# Mail API

Рассылка почты осуществляется при помощи компонента TDBMailStore.

**Сводная таблица методов и свойств TDBMailStore**

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя** | **Тип данных** | **Описание** | **Значение по умолчанию или**  **возвращаемое значение** |
| **Свойства** | | | |
| ClassID | int | ID класса к которому привязан шаблон рассылки. Требуется указывать только при создании нового шаблона перед вызовом метода Design. Значение принимается только при TemplateID = 0. | 0 |
| TemplateID | int | ID шаблона рассылки из таблицы \_STMailTemplates базы данных. Требуется указывать перед вызовом методов Design и SendTemplate | 0 |
| ShowEditorOnSend | bool | Показывать или нет редактор при рассылке | True  (Показывать) |
| SendOnlySelected | bool | Обрабатывать все или выделенные записи источника рассылки | True  (Только выделенные) |
| SourceDataSet | TX2Query | Источник данных рассылки. Требуется указывать перед вызовом SendTemplate | NULL  (Не инициализирован) |
| SendCompleteSQLProc | string | Имя SQL процедуры которая будет обрабатывать результаты рассылки |  |
| **Методы** | | | |
| Design | bool | Вызов редактора шаблона. Для создания шаблона перед вызовом необходимо указать ClassID, к которому будет привязан шаблон. Для редакции шаблона перед вызовом необходимо указать TemplateID. | True – если шаблон был сохранён  False – если пользователь закрыл редактор без сохранения |
| SendTemplate | bool | Рассылка шаблона. Перед вызовом необходимо указать сначала TemplateID и SourceDataSet. Установка SourceDataSet при TemplateID = 0 приведёт к генерации ошибки. | True – если рассылка добавлена в очередь рассылки |

## Примеры использования

**Создание нового шаблона:**

DBMailStore = this.create(“TDBMailStore”,this,”DBMailStore”)

DBMailStore.ClassID = 5

DBMailStore.Design

**Редакция существующего шаблона:**

DBMailStore = this.create(“TDBMailStore”,this,”DBMailStore”)

DBMailStore.TemplateID = 1

DBMailStore.Design

**Отправка шаблона:**

DBMailStore = this.create(“TDBMailStore”,this,”DBMailStore”)

DBMailStore.TemplateID = 1

DBMailStore.SourceDataSet = this(“MyDataSet”)

DBMailStore.ShowEditorOnSend = false

DBMailStore.SendOnlySelected = false

DBMailStore.SendCompleteSQLProc = “[dbo].[MyProcedure]”

DBMailStore.SendTemplate

## Хранимая процедура для обработки результатов отправки

Для обработки результатов рассылки на стороне SQL сервера необходимо создать процедуру вида [dbo].[MyProcedure] @TemplateID int, @XML varchar(max). Названия параметров изменять нельзя, так как это приведёт