ТЗ Email-рассылки. Общая Часть.

Версия 0

Запрос 1801: Solution: модуль рассылок

[ТЗ Email-рассылки. Общая Часть. 1](#_Toc473274456)

[Версия 1.0.0 1](#_Toc473274457)

[Запрос 1801: Solution: модуль рассылок 1](#_Toc473274458)

[Лист изменений 3](#_Toc473274459)

[Техническое задание 3](#_Toc473274460)

[Email-рассылки: общая часть 3](#_Toc473274461)

[1. Введение 3](#_Toc473274462)

[1.1 Термины. 3](#_Toc473274463)

[1.2 Ссылки 3](#_Toc473274464)

[2. Описание версии 1.0.0 3](#_Toc473274465)

[2.1 Постановка проблемы ( функциональные требования) 3](#_Toc473274466)

[2.2 Концепция решения ( описание функционала) 3](#_Toc473274467)

[3. Производительность 3](#_Toc473274468)

[4. Тестирование 3](#_Toc473274469)

[4.1 Тестовые сценарии (дополнительные) 3](#_Toc473274470)

[4.1.1 Сценарии для DB-части 3](#_Toc473274471)

[4.1.2 Сценарии для APP-части. 3](#_Toc473274472)

[5. Реализация 3](#_Toc473274473)

[5.1 Разграничение прав 3](#_Toc473274474)

[5.2 DB, Пререквизиты 3](#_Toc473274475)

[5.3 DB, Размещение объектов модуля 3](#_Toc473274476)

[5.4 Справочник проектов 3](#_Toc473274477)

[5.4.1 Функция добавления/обновления ( регистрации) проекта рассылки 3](#_Toc473274478)

[5.5 Сообщения 3](#_Toc473274479)

[5.5.1 Временная интерфейсная таблица для добавления сообщений. 3](#_Toc473274480)

[5.5.2 Функция отправки сообщений. 3](#_Toc473274481)

[5.6 Данные отписки 3](#_Toc473274482)

[5.6.1 Функция отписки от рассылки и получения результирующего URL. 3](#_Toc473274483)

[5.6.2 Отмена отписки от рассылки. 3](#_Toc473274484)

[5.7 DB, Вспомогательные методы 3](#_Toc473274485)

[5.8 DB, Пакетные задания 3](#_Toc473274486)

[5.8.1 Job: отправка сообщений 3](#_Toc473274487)

[5.8.2 Пакетное задание “Очистка устаревших сообщений модуля” 3](#_Toc473274488)

[5.9 APP-часть. Отписка. 3](#_Toc473274489)

[6. Развитие 3](#_Toc473274490)

Лист изменений

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
| 19.01.2017 | 1.0.0 | Создание модуля.  Создание общего функционала для реализации рассылок: макросы, отписка. | Лысёнок Е. |

**Техническое задание**

**Email-рассылки: общая часть**

# Введение

Данный документ описывает функционал по общей части для создания Email-рассылок.

Префикс модуля - *nsc*.

## Термины.

БД – База данных

NewsletterCore – наименование модуля

## Ссылки

[1] Ссылка в SVN на модуль:

SVN: Jep/Module/Newsletter

[2] Стандарт разработки исходного кода:

Svn: Management/Module/Agreement/Trunk/Doc/SourceCode/Стандарт разработки исходного кода (Oracle).doc

# Описание версии 1.0.0

## Постановка проблемы ( функциональные требования)

Необходимо обеспечить возможность простого добавления новых рассылок с минимально необходимой информацией с возможностью отписки. Пользователем модуля является разработчик, которому поставлено задание разработать Email-рассылку клиентам с заданными определёнными критериями отбора.

## Концепция решения ( описание функционала)

Для использования модуля NewsletterCore разработчик задаёт расписание, отправителя, шаблон, процедуру добавления выборки во временную таблицу модуля NewsletterCore, а такжеабсолютные адреса двух страниц, на которые будет перенаправляться пользователь при успешной и неуспешной отписке, соответственно. Этих данных должно быть достаточно для работы рассылки.

Технически расписание, вызов процедуры задаются в батчах прикладных модулей. Часть батча ( job) содержится в модуле NewsletterCore. Процедура сводится к формированию выборки и вставке во временную таблицу модуля. Во временной таблице должна быть возможность задавать значения макропеременных в строке.

Шаблон, отправитель сообщения и страницы результата отписки задаются при регистрации проекта.

