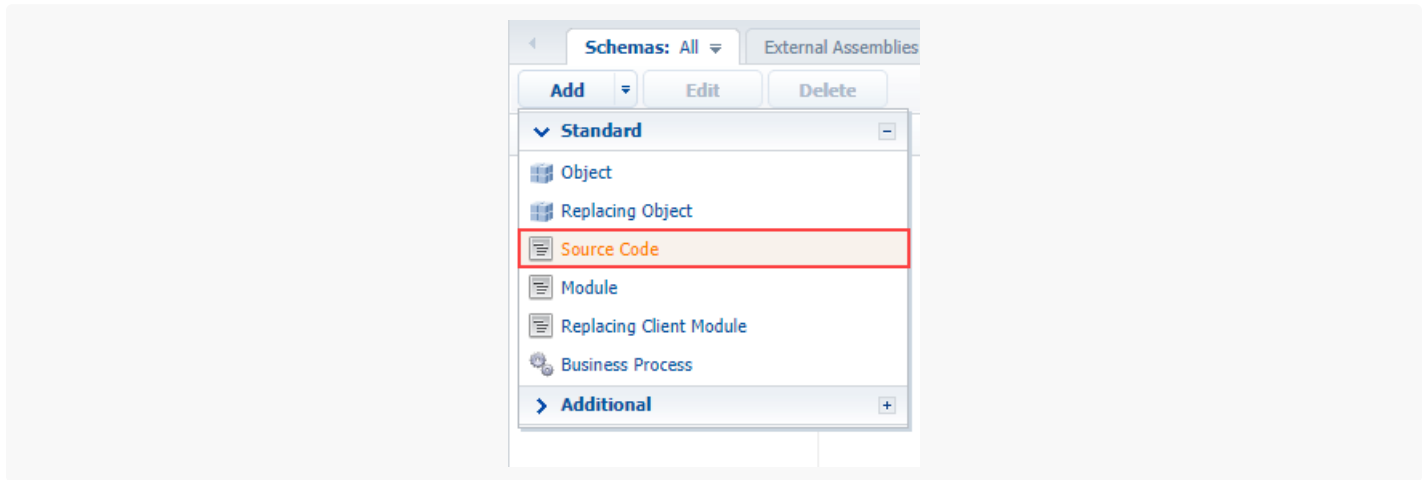
Пакет с реализацией примера можно скачать по [ссылке](#).

Пример реализации примера

1. Создание класса, реализующего интерфейс IMacrosInvokable

Чтобы создать класс, реализующий интерфейс `IMacrosInvokable`, добавьте в пакет разработки схему [Исходный код] ([*Source Code*]). Для этого в разделе [Конфигурация] на вкладке [Схемы] ([*Schemas*]) выполните команду меню [Добавить] ([*Add*]) — [Исходный код] ([*Source Code*]) (рис. 1).

Рис. 1. — Создание новой схемы [Исходный код] ([*Source Code*])



Для созданной схемы объекта укажите:

- [Заголовок] ([Title]) — "Генератор текстовой строки" ("Text string generator").
- [Название] ([Name]) — "UsrTestStringGenerator".

Исходный код схемы:

```
namespace Terrasoft.Configuration
{
    using System;
    using Terrasoft.Core;
    // Класс обработчика макроса шаблона Email-сообщения.
    public class UsrTestStringGenerator : IMacrosInvokable
    {
        // Пользовательское соединение.
        public UserConnection UserConnection {
            get;
            set;
        }
    }
}
```

```

    }
    // Метод, возвращающий подставляемое значение.
    public string GetMacrosValue(object arguments) {
        return "Тестовая строка";
    }
}
}

```

Опубликуйте созданную схему.

2. Регистрация макроса в таблице EmailTemplateMacros

Чтобы зарегистрировать макрос в таблице `EmailTemplateMacros`, выполните следующий SQL-запрос:

```

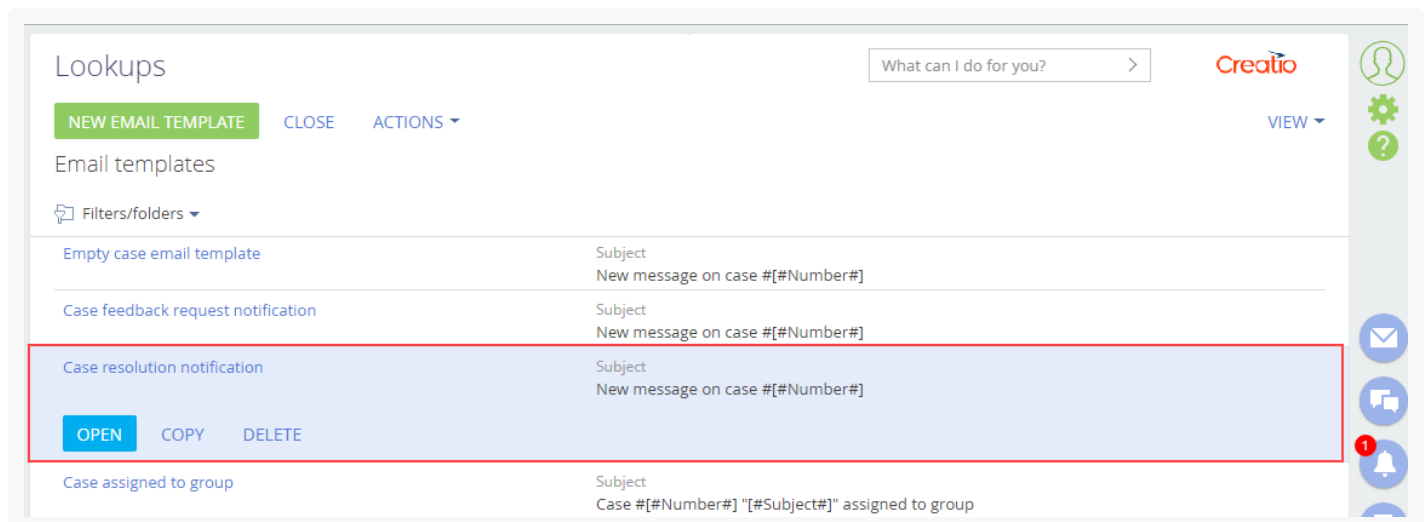
INSERT INTO EmailTemplateMacros(Name, Parentid, ColumnPath)
VALUES (
    'UsrTestStringGenerator',
    (SELECT TOP 1 Id
     FROM EmailTemplateMacros
     WHERE Name = '@Invoke'),
    'Terrasoft.Configuration.UsrTestStringGenerator'
)

```

3. Добавление макроса в шаблон email-сообщения

После регистрации макроса его можно использовать в шаблонах email-сообщений. Для этого измените одну или несколько записей справочника [*Шаблоны email-сообщений*] ([*Email templates*]) (рис. 2).

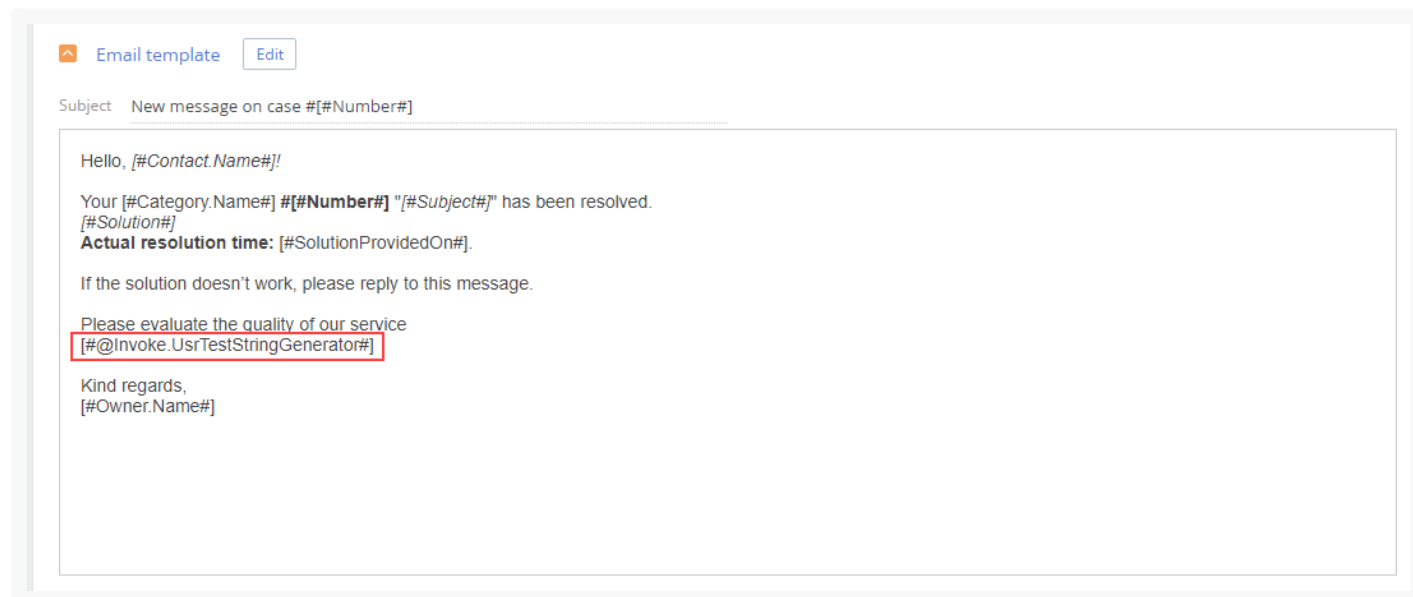
Рис. 2. — Справочник [*Шаблоны email-сообщений*]



Например, для изменения содержимого сообщений, автоматически отправляемых при разрешении

обращения, необходимо изменить запись [Сообщение о разрешении обращения] ([Case resolution notification]). Если в шаблон добавить макрос [#@Invoke.UsrTestStringGenerator#] (рис. 3), то при отправке сообщения пользователю вместо макроса будет подставлено значение "Test".

