

# Email-сообщения

Общие принципы работы с email-сообщениями

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>Общие принципы работы с email-сообщениями</b>	<b>4</b>
Пользовательская фильтрация нежелательных обращений	4
Работа с цепочками email-сообщений	4
Обработчик макроса в шаблоне email-сообщения	8
<b>Добавить свойство для фильтрации email-сообщений</b>	<b>8</b>
Описание кейса	8
Алгоритм реализации кейса	8
<b>Добавить обработчик макроса шаблона email-сообщения</b>	<b>9</b>
Описание примера	9
Исходный код	10
Пример реализации примера	10

# 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 предустановленные 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

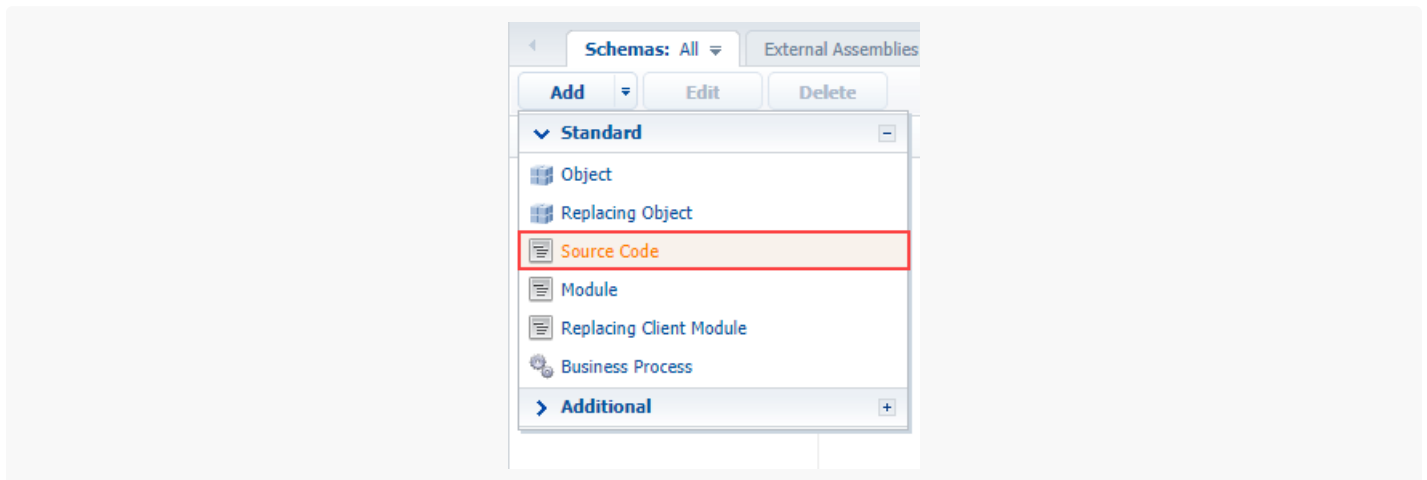
Пакет с реализацией примера можно скачать по [ссылке](#).

## 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:

```

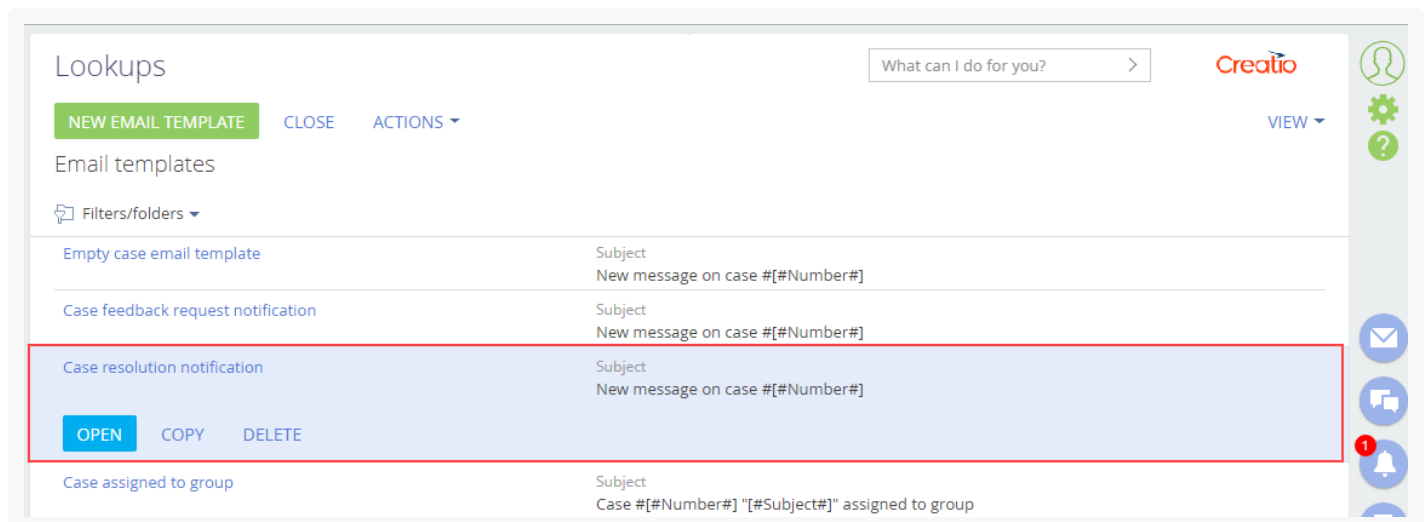
INSERT INTO EmailTemplateMacros(Name, Parentid, ColumnPath)
VALUES (
    'UsrTestStringGenerator',
    (SELECT TOP 1 Id
     FROM EmailTemplateMacros
     WHERE Name = '@Invoke'),
    'Terrasoft.Configuration.UsrTestStringGenerator'
)

```

## 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