Шаблон сообщения может содержать макропеременные. Должны быть также предопределённые макропеременные для возможности указания в шаблоне ссылки для отписки, важной из которых является HASH-ключ сообщения, для которого в БД хранится адрес электронной почты и проект. Данный ключ является достаточным для выполнения отписки.

Следует также разработать APP-часть ( сервлет), которая будет вызывать функцию отписки при передаче ей GET-параметра ключа и перенаправлять клиента на страницы успешного или неуспешного ( например, при некорректном HASH-ключе) результата отписки.

# Производительность

Время отклика API базы данных не должно превышать 1 секунду.

Время отклика GUI на операциях с API базы данных не должно превышать 2 секунды.

# Тестирование

Автоматизированные тесты создаются до написания функционального кода, как на уровне базы данных, так и на уровне GUI.

Необходимо создать функциональные тесты на каждый метод API базы данных.

Для методов поиска данных необходимо создать тесты производительности.

Все автоматизированные тесты объединяются в один тест модуля (один на уровне базы данных и один на уровне GUI), запускаемый одной командой, который в обязательном порядке должен проходить при выпуске релиза (создании новой версии) модуля.

Все тесты должны запускаться без обязательной установки средств разработки (должна быть возможность запуска тестов Системными Аналитиками и Сопровождением). Допускается необходимость наличия инструментов установки приложений.

## Тестовые сценарии (дополнительные)

Сценарии, пополняемые (в обязательном порядке) при наступлении инцидента, или в случае необходимости тщательной проверки сложного и/или критичного функционала.

### Сценарии для DB-части

#### Подготовка сценариев:

В тестовом пакете должна быть процедура pkg\_NewsletterCoreTest.simpleNewsletter()

добавления двух записей во временную таблицу *nsc\_message\_tmp* со значениями полей

с предварительной очисткой nsc\_message\_tmp.

1)

sender=null

recipient = [fakeemail1@oramake.com](mailto:fakeemail1@oramake.com%20)

message\_text=null

macro\_data=

”macro1=macro\_1\_value

macro2=macro\_2\_value”

is\_html\_message=null

2)

sender=null

recipient=fakeemail2@oramake.com

message\_text=’fake body 2’

macro\_data=null

is\_html\_message=1

Создаётся процедура loadTestBatch, которая загружает батч ( с помощью пакета pkg\_SchedulerLoad):

batch.xml

<?xml version="1.0" encoding="Windows-1251"?>

<batch short\_name="NewsletterCoreTestBatch">

<name>Тестовый батч NewsLetter </name>

<content id="1" job="initialization" module="Scheduler"/>

<!— В job вызывается процедура pkg\_NewsletterCoreTest.simpleNewsletter()-->

<content id="2" job="simple\_newsletter">

<condition id="1">true</condition>

</content>

<content id="3" job="send\_message" module=”NewsletterCore”>

<condition id="2">true</condition>

</content>

</batch>

Создаётся процедура, которая выполняет тестовый батч в той же сессии и удаляет его

pkg\_NewsletterCore.runTestBatch

c необязательным параметром skipDeletionFlag, позволяющим пропускать удаление батча.

Выполняются процедуры:

pkg\_NewsletterCore.clearUnsubscribe( ‘fakeemail1@newsletter.oramake.com’);

pkg\_NewsletterCore.clearUnsubscribe( ‘fakeemail2@newsletter.oramake.com’);

Из справочника проектов удаляются проекты

‘TestProjectShortName’, ‘TestProjectShortName2’ ( и связанные записи)

#### Сценарий 1. Создание рассылки

Выполняется pkg\_NewsletterCoreTest.loadTestBatch

Регистрируется проект рассылки

pkg\_Newsletter.mergeProject(

projectShortName => ‘TestProjectShortName’

, projectName => ‘Тестовый проект’

, projectNameEn => ‘Test project’

, batchShortName => ‘NewsletterCoreTestBatch’

, messageTextTemplate => ‘fake\_body\_1 $(macro1) $(macro2) $(projectShortName) $(email) *http://www.create\_your\_newsletter.com/NewsletterCore/Unsubscribe?messageKey=*$(messageKey)’

, sender => ‘fakesender@oramake.com’

);

Выполняется pkg\_NewsletterCoreTest.runTestBatch

После выполнения теста данные сообщения независимо от результатов теста следует отменить средствами модуля Mail.