Рис. 3. — Макрос в шаблоне Email-сообщения



Отправка email-сообщений



Важно. Возможность отправки email-сообщений с явным указанием учетных данных доступна в Creatio версии 7.16 и выше.

В Creatio вы можете отправлять email-сообщения не только пользовательскими средствами, но и средствами разработки. Это реализуется следующими способами:

- С существующей учетной записи.
- С явным указанием учетных данных.

Отправка email-сообщения реализована с помощью бизнес-процесса. Для настройки бизнес-процесса используются элементы [\[Автогенерируемая страница \]](#) ([Auto-generated page]) и [\[Задание-сценарий \]](#) ([Script task]).

Алгоритм отправки email-сообщения с существующей учетной записи

1. Создать бизнес-процесс, который должен включать обязательные элементы [Автогенерируемая страница] и [Задание-сценарий].
2. Создать конфиг отправляемого email-сообщения.

3. Добавить вложение (не обязательно).
4. Запустить бизнес-процесс для выполнения отправки.

Создание конфига отправляемого email-сообщения

Для создания конфига отправляемого email-сообщения необходимо использовать класс

`Terrasoft.Mail.Sender.EmailMessage`. Для формирования валидного email-сообщения необходимо заполнить параметры.

Формирование валидного email-сообщения

```
var message = new Terrasoft.Mail.Sender.EmailMessage {
    // Email-адрес отправителя.
    From = "Sender@email.com",
    // Email-адреса получателей.
    To = List<string>{ "first@recepient.co", "second@recepient.co"},
    // Копия (не обязательно).
    Cc = List<string>{ "first@recepient.co", "second@recepient.co"},
    // Скрытая копия (не обязательно).
    Bcc = List<string>{ "first@recepient.co", "second@recepient.co"},
    // Тема письма.
    Subject = "Message subject",
    // Тело письма.
    Body = "Body",
    // Приоритет, значения из перечисления Terrasoft.Mail.Sender.EmailPriority.
    Priority = Terrasoft.Mail.Sender.EmailPriority.Normal
};
```

Добавление вложения (не обязательно)

В email-сообщение можно добавить вложение. Для этого необходимо заполнить поле [*Attachments*].

Вложения представляют собой список экземпляров `Terrasoft.Mail.Sender.EmailAttachments`.

Добавление вложения

```
// Создание вложения.
var attachment = new Terrasoft.Mail.Sender.EmailAttachment {
    // Идентификатор вложения.
    Id = new Guid("844F0837-EAA0-4F40-B965-71F5DB9EAE6E"),
    // Имя вложения.
    Name = "attachName.txt",
    // Данные.
    Data = byteData
};
// Добавление вложения в письмо.
message.Attachments.Add(attachment);
```

Отправка email-сообщения

Для отправки письма необходимо использовать метод `Send` класса `EmailSender` с переданными параметрами email-сообщения и конфига подключения.

Отправка email-сообщения

```
// Отправка сформированного email-сообщения. Для игнорирования прав доступа при отправке
// требуется присвоить параметру ignoreRights значение true.
emailSender.Send(message, ignoreRights);
```

Алгоритм отправки email-сообщения с явным указанием учетных данных

1. Создать бизнес-процесс, который должен включать обязательные элементы [*Автогенерируемая страница*] и [*Задание-сценарий*].
2. Создать экземпляр класса `EmailClientFactory`.
3. Создать экземпляр класса `EmailSender`.
4. Создать конфиг подключения к почтовому ящику.
5. Создать конфиг отправляемого email-сообщения.
6. Добавить вложение (не обязательно).
7. Запустить бизнес-процесс для выполнения отправки.

Создание экземпляра класса EmailClientFactory

Для создания экземпляра класса `EmailClientFactory` необходимо наличие пользовательского подключения `UserConnection`.

Создание экземпляра класса EmailClientFactory

```
var emailClientFactory = ClassFactory.Get<IEmailClientFactory>(
    new ConstructorArgument("userConnection", UserConnection));
```

Создание экземпляра класса EmailSender

Для создания экземпляра класса `EmailSender` необходимо передать в конструктор созданный экземпляр `EmailClientFactory` и пользовательское подключение `UserConnection`.

Создание экземпляра класса EmailClientFactory

```
var emailSender = ClassFactory.Get<IEmailSender>(
    new ConstructorArgument("emailClientFactory", emailClientFactory),
    new ConstructorArgument("userConnection", UserConnection));
```

Создание конфига подключения к почтовому ящику

Для создания конфига подключения к почтовому ящику используется класс

`EmailContract.DTO.Credentials`. Чтобы создать конфиг, заполните параметры.

Создание конфига подключения к почтовому ящику

```
var credentialConfig = new EmailContract.DTO.Credentials {
    // Имя или IP-адрес сервера исходящей почты.
    ServiceUrl = "mail service host",
    // Может быть пустым для некоторых протоколов.
    Port = "Port",
    // Использовать протокол SSL для шифрования подключения.
    UseSsl = false,
    // Имя пользователя почтового ящика.
    UserName = "EmailUserName",
    // Пароль пользователя почтового ящика.
    Password = "UserPassword",
    // Тип почтового сервера (Exchange либо Imap/Smtp).
    ServerTypeId = EmailDomain.IntegrationConsts.ExchangeMailServerTypeId ||
        EmailDomain.IntegrationConsts.ImapMailServerTypeId,
    // Почтовый ящик отправителя.
    SenderEmailAddress = "sender@test.com"
};
```

Важно. Параметры `ServiceUrl`, `UserName`, `Password`, `ServerTypeId`, `SenderMailbox` являются обязательными для заполнения.

Создание конфига отправляемого email-сообщения

Для создания конфига отправляемого email-сообщения необходимо использовать класс

`EmailContract.DTO.Email`. Для формирования валидного email-сообщения заполните параметры.

Создание конфига отправляемого email-сообщения

```
var message = new EmailContract.DTO.Email {
    // Email-адрес отправителя.
    Sender = "Sender@email.com",
```

```
// Email-адреса получателей.
Receipients = List<string>{ "first@recepient.co", "second@recepient.co"},
// Тема письма.
Subject = "Message subject",
// Тело письма.
Body = "Body",
// Приоритет, значения из перечисления EmailContract.EmailImportance.
Importance = EmailContract.EmailImportance.High
};
```

Добавление вложения (не обязательно)

В email-сообщение можно добавить вложение. Для этого необходимо заполнить поле [*Attachments*]. Вложения представляют собой список экземпляров `EmailContract.DTO.Attachment`.

Добавление вложения

```
// Создание вложения.
var attachment = new EmailContract.DTO.Attachment {
    Name = "FileName",
    Id = "844F0837-EAA0-4F40-B965-71F5DB9EAE6E"
};
// Установка данных для вложения.
attachment.SetData(byteData);
// Добавление вложения в письмо.
message.Attachments.Add(attachment);
```

Отправка email-сообщения

Для отправки письма необходимо использовать метод `Send` класса `EmailSender` с переданными аргументами email-сообщения и конфига подключения.

Отправка email-сообщения

```
// Отправка сформированного email-сообщения с параметрами подключения к почтовому
// ящику отправителя. Для игнорирования прав доступа при отправке
// требуется присвоить параметру ignoreRights значение true.
emailSender.Send(message, credentialConfig, ignoreRights);
```

Отправить email-сообщение с существующей учетной записи

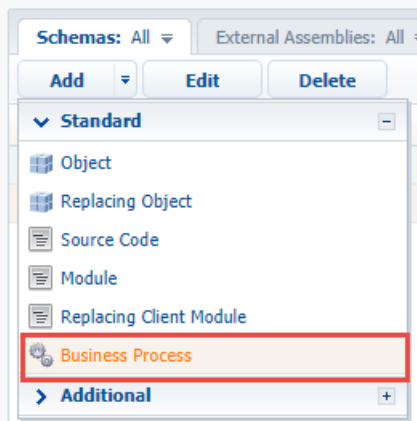


Пример. Создайте бизнес-процесс в ходе которого будет открываться страница, содержащая параметры email-сообщения для заполнения и дальнейшей отправки email-сообщения с существующей учетной записи.

Алгоритм реализации примера

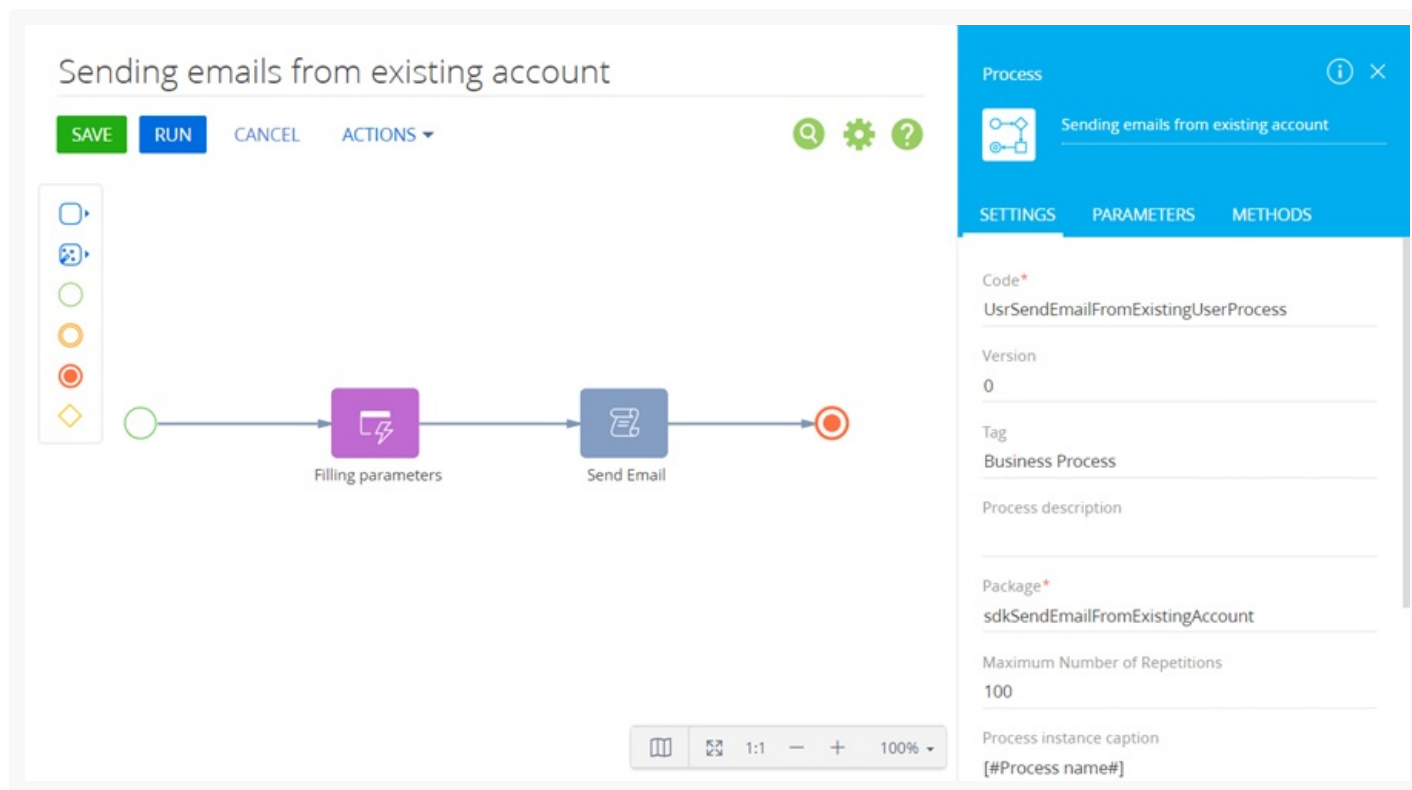
1. Создайте бизнес-процесс

В разделе [Конфигурация] ([*Configuration*]) выполните действие [Добавить] —> [Бизнес-процесс] ([*Add*] —> [*Business process*]).



В открывшемся дизайнера процессов установите следующие значения на панели настройки свойств процесса:

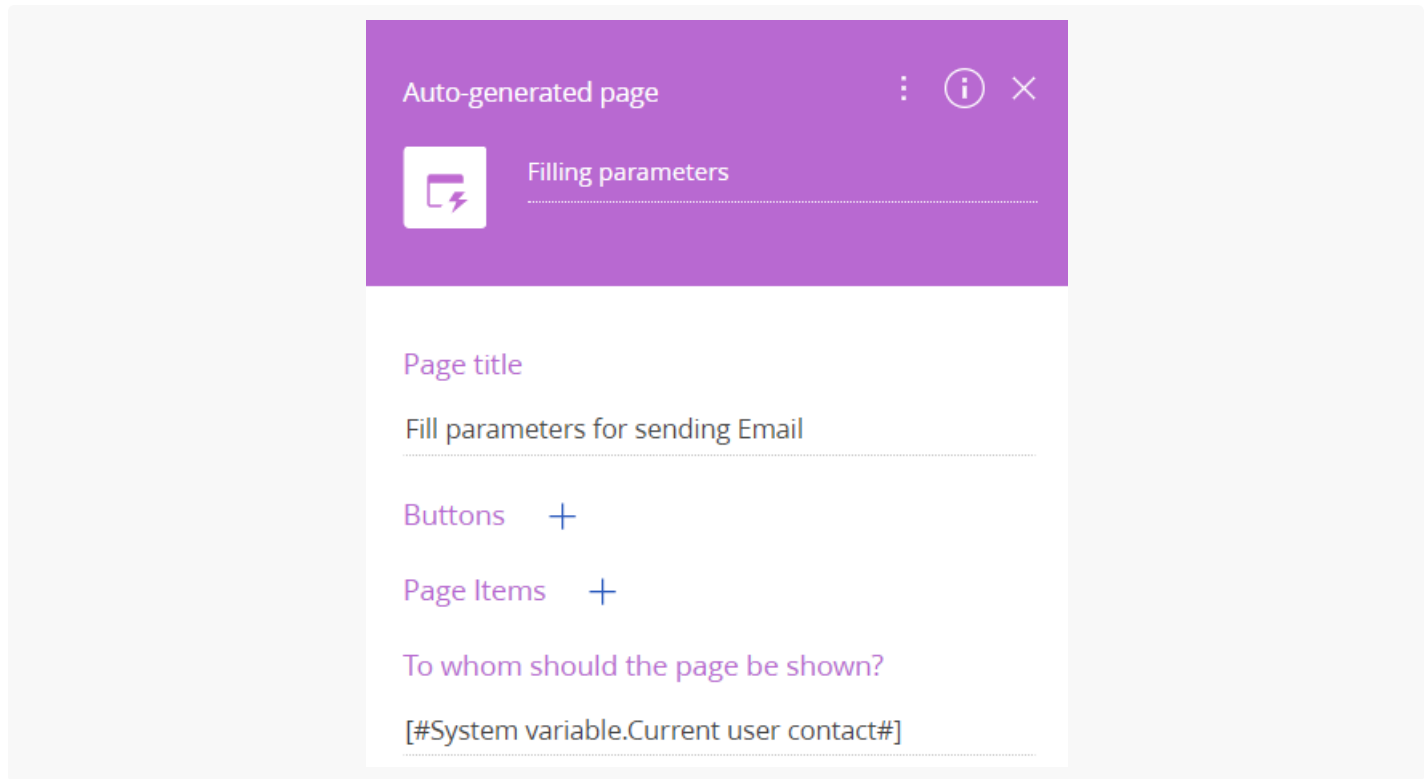
- [Заголовок] ([*Title*]) — "Sending emails from existing account".
- [Код] ([*Code*]) — "UsrSendEmailFromExistingUserProcess".



2. Добавьте элемент [Автогенерируемая страница]

С помощью элемента [Автогенерируемая страница] ([*Auto-generated page*]) в ходе выполнения процесса можно открыть произвольную страницу, которая создана пользователем системы. Для этого элемента добавьте подпись [*Filling parameters*] и установите следующие свойства:

- [*Название страницы*] ([*Page title*]) — "Fill parameters for sending Email".
- [*Кому отобразить страницу?*] ([*To whom should the page be shown?*]) — выберите "Formula" и установите "[#System variable.Current user contact#]".



3. Добавьте кнопку на страницу

Для добавления кнопки [*Continue*] на страницу в блоке [*Кнопки*] ([*Buttons*]) нажмите **+** и введите следующие параметры:

- [*Название*] ([*Caption*]) — "Continue".
- [*Код*] ([*Code*]) — "ContinueButton".
- [*Стиль*] ([*Style*]) — выберите "Green".
- Установите признак [*Активная*] ([*Active*]).
- Установите признак [*Выполняет проверку значений*] ([*Performs value validation*]).

Buttons +

Caption*
Continue

Code*
ContinueButton

Style*
Green

Generate signal

☒ Active

☒ Performs value validation

SAVE CANCEL

Нажмите [Сохранить] ([Save]).

4. Добавьте элементы на страницу

Для добавления на страницу элемента, который будет содержать почтовый ящик отправителя email-сообщения, в блоке [Параметры страницы] ([Page Items]) нажмите +, выберите тип [Справочник] ([Selection field]) и введите следующие параметры:

- [Заголовок] ([Title]) — "Sender Mailbox".
- [Код] ([Code]) — "SenderMailbox".
- [Источник данных] ([Data source]) — выберите "Mailbox synchronization settings".
- [Представление] ([View]) — выберите "Drop down list".