**Результат:**

Cозданы два сообщения в ml\_message со значениями полей:

1)\_ message\_body=fake\_body\_1 macro\_1\_value macro\_2\_value TestProjectShortName fakeemail1@oramake.com  *http://www.create\_your\_newsletter.com/NewsletterCore/Unsubscribe?messageKey=*<строка из не более 50 символов>

recipient= fakeemail1@oramake.com

sender=fakesender@oramake.com

2) message\_body=fake\_body\_2

recipient= fakeemail2@oramake.com

sender=fakesender@oramake.com

#### Сценарий 2. Создание рассылки из шаблона в файле

Данный сценарий предполагает опцию TestRootDirectory, создаваемую при проливке install-test.

При помощи TEST\_ROOT\_DIRECTORY

( Развитие: данные тестовый путь может быть корневым для всех модулей и браться из тестовой части модуля File и расположен, возможно, в проектируемом модуле OraMakeConfig)

Если опция не задана, то тест сразу выдаёт FAILED.

Выполняется pkg\_NewsletterCoreTest.loadTestBatch

Регистрируется проект рассылки

pkg\_Newsletter.mergeProject(

projectShortName => ‘TestProjectShortName’

, projectName => ‘Тестовый проект’

, projectNameEn => ‘Test project’

, batchShortName => ‘NewsletterCoreTestBatch’

, messageTemplatePath => <Значение опции TestRootDirectory>

, sender => ‘fakesender@oramake.com’

);

В динамическом SQL средствами модуля File создаётся / пересоздаётся файл fake\_template.txt

c содержимым

fake\_body\_1 $(macro1) $(macro2) $(projectShortName) $(email) *http://www.create\_your\_newsletter.com/NewsletterCore/Unsubscribe?messageKey=*$(messageKey)

Выполняется pkg\_NewsletterCoreTest.runTestBatch.

После выполнения теста данные сообщения независимо от результатов теста следует отменить средствами модуля Mail.

**Результат:**

Cозданы два сообщения в ml\_message со значениями полей:

1)\_ message\_body=fake\_body\_1 macro\_1\_value macro\_2\_value TestProjectShortName fakeemail1@oramake.com  *http://www.create\_your\_newsletter.com/NewsletterCore/Unsubscribe?messageKey=*<строка из не более 50 символов>

recipient=fakeemail1@oramake.com

sender=fakesender@oramake.com

2) message\_body=fake\_body\_2

recipient=fakeemail2@oramake.com

sender=fakesender@oramake.com

#### Сценарий 3. Отписка.

Выполняется pkg\_NewsletterCoreTest .loadTestBatch

Регистрируется проект рассылки

pkg\_Newsletter.mergeProject(

projectShortName => ‘TestProjectShortName’

, projectName => ‘Тестовый проект’

, projectNameEn => ‘Test project’

, batchShortName => ‘NewsletterCoreTestBatch’

, messageTextTemplate => ‘$(messageKey)’

);

Выполняется runTestBatch( skipDeletionFlag => 0);

Извлекается созданное сообщение и выполняется

pkg\_NewsletterCore.unsubscribe( messageKey => <значение поля message\_body для fakemail1@oramake.com>)

Выполняется runTestBatch

**Результат:**

Cозданы три сообщения в ml\_message со значениями полей:

1) recipient= fakeemail1@oramake.com

2) recipient= fakeemail2@oramake.com

3) recipient= fakeemail2@oramake.com

#### Сценарий 4. Проверка очистки устаревших сообщений.

Выбирается одна запись по тестовым данным из nsc\_message

date\_ins устанавливается в 1900-01-01.

Вызывается процедура pkg\_NewsletterCore.clearOldMessage( dateBefore => date '1900-01-01' + <1 секунда>)

**Результат:**

Запись отсутствует в таблице.

### Сценарии для APP-части.

#### Подготовка сценариев при задании URL.

Выполняется pkg\_NewsletterCoreTest.loadTestBatch

Регистрируется проект

pkg\_Newsletter.mergeProject(

projectShortName => ‘TestProjectShortName’

, projectName => ‘Тестовый проект’

, projectNameEn => ‘Test project’

, batchShortName => ‘NewsletterCoreTestBatch’

, messageTextTemplate => ‘$(messageKey)’

, unsubscribeSuccessUrl => ‘<URL тестовой статической страницы 1>’

, unsubscribeFailUrl => ‘<URL тестовой статической страницы 2>’

);

Выполняется pkg\_NewsletterCoreTest.runTestBatch

В переменную messageKey записывается значение одного из созданных

сообщений nsc\_message.