Page Items +

Title*
Sender Mailbox

Code*
SenderMailbox

☐ Required

Data source*
Mailbox synchronization settings

View*
Drop down list

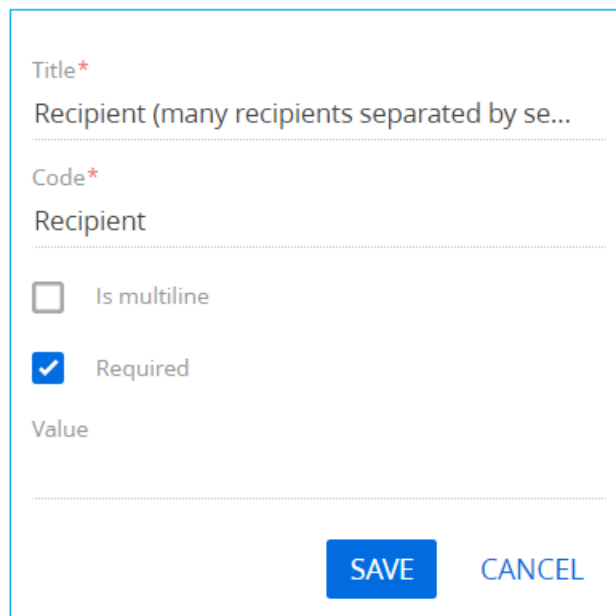
Value

SAVE CANCEL

Нажмите [Сохранить] ([Save]).

Для добавления на страницу элемента, который будет содержать почтовый ящик получателя email-сообщения, в блоке [Параметры страницы] ([Page Items]) нажмите +, выберите тип [Строковое поле] ([Text field]) и введите следующие параметры:

- [Заголовок] ([Title]) — "Recipient (many recipients separated by semicolon ";")".
- [Код] ([Code]) — "Recipient".
- Установите признак [Обязательное] ([Required]).



Title*

Recipient (many recipients separated by se...)

Code*

Recipient

☐ Is multiline

☒ Required

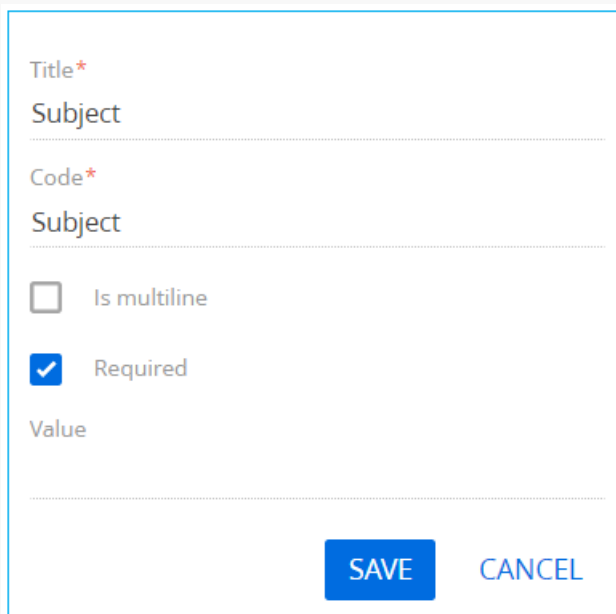
Value

SAVE CANCEL

Нажмите [Сохранить] ([Save]).

Для добавления на страницу элемента, который будет содержать тему email-сообщения, в блоке [Параметры страницы] ([Page Items]) нажмите **+**, выберите тип [Строковое поле] ([Text field]) и введите следующие параметры:

- [Заголовок] ([Title]) — "Subject".
- [Код] ([Code]) — "Subject".
- Установите признак [Обязательное] ([Required]).



Title*

Subject

Code*

Subject

☐ Is multiline

☒ Required

Value

SAVE CANCEL

Нажмите [Сохранить] ([Save]).

Для добавления на страницу элемента, который будет содержать тело email-сообщения, в блоке

[*Параметры страницы*] ([*Page Items*]) нажмите **+**, выберите тип [*Строковое поле*] ([*Text field*]) и введите следующие параметры:

- [*Заголовок*] ([*Title*]) — "Body".
- [*Код*] ([*Code*]) — "Body".
- Установите признак [*Обязательное*] ([*Required*]).

The image shows a configuration window for a 'Text field'. It has two text input fields: 'Title*' and 'Code*', both containing the text 'Body'. Below these are two checkboxes: 'Is multiline' (unchecked) and 'Required' (checked). At the bottom, there is a 'Value' field and two buttons: 'SAVE' and 'CANCEL'.

Нажмите [*Сохранить*] ([*Save*]).

5. Добавьте элемент [*Задание-сценарий*]

Свойству [*Заголовок*] ([*Title*]) элемента [*Задание-сценарий*] ([*Script task*]) присвойте значение "Send Email". Элемент должен выполнять программный код.

Send Email

```
// Id выбранного почтового ящика.
var mailBoxSettingId = Get<Guid>("SenderMailbox");
// Создание экземпляра EmailClientFactory.
var emailClientFactory = ClassFactory.Get<IEmailClientFactory>(
    new ConstructorArgument("userConnection", UserConnection));
// Создание экземпляра IEmailSender.
var emailSender = ClassFactory.Get<IEmailSender>(
    new ConstructorArgument("emailClientFactory", emailClientFactory),
    new ConstructorArgument("userConnection", UserConnection));

var entity = UserConnection.EntitySchemaManager.GetInstanceByName("MailboxSyncSettings").CreateEntity
if (entity.FetchFromDB("Id", mailBoxSettingId, new List<string> { "SenderEmailAddress" }))) {
    // Получение почтового адреса отправителя из выбранного почтового ящика.
```

```

var senderEmailAddress = entity.GetTypedColumnValue<string>("SenderEmailAddress");
// Заполнение параметров отправляемого сообщения.
var message = new Terrasoft.Mail.Sender.EmailMessage {
    // Email-адрес отправителя.
    From = senderEmailAddress,
    // Email-адреса получателей.
    To = Get<string>("Recipient").Split(';').ToList<string>(),
    // Копия (не обязательно).
    // Сс = List<string>{ "first@recepient.co", "second@recepient.co"},
    // Скрытая копия (не обязательно).
    // Вcc = List<string>{ "first@recepient.co", "second@recepient.co"},
    // Тема письма.
    Subject = Get<string>("Subject"),
    // Тело письма.
    Body = Get<string>("Body"),
    // Приоритет, значения из перечисления Terrasoft.Mail.Sender.EmailPriority.
    Priority = Terrasoft.Mail.Sender.EmailPriority.Normal
};
// Дополнительно можно прикреплять вложения (в примере используются тестовые значения).
// Создание вложения.
var attachment = new Terrasoft.Mail.Sender.EmailAttachment {
    // Идентификатор вложения.
    Id = Guid.NewGuid(),
    // Название файла.
    Name = "test.txt",
    // Данные.
    Data = Encoding.ASCII.GetBytes("some test text")
};
// Добавление вложения в письмо.
message.Attachments.Add(attachment);
// Отправка письма.
emailSender.Send(message);
}

return true;

```

6. Добавьте параметры

Для добавления параметра бизнес-процесса, который будет содержать почтовый ящик получателя email-сообщения, на вкладке [*Параметры*] ([*Parameters*]) панели настройки свойств процесса выполните действие [*Добавить параметр*] —> [*Текст*] ([*Add parameters*] —> [*Text*]) и установите следующие свойства параметра:

- [*Заголовок*] ([*Title*]) — "Recipient".
- [*Код*] ([*Code*]) — "Recipient".
- [*Значение*] ([*Value*]) — нажмите ⚡ —> [*Параметр процесса*] ([*Process parameter*]) и выберите элемент процесса "Recipient (many recipients separated by semicolon ";")".

Title*

Recipient

Code*

Recipient

Data type*

Text (500 characters)

Value

[#Filling parameters.Recipient (many recipi...]

SAVE CANCEL

Нажмите [Сохранить] ([Save]).

Для добавления параметра бизнес-процесса, который будет содержать тему email-сообщения, на вкладке [Параметры] ([Parameters]) панели настройки свойств процесса выполните действие [Добавить параметр] → [Текст] ([Add parameters] → [Text]) и установите следующие свойства параметра:

- [Заголовок] ([Title]) — "Subject".
- [Код] ([Code]) — "Subject".
- [Значение] ([Value]) — нажмите ⚡ → [Параметр процесса] ([Process parameter]) и выберите элемент процесса "Subject".

Title*

Subject

Code*

Subject

Data type*

Text (500 characters)

Value

[#Filling parameters.Subject#]

SAVE CANCEL

Нажмите [Сохранить] ([Save]).

Для добавления параметра бизнес-процесса, который будет содержать тело email-сообщения, на

вкладке [*Параметры*] ([*Parameters*]) панели настройки свойств процесса выполните действие [*Добавить параметр*] —> [*Текст*] ([*Add parameters*] —> [*Text*]) и установите следующие свойства параметра:

- [*Заголовок*] ([*Title*]) — "Body".
- [*Код*] ([*Code*]) — "Body".
- [*Значение*] ([*Value*]) — нажмите ⚡ —> [*Параметр процесса*] ([*Process parameter*]) и выберите элемент процесса "Body".

The screenshot shows a dialog box for adding a parameter. It has four main sections: 'Title*' with the value 'Body', 'Code*' with the value 'Body', 'Data type*' with a dropdown menu showing 'Text (500 characters)', and 'Value' with the expression '[#Filling parameters.Body#]'. At the bottom right, there are two buttons: 'SAVE' and 'CANCEL'.

Нажмите [*Сохранить*] ([*Save*]).

Для добавления параметра бизнес-процесса, который будет содержать почтовый ящик отправителя email-сообщения, на вкладке [*Параметры*] ([*Parameters*]) панели настройки свойств процесса выполните действие [*Добавить параметр*] —> [*Другое*] —> [*Уникальный идентификатор*] ([*Add parameters*] —> [*Other*] —> [*Unique identifier*]) и установите следующие свойства параметра:

- [*Заголовок*] ([*Title*]) — "Sender Mailbox".
- [*Код*] ([*Code*]) — "SenderMailbox".
- [*Значение*] ([*Value*]) — нажмите ⚡ —> [*Параметр процесса*] ([*Process parameter*]) и выберите элемент процесса "Sender Mailbox".

The dialog box contains the following fields and controls:

- Title***: A text field with the value "Sender Mailbox".
- Code***: A text field with the value "SenderMailbox".
- Data type***: A text field with the value "Unique identifier".
- Value**: A text field with the value "[#Filling parameters.Sender Mailbox#]".
- Buttons**: "SAVE" and "CANCEL" buttons at the bottom right.

Нажмите [Сохранить] ([Save]).

7. Добавьте методы

Для добавления методов бизнес-процесса на вкладке [Методы] ([Methods]) панели настройки свойств процесса в блоке [Usings] нажмите **+** и в поле [Пространство имён] ([Name Space]) добавьте значение `Terrasoft.Configuration`. Нажмите [Сохранить] ([Save]).

Таким же способом добавьте следующие пространства имен:

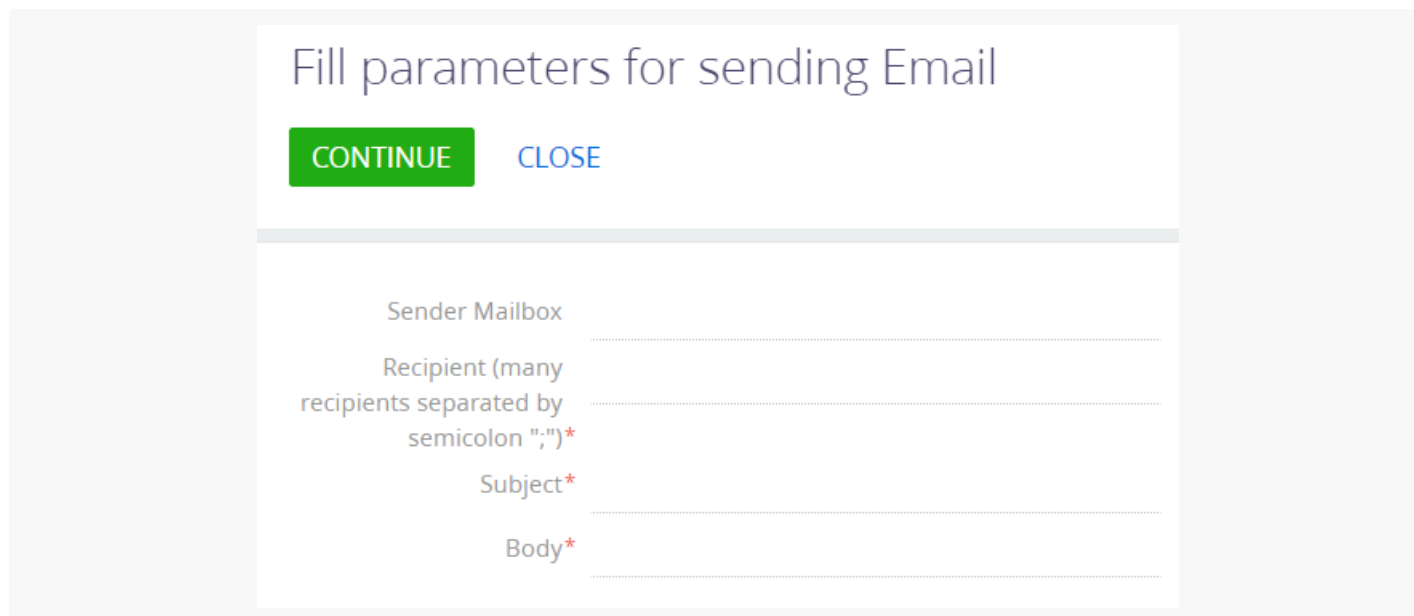
- `Terrasoft.Mail.Sender`
- `Terrasoft.Core.Factories`
- `Terrasoft.Core`
- `Terrasoft.Mail`
- `IntegrationApi`
- `System.Linq`

Сохраните все изменения в дизайнера процессов.

8. Запустите бизнес-процесс

Важно. Для успешного запуска бизнес-процесса необходимо предварительно [добавить учетную запись отправителя в приложение Creatio](#).

После запуска бизнес-процесса по кнопке [Запустить] ([Run]) будет открыта страница для заполнения параметров email-сообщения.



Fill parameters for sending Email

CONTINUE **CLOSE**

Sender Mailbox _____

Recipient (many recipients separated by semicolon ";")* _____

Subject* _____

Body* _____

Для отправки email-сообщения с существующей учетной записи нажмите [*Continue*].

Отправить email-сообщение с явным указанием учетных данных

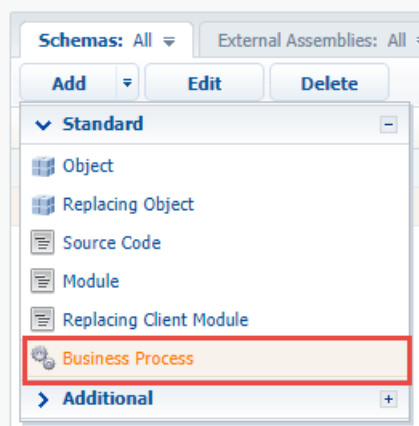
 Сложный

Пример. Создайте бизнес-процесс в ходе которого будет открываться страница, содержащая параметры email-сообщения для заполнения и дальнейшей отправки email-сообщения с явным указанием учетных данных.