#### Сценарий 5. Успешная отписка.

Происходит переход по URL страницы

Unsubscribe?messageKey=<значение переменной messageKey>

Результат:

Пользователь переводится на страницу <URL тестовой статической страницы 1>

#### Сценарий 6. Неуспешная отписка.

Происходит переход по URL страницы

Unsubscribe?messageKey=<fake-значение >

Результат:

Пользователь переводится на страницу <URL тестовой статической страницы 2>

#### Сценарий 5. Успешная отписка без результирующего URL

Обновление проекта:

pkg\_Newsletter.mergeProject(

projectShortName => ‘TestProjectShortName’

, projectName => ‘Тестовый проект’

, projectNameEn => ‘Test project’

, batchShortName => ‘NewsletterCoreTestBatch’

, messageTextTemplate => ‘$(messageKey)’

, unsubscribeSuccessUrl =>null

, unsubscribeFailUrl => null

);

Происходит переход по URL страницы

Unsubscribe?messageKey=<значение переменной messageKey>

Результат:

Пользователь видит сообщение

«Вы успешно отписались от рассылки»

# Реализация

## Разграничение прав

Для разграничения прав необходимо создать следующие роли:

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Наименование** | **Роль**  **(краткое наименование)** | **Описание** |
|  |  |  | В текущей версии не используется. |

## DB, Пререквизиты

Функционал данного модуля зависит от следующих модулей:

* *AccessOperator;*
* *Common.*

Функционал данного модуля напрямую зависит от следующих модулей:

* *Common ( определение отправителя по-умолчанию);*
* *Mail ( основной функционал);*
* *Option ( для тестирования).*

*Опционально можно использовать*

* *Scheduler ( для создания батчей для рассылок);*
* *File ( для размещения шаблона).*

## DB, Размещение объектов модуля

Объекты модуля должны располагаться в схеме с общими модулями базы данных (например: *Common*).

Необходимо предоставить возможность выдачи прав и создания синонимов для возможности работы с пакетом модуля заданному пользователю базы данных.

## Справочник проектов

В качестве вспомогательной справочной информации используются следующие сущности:

* nsc\_project – проекты рассылки

Справочник является внутренним ( закрытым) для модуля. Т.е. логически запрещено использовать/ ”знать” о существовании данного справочника другим модулям.

Таблица nsc\_project:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Название поля** | **Тип** | **Nullable** | **Комментарий** |
| 1 | *project\_id* | *Integer* | *not null* | *Primary Key*. Id проекта. |
| 2 | *project\_short\_name* | *varchar2(30)* | *not null* | Уникальное короткое наименование проекта. |
| 3 | *project\_name* | *varchar2(255)* | *not null* | Наименование проекта на языке по умолчанию. |
| 4 | *project\_name\_en* | *varchar2(255)* |  | Наименование проекта на английском языке. |
| 5 | *description* | *Varchar2(4000)* |  | Описание проекта |
| 6 | *batch\_short\_name* | *varchar2(50)* |  | Наименование батча на английском языке. Без внешнего ключа, чтобы уменьшить зависимости ( см. реализацию mergeProject). |
| 7 | *message\_text\_template* | *clob* |  | Шаблон сообщения, содержащий макросы. |
| 8 | *message\_template\_path* | *varchar2(1000)* |  | Путь к шаблону сообщения. Взаимноисключаемое с message-text\_template поле. |
| 7 | *unsubscribe\_success\_url* | *varchar2(1000)* |  | URL страницы при успешной отписке |
| 8 | *unsubscribe\_fail\_url* | *varchar2(1000)* |  | URL страницы при неуспешной отписке |
| 9 | *date\_ins* | *Date* | *not null* | Дата создания записи.  Значение по умолчанию: *sysdate*. |
| 10 | *operator\_id* | *Integer* | *not null* | Пользователь, создавший запись.  Значение по умолчанию: текущий пользователь. |

### Функция добавления/обновления ( регистрации) проекта рассылки

Функция добавления/обновления проекта рассылки mergeProject.