Алгоритм реализации примера

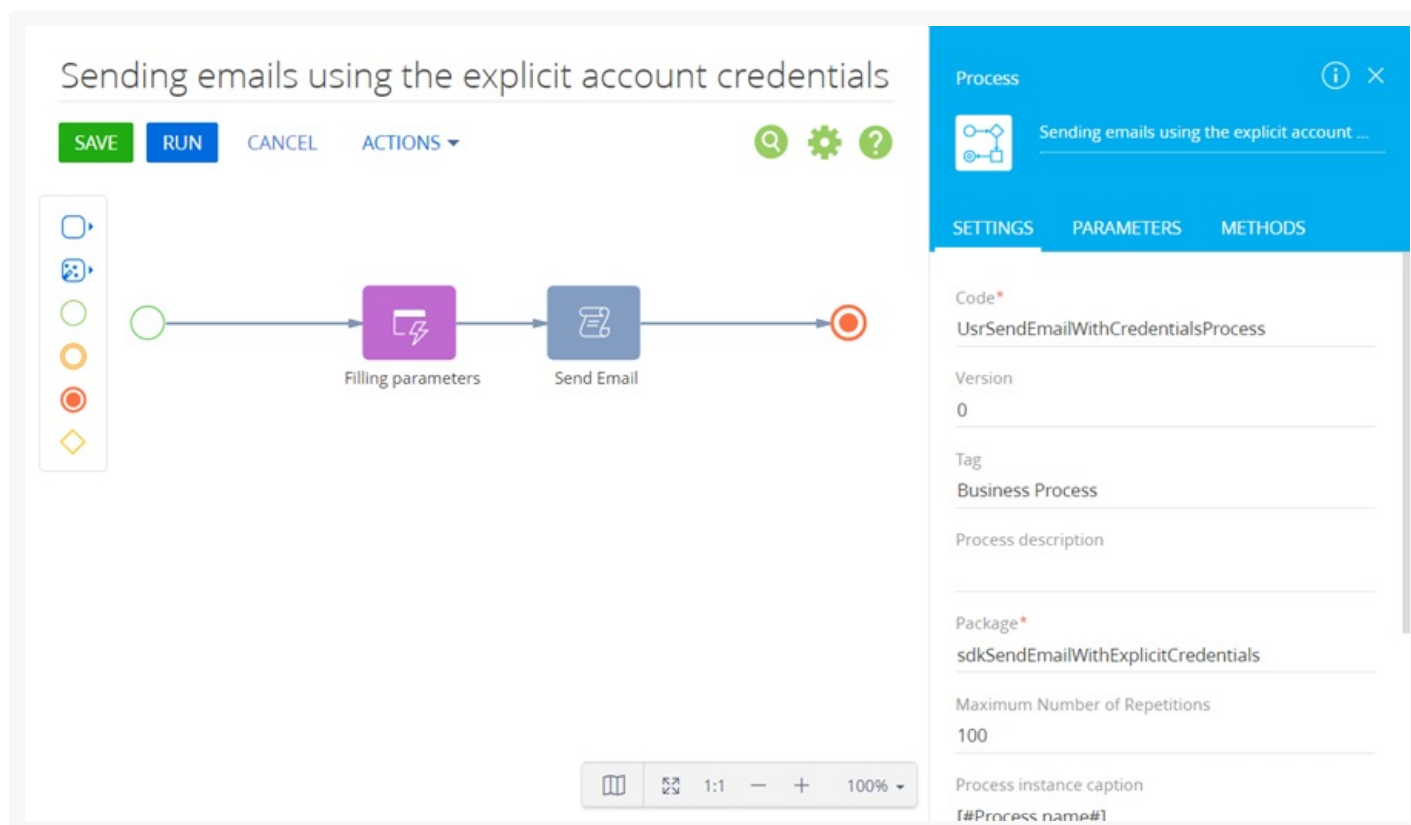
1. Создайте бизнес-процесс

В разделе [*Конфигурация*] ([*Configuration*]) выполните действие [*Добавить*] —> [*Бизнес-процесс*] ([*Add*] —> [*Business process*]).



В открывшемся дизайнере процессов установите следующие значения на панели настройки свойств процесса:

- [Заголовок] ([Title]) — "Sending emails using the explicit account credentials".
- [Код] ([Code]) — "UsrSendEmailWithCredentialsProcess".

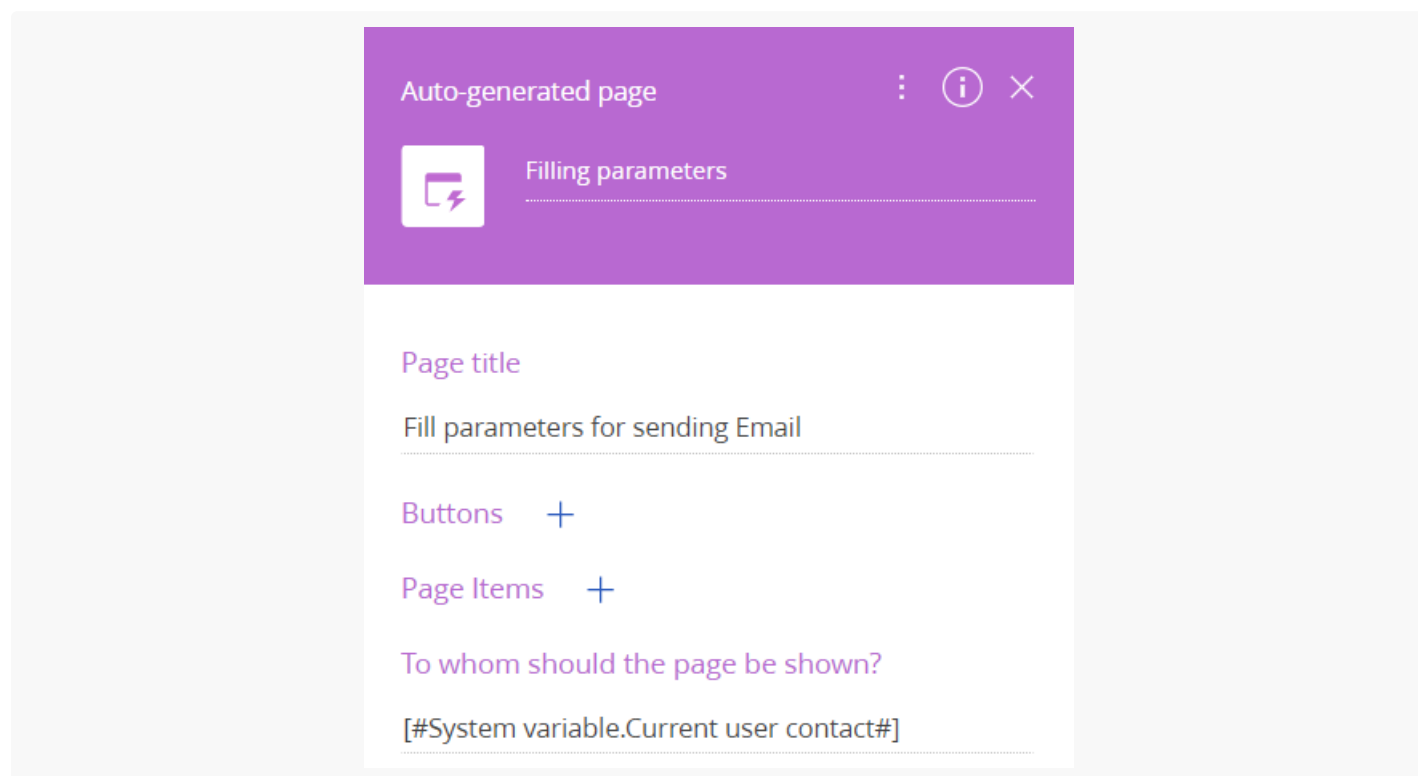


2. Добавьте элемент [Автогенерируемая страница]

С помощью элемента [Автогенерируемая страница] ([Auto-generated page]) в ходе выполнения процесса можно открыть произвольную страницу, которая создана пользователем системы. Для этого элемента добавьте подпись [Filling parameters] и установите следующие свойства:

- [Название страницы] ([Page title]) — "Fill parameters for sending Email".

- [*Кому отобразить страницу?*] ([*To whom should the page be shown?*]) — выберите "Formula" и установите [#System variable.Current user contact#].



3. Добавьте кнопку на страницу

Для добавления кнопки [*Continue*] на страницу, в блоке [*Кнопки*] ([*Buttons*]) нажмите + и введите следующие параметры:

- [*Название*] ([*Caption*]) — "Continue".
- [*Код*] ([*Code*]) — "ContinueButton".
- [*Стиль*] ([*Style*]) — выберите "Green".
- Установите признак [*Активная*] ([*Active*]).
- Установите признак [*Выполняет проверку значений*] ([*Performs value validation*]).

Caption*
 Continue

Code*
 ContinueButton

Style*
 Green

Generate signal

☒ Active

☒ Performs value validation

SAVE CANCEL

Нажмите [Сохранить] ([Save]).

4. Добавьте элементы на страницу

Для добавления на страницу элемента, который будет содержать почтовый ящик отправителя email-сообщения, в блоке [Параметры страницы] ([Page Items]) нажмите **+**, выберите тип [Строковое поле] ([Text field]) и введите следующие параметры:

- [Заголовок] ([Title]) — "Sender Mailbox".
- [Код] ([Code]) — "SenderMailbox".
- Установите признак [Обязательное] ([Required]).

Page Items +

Title*
Sender Mailbox

Code*
SenderMailbox

☐ Is multiline

☒ Required

Value

SAVE CANCEL

Нажмите [Сохранить] ([Save]).

Для добавления на страницу элемента, который будет содержать имя отправителя email-сообщения, в блоке [Параметры страницы] ([Page Items]) нажмите +, выберите тип [Строковое поле] ([Text field]) и введите следующие параметры:

- [Заголовок] ([Title]) — "User Name".
- [Код] ([Code]) — "UserName".
- Установите признак [Обязательное] ([Required]).

Title*
User Name

Code*
UserName

☐ Is multiline

☒ Required

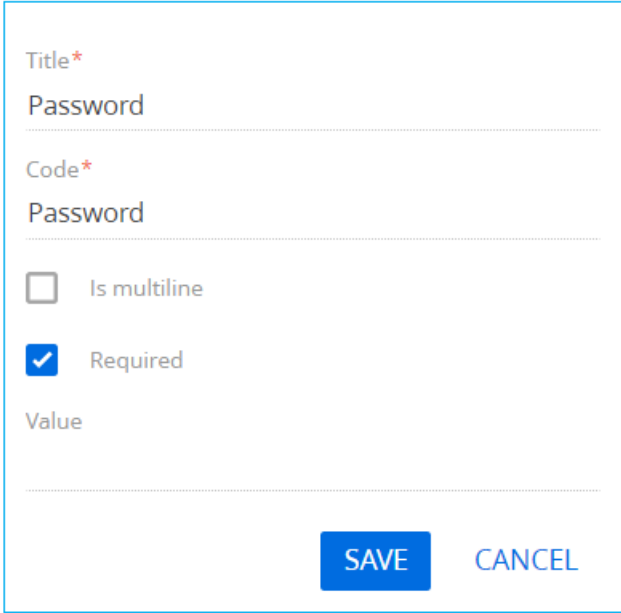
Value

SAVE CANCEL

Нажмите [Сохранить] ([Save]).

Для добавления на страницу элемента, который будет содержать пароль к почтовому ящику отправителя email-сообщения, в блоке [*Параметры страницы*] ([*Page Items*]) нажмите **+**, выберите тип [*Строковое поле*] ([*Text field*]) и введите следующие параметры:

- [*Заголовок*] ([*Title*]) — "Password".
- [*Код*] ([*Code*]) — "Password".
- Установите признак [*Обязательное*] ([*Required*]).