Параметры:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | | **Наименование** | **Тип** | **Обязательность** | **Комментарий** |
| *1* | *projectShortName* | *varchar2* | Да | Уникальное короткое уникальное наименование проекта ( только латинские буквы, все слова слитно, каждое, включая первое, с заглавной буквы) |
| *2* | *projectName* | *varchar2* | Да | Наименование на языке по-умолчанию |
| *3* | *projectNameEn* | *varchar2* | Да | Наименование на английском языке |
| *4* | *batchShortName* | *varchar2* | Нет | Короткое уникальное наименование батча |
| *5* | *description* | *varchar2* | Нет | Описание проекта |
| *6* | *messageTextTemplate* | *varchar2* | Нет | Текст шаблона сообщений, который может содержать макросы. |
| *7* | *messageTemplatePath* | *varchar2* | Нет | Файловый путь к тексту шаблона сообщения ( параметр, взаимноисключаемый с messageTemplatePath) |
| *8* | *isHtmlMessage* | *varchar2* | Нет | Флаг HTML-сообщения по-умолчанию |
| *9* | *unsubscribeSuccessUrl* | *varchar2* | Нет | URL страницы при успешной отписке |
| *10* | *unsubscrubeFailUrl* | *varchar2* | Нет | URL страницы при неуспешной отписке |
| *11* | *operatorId* | *Integer* | Нет | Id оператора, меняющего запись. По-умолчанию текущий. |

Функция обновляет справочник проектов и возвращает число изменённых/созданных записей.

В случае задания batchShortName в динамическом SQL происходит проверка существования батча в v\_sch\_batch.

## Сообщения

Таблица nsc\_message

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Название поля** | **Тип** | **Nullable** | **Комментарий** |
| 1 | *message\_key* | *varchar2(50)* | *not null* | *Primary Key*. Уникальный ключ сообщения.  Генерируется с помощью последовательности и системной функции *dbms\_obfuscation\_toolkit.md5* |
| 2 | *project\_id* | *Integer* | *not null* | Id проекта рассылки. Внешний ключ. |
| 3 | *recipient* | *varchar2(255)* |  | Получатель(и) сообщения |
| 4 | *message\_id* | *Integer* |  | message\_id модуля Mail ( внешний ключ не обязательно создавать) |
| 5 | *date\_ins* | *Date* | *not null* | Дата создания записи.  Значение по умолчанию: *sysdate*. |

Следует создать уникальный индекс по message\_id

Следует создать индекс по date\_ins

### Временная интерфейсная таблица для добавления сообщений.

Таблица уровня транзакции nsc\_message\_tmp:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Название поля** | **Тип** | **Nullable** | **Комментарий** |
| 1 | recipient | *varchar2(1000)* | *not null* | Получатель(и) сообщения. |
| 2 | sender | *varchar2(1000)* |  | Отправитель сообщения ( по-умолчанию используется отправитель из проекта). Если отправитель не может быть установлен, выбрасыватеся исключение. |
| 3 | *message\_text* | *clob* |  | Тело сообщения. Если не задано и не задан шаблон –выбрасывается исключение. |
| 4 | *macro\_data* | *varchar2(4000)* |  | Значения макропеременных в формате: <переменная>=<значение>;<переменная>=<значение>. Пары разделены либо ';' либо концом строки ( либо CRLF). В случае если значение содержит служебные символы “=”, пробел, конец строки или возврат каретки, то значение должно быть заключено в двойные кавычки. В случае двойных кавычек в значении они должны быть продублированы. По-умолчанию макропеременных нет. Не должно быть задано одновременно с message\_text. |
| 5 | is\_html\_message | *number(1,0)* |  | Интерпретировать ли тело сообщения как HTML ( по-умолчанию нет) |

### Функция отправки сообщений.

Функция отправки сообщений pkg\_NewsletterCore.sendMessage

Параметры:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Наименование** | **Тип** | **Обязательность** | **Комментарий** |
| 5 | *projectShortName* | *varchar2* | Нет | Короткое наименование проекта |
| 6 | *batchShortName* | *varchar2* | Нет | Короткое наименование батча ( взаимноисключаемый с projectShortName) |

Реализация функции:

* Определяется проект projectShortName, если задан batchShortName
* Для каждой записи nsc\_message\_tmp проверяется наличие соответствующей записи в nsc\_unsubsribe ( см. ниже). Если соответствие найдено – запись игнорируется.
* Определяется используемый шаблон для сообщения – в зависимости от задания message\_text в таблице в приоритете, либо шаблона, заданного в nsc\_project. Если шаблон задан в виде пути, то происходит попытка чтения в динамическом SQL для исключения статической зависимости от модуля File. Если шаблон не определён ( пуст), то выдаётся исключение.
* Происходит подмена предопределённых макросов
* $(email ) – получатель(получатели) сообщения
* $(sender) -отправитель
* $(projectShortName) – короткое наименование проекта
* $(messageKey) – уникальный ключ для добавления в nsc\_message
* Происходит разбор macro\_data ( см. описание таблицы) и подстановка в шаблоне.
* Происходит проверка замены всех макросов вида $(имя макроса) в сообщении
* Происходит передача данных модулю Mail с помощью функции sendMessage ( транзакция не фиксируется в данной реализации). Следует учитывать параметр isHtmlMessage и флаг is\_html\_message во временной таблице ( в приоритете)
* Создаётся запись в nsc\_message

## Данные отписки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Название поля** | **Тип** | **Nullable** | **Комментарий** |
| 1 | recipient | *varchar2(1000)* | *not null* | Получатель(и) сообщения. Нормализуется приведением к нижнему регистру. Следует проверять вхождение “@”, отсутствие пробельных символов, присутствие домена первого уровня. |
| 2 | project\_short\_name | *varchar2(30)* | *not null* | Короткое наименование проекта. Внешний ключ. Отсутсвие задания проекта означает для всех. |

Следует создать уникальный ключ на поля recipient, project\_short\_name.

### Функция отписки от рассылки и получения результирующего URL.

Функция

pkg\_NewsletterCore.unsubscribe

Параметры:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Наименование** | **Тип** | **Обязательность** | **Комментарий** |
| 1 | *resultUrl* | *varchar2* |  | Выходной параметр. Результирующий URL. |
| 2 | *messageKey* | *varchar2* | Да | Уникальный ключ сообщения. |

Возвращаемое функцией значение:

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Название поля** | **Тип** | **Комментарий** |
| 1 | *-* | *Boolean* | True, если отписка существовала или удалось добавить отписку  False – иначе |

Реализация.

По уникальному ключу происходит поиск записи в nsc\_message.

Если запись не найдена, функция возвращает false.

Данные добавляются в таблицу nsc\_unsubsribe, если их не существовало. В случае существования действий не происходит. Электронная почта нормализуется, приводится к нижнему регистру при добавлении в таблицу.

Результирующий URL берётся как unsubscribe\_fail\_url или unsubscribe\_success\_url проекта в зависимости от результата функции. Результат URL может быть null ( тогда сервлет использует собственный ответ без перенаправления).

### Отмена отписки от рассылки.

Функция

pkg\_NewsletterCore.clearUnsubsribe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Наименование** | **Тип** | **Обязательность** | **Комментарий** |
| 1 | *recipient* | *varchar2* | Да | Получатель сообщения для отписки. |

Удаляет данные из таблицы nsc\_unsubscribe для данного получателя.

## DB, Вспомогательные методы

Функция очистки данных сообщений.

pkg\_NewsletterCore.clearOldMessage

Параметры:

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Наименование** | **Тип** | **Комментарий** |
| 1 | *dateBefore* | *Date* | Дата, до которой удалять данные сообщений |

Возвращаемое функцией значение:

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Название поля** | **Тип** | **Комментарий** |
| 1 | *-* | *Integer* | Число удаленных записей |

Удаляет данные в таблице nsc\_message.

## DB, Пакетные задания

### Job: отправка сообщений

В модуле должна быть реализация job: send\_message

вызывающего процедуру pkg\_NewsletterCore.sendMessage.

### Пакетное задание “Очистка устаревших сообщений модуля”

NewsletterCoreClearOldMessage

Параметры:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Наименование** | **Тип** | **Комментарий** | **Значение по-умолчанию** |
| 1 | *DaysBefore* | *Integer* | Количество дней до текущей даты, до которой удалять сообщения. | 60 |

Условия запуска:

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Условие** | **Значение** | **Комментарий** |
| 1 | Запуск | *Каждый день в 4:51* |  |
| 2 | Количество попыток | *2* |  |
| 3 | Интервал между попытками | *10 минут* |  |

Реализация: вызывает процедуру pkg\_NewsletterCore.clearOldMessage с параметром

sysdate - <DaysBefore>.

## APP-часть. Отписка.

Следует реализовать сервлет Unsubscribe, выполняющий следующие действия при вызове:

* Вызов функции pkg\_NewsletterCore.unsubscribe с GET-параметром messageKey в качестве параметра messageKey
* Если возвращаемый параметр resultUrl не пуст, то происходит перенаправление на возвращаемый URL
* Иначе ( если resultUrl пуст) сервлетом выводятся сообщения в зависимости от результата функции:

«Вы успешно отписались от рассылки», если true

«Ссылка не актуальна», если false

# Развитие

Сделать возможность подписки на рассылку, в том числе из интерфейса.

Сделать возможность связи батчей и проектов: многие ко многим.

Выделить сущность страницы результата из сущности проекта.