The screenshot shows a configuration window for a 'Text field' element. It contains the following fields and options:

- Title***: A text input field containing the value 'Password'.
- Code***: A text input field containing the value 'Password'.
- Is multiline**: A checkbox that is currently unchecked.
- Required**: A checkbox that is currently checked, indicated by a blue checkmark.
- Value**: A text input field that is currently empty.
- At the bottom right, there are two buttons: **SAVE** (in a blue box) and **CANCEL** (in a light blue box).

Нажмите [*Сохранить*] ([*Save*]).

Для добавления на страницу элемента, который будет содержать адрес почтового сервера отправителя email-сообщения, в блоке [*Параметры страницы*] ([*Page Items*]) нажмите **+**, выберите тип [*Строковое поле*] ([*Text field*]) и введите следующие параметры:

- [*Заголовок*] ([*Title*]) — "Service Url".
- [*Код*] ([*Code*]) — "ServiceUrl".
- Установите признак [*Обязательное*] ([*Required*]).

Нажмите [Сохранить] ([Save]).

Для добавления на страницу элемента, который будет содержать номер порта почтового провайдера отправителя email-сообщения, в блоке [Параметры страницы] ([Page Items]) нажмите **+**, выберите тип [Целое число] ([Integer]) и введите следующие параметры:

- [Заголовок] ([Title]) — "Port".
- [Код] ([Code]) — "Port".

Нажмите [Сохранить] ([Save]).

Для добавления на страницу элемента, который будет содержать криптографический протокол для обеспечения безопасной связи, в блоке [Параметры страницы] ([Page Items]) нажмите **+**, выберите тип [Логическое] ([Boolean]) и введите следующие параметры:

- [Заголовок] ([Title]) — "Use Ssl".
- [Код] ([Code]) — "UseSsl".

Title*

Use Ssl

Code*

UseSsl

Value

SAVE CANCEL

Нажмите [Сохранить] ([Save]).

Для добавления на страницу элемента, который будет содержать почтовый ящик получателя email-сообщения, в блоке [Параметры страницы] ([Page Items]) нажмите **+**, выберите тип [Строковое поле] ([Text field]) и введите следующие параметры:

- [Заголовок] ([Title]) — "Recipient (many recipients separated by semicolon ";")".
- [Код] ([Code]) — "Recipient".
- Установите признак [Обязательное] ([Required]).

Title*

Recipient (many recipients separated by se...

Code*

Recipient

☐ Is multiline

☒ Required

Value

SAVE CANCEL

Нажмите [Сохранить] ([Save]).

Для добавления на страницу элемента, который будет содержать тему email-сообщения, в блоке

[*Параметры страницы*] ([*Page Items*]) нажмите **+**, выберите тип [*Строковое поле*] ([*Text field*]) и введите следующие параметры:

- [*Заголовок*] ([*Title*]) — "Subject".
- [*Код*] ([*Code*]) — "Subject".
- Установите признак [*Обязательное*] ([*Required*]).

The screenshot shows a configuration window for a 'Text field' with the following fields and options:

- Title***: Subject
- Code***: Subject
- Is multiline**: ☐ (unchecked)
- Required**: ☒ (checked)
- Value**: (empty text field)
- SAVE** and **CANCEL** buttons at the bottom right.

Нажмите [*Сохранить*] ([*Save*]).

Для добавления на страницу элемента, который будет содержать тело email-сообщения, в блоке [*Параметры страницы*] ([*Page Items*]) нажмите **+**, выберите тип [*Строковое поле*] ([*Text field*]) и введите следующие параметры:

- [*Заголовок*] ([*Title*]) — "Body".
- [*Код*] ([*Code*]) — "Body".
- Установите признак [*Обязательное*] ([*Required*]).

Title*
 Body
 Code*
 Body
☐ Is multiline
☒ Required
 Value
 SAVE CANCEL

Нажмите [Сохранить] ([Save]).

Для добавления на страницу элемента, который будет содержать тип почтового провайдера отправителя email-сообщения, в блоке [Параметры страницы] ([Page Items]) нажмите **+**, выберите тип [Справочник] ([Selection field]) и введите следующие параметры:

- [Заголовок] ([Title]) — "Type of mail server".
- [Код] ([Code]) — "ServerTypeId".
- Установите признак [Обязательное] ([Required]).
- [Источник данных] ([Data source]) — выберите "Mail service provider type".
- [Представление] ([View]) — выберите "Drop down list".

Title*

Type of mail server

Code*

ServerTypeId

☒ Required

Data source*

Mail service provider type

View*

Drop down list

Value

SAVE

CANCEL

Нажмите [Сохранить] ([Save]).

5. Добавьте элемент [Задание-сценарий]

Свойству [Заголовок] ([Title]) элемента [Задание-сценарий] ([Script task]) присвойте значение "Send Email". Элемент должен выполнять программный код.

Send Email

```
// Создание экземпляра EmailClientFactory.
var emailClientFactory = ClassFactory.Get<EmailClientFactory>(new ConstructorArgument("userConnection"));
// Установка параметров подключения к почтовому сервису.
var credentialConfig = new EmailContract.DTO.Credentials {
    ServiceUrl = Get<string>("ServiceUrl"),
    Port = Get<int>("Port"),
    UseSsl = Get<bool>("UseSsl"),
    UserName = Get<string>("UserName"),
    Password = Get<string>("Password"),
    ServerTypeId = Get<Guid>("ServerTypeId"),
    SenderEmailAddress = Get<string>("SenderMailbox")
};
// Создание экземпляра IEmailSender.
var emailSender = ClassFactory.Get<IEmailSender>(new ConstructorArgument("emailClientFactory", emailClientFactory),
    new ConstructorArgument("userConnection", UserConnection));
// Установка параметров отправляемого сообщения.
var message = new EmailContract.DTO.Email {
```

```

Sender = credentialConfig.SenderEmailAddress,
Receipients = Get<string>("Recipient").Split(';').ToList<string>(),
Subject = Get<string>("Subject"),
Body = Get<string>("Body"),
Importance = EmailContract.EmailImportance.Normal,
// Если тело письма было сформировано в формате HTML.
IsHtmlBody = true,
// Дополнительно можно указать список получателей в копии, в т.ч. скрытых
// CopyReceipients = new List<string> { "user@mail.service" },
// BlindCopyReceipients = new List<string> { "user@mail.service" }
};
// Создание вложения.
var attachment = new EmailContract.DTO.Attachment {
    // Идентификатор вложения.
    Id = Guid.NewGuid().ToString(),
    // Название файла.
    Name = "test.txt",
};
// Данные (используется тестовое значение данных в виде текста).
byte[] data = Encoding.ASCII.GetBytes("some test text");
// Добавление данных во вложение.
attachment.SetData(data);
// Добавление вложения в сообщение.
message.Attachments.Add(attachment);
// Отправка email-сообщения.
emailSender.Send(message, credentialConfig);
return true;

```

6. Добавьте параметры

Для добавления параметра бизнес-процесса, который будет содержать адрес почтового сервера отправителя email-сообщения, на вкладке [*Параметры*] ([*Parameters*]) панели настройки свойств процесса выполните действие [*Добавить параметр*] —> [*Текст*] ([*Add parameters*] —> [*Text*]) и установите следующие свойства параметра:

- [*Заголовок*] ([*Title*]) — "Service Url".
- [*Код*] ([*Code*]) — "ServiceUrl".
- [*Значение*] ([*Value*]) — нажмите ⚡ —> [*Параметр процесса*] ([*Process parameter*]) и выберите элемент процесса "Service Url".

Title*

Service Url

Code*

ServiceUrl

Data type*

Text (500 characters)

Value

[#Filling parameters.Service Url#]

SAVE CANCEL

Нажмите [Сохранить] ([Save]).

Для добавления параметра бизнес-процесса, который будет содержать номер порта почтового провайдера отправителя email-сообщения, на вкладке [Параметры] ([Parameters]) панели настройки свойств процесса выполните действие [Добавить параметр] —> [Целое число] ([Add parameters] —> [Integer]) и установите следующие свойства параметра:

- [Заголовок] ([Title]) — "Port".
- [Код] ([Code]) — "Port".
- [Значение] ([Value]) — нажмите ⚡ —> [Параметр процесса] ([Process parameter]) и выберите элемент процесса "Port".

Title*

Port

Code*

Port

Data type*

Integer

Value

[#Filling parameters.Port#]

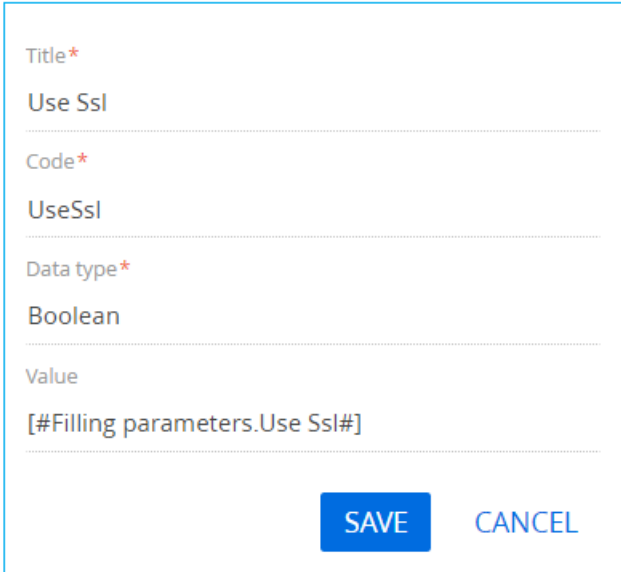
SAVE CANCEL

Нажмите [Сохранить] ([Save]).

Для добавления параметра бизнес-процесса, который будет содержать криптографический протокол

для обеспечения безопасной связи, на вкладке [*Параметры*] ([*Parameters*]) панели настройки свойств процесса выполните действие [*Добавить параметр*] —> [*Логическое*] ([*Add parameters*] —> [*Boolean*]) и установите следующие свойства параметра:

- [*Заголовок*] ([*Title*]) — "Use Ssl".
- [*Код*] ([*Code*]) — "Use Ssl".
- [*Значение*] ([*Value*]) — нажмите ⚡ —> [*Параметр процесса*] ([*Process parameter*]) и выберите элемент процесса "Use Ssl".



The screenshot shows a dialog box for adding a parameter. It contains the following fields and values:

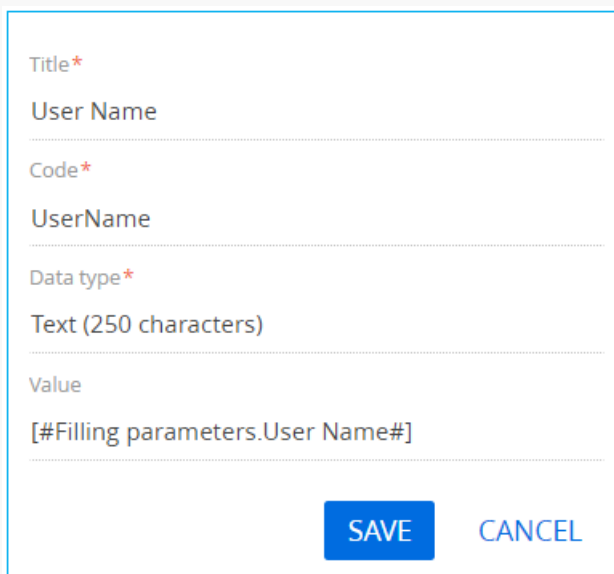
- Title***: Use Ssl
- Code***: UseSsl
- Data type***: Boolean
- Value**: [#Filling parameters.Use Ssl#]

At the bottom right, there are two buttons: **SAVE** (highlighted in blue) and **CANCEL**.

Нажмите [*Сохранить*] ([*Save*]).

Для добавления параметра бизнес-процесса, который будет содержать имя отправителя email-сообщения, на вкладке [*Параметры*] ([*Parameters*]) панели настройки свойств процесса выполните действие [*Добавить параметр*] —> [*Текст*] ([*Add parameters*] —> [*Text*]) и установите следующие свойства параметра:

- [*Заголовок*] ([*Title*]) — "User Name".
- [*Код*] ([*Code*]) — "UserName".
- [*Тип данных*] ([*Data type*]) — выберите "Text (250 characters)".
- [*Значение*] ([*Value*]) — нажмите ⚡ —> [*Параметр процесса*] ([*Process parameter*]) и выберите элемент процесса "User Name".



Title*

User Name

Code*

UserName

Data type*

Text (250 characters)

Value

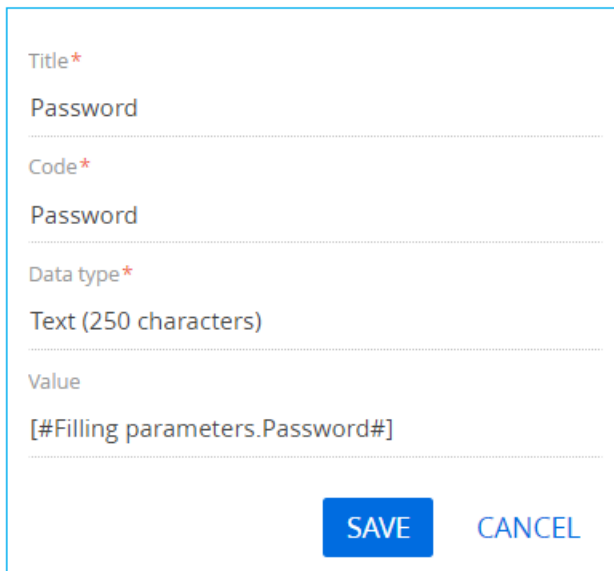
[#Filling parameters.User Name#]

SAVE CANCEL

Нажмите [Сохранить] ([Save]).

Для добавления параметра бизнес-процесса, который будет содержать пароль к почтовому ящику отправителя email-сообщения, на вкладке [Параметры] ([Parameters]) панели настройки свойств процесса выполните действие [Добавить параметр] —> [Текст] ([Add parameters] —> [Text]) и установите следующие свойства параметра:

- [Заголовок] ([Title]) — "Password".
- [Код] ([Code]) — "Password".
- [Тип данных] ([Data type]) — выберите "Text (250 characters)".
- [Значение] ([Value]) — нажмите ⚡ —> [Параметр процесса] ([Process parameter]) и выберите элемент процесса "Password".



Title*

Password

Code*

Password

Data type*

Text (250 characters)

Value

[#Filling parameters.Password#]

SAVE CANCEL

Нажмите [Сохранить] ([Save]).

Для добавления параметра бизнес-процесса, который будет содержать почтовый ящик получателя email-сообщения, на вкладке [*Параметры*] ([*Parameters*]) панели настройки свойств процесса выполните действие [*Добавить параметр*] —> [*Текст*] ([*Add parameters*] —> [*Text*]) и установите следующие свойства параметра:

- [*Заголовок*] ([*Title*]) — "Recipient".
- [*Код*] ([*Code*]) — "Recipient".
- [*Значение*] ([*Value*]) — нажмите ⚡ —> [*Параметр процесса*] ([*Process parameter*]) и выберите элемент процесса "Recipient (many recipients separated by semicolon ";")".

Нажмите [*Сохранить*] ([*Save*]).

Для добавления параметра бизнес-процесса, который будет содержать тип почтового провайдера отправителя email-сообщения, на вкладке [*Параметры*] ([*Parameters*]) панели настройки свойств процесса выполните действие [*Добавить параметр*] —> [*Другое*] —> [*Уникальный идентификатор*] ([*Add parameters*] —> [*Other*] —> [*Unique identifier*]) и установите следующие свойства параметра:

- [*Заголовок*] ([*Title*]) — "Type of mail server".
- [*Код*] ([*Code*]) — "ServerTypeId".
- [*Значение*] ([*Value*]) — нажмите ⚡ —> [*Параметр процесса*] ([*Process parameter*]) и выберите элемент процесса "Type of mail server".

Title*

Type of mail server

Code*

ServerTypeId

Data type*

Unique identifier

Value

[#Filling parameters.Type of mail server#]

SAVE CANCEL

Нажмите [Сохранить] ([Save]).

Для добавления параметра бизнес-процесса, который будет содержать почтовый ящик отправителя email-сообщения, на вкладке [Параметры] ([Parameters]) панели настройки свойств процесса выполните действие [Добавить параметр] —> [Текст] ([Add parameters] —> [Text]) и установите следующие свойства параметра:

- [Заголовок] ([Title]) — "Sender Mailbox".
- [Код] ([Code]) — "SenderMailbox".
- [Тип данных] ([Data type]) — выберите "Text (250 characters)".
- [Значение] ([Value]) — нажмите ⚡ —> [Параметр процесса] ([Process parameter]) и выберите элемент процесса "Sender Mailbox".

Title*

Sender Mailbox

Code*

SendeMailbox

Data type*

Text (250 characters)

Value

[#Filling parameters.Sender Mailbox#]

SAVE CANCEL

Нажмите [Сохранить] ([Save]).

7. Добавьте методы

Для добавления методов бизнес-процесса на вкладке [*Методы*] ([*Methods*]) панели настройки свойств процесса в блоке [*Usings*] нажмите **+** и в поле [*Пространство имён*] ([*Name Space*]) добавьте значение `Terrasoft.Mail.Sender`. Нажмите [*Сохранить*] ([*Save*]).

Таким же способом добавьте следующие пространства имен:

- `Terrasoft.Core.Factories`
- `System.Linq`

Сохраните все изменения в дизайнера процессов.

8. Запуск бизнес-процесса

После запуска бизнес-процесса по кнопке [*Запустить*] ([*Run*]) будет открыта страница для заполнения параметров email-сообщения.

Fill parameters for sending Email

CONTINUE **CLOSE**

Sender Mailbox*

User Name*

Password*

Service Url*

Port 0

Use Ssl ☐

Recipient (many recipients separated by semicolon ";")*

Subject*

Body*

Type of mail server*

Для отправки email-сообщения с явным указанием учетных данных нажмите [*Continue*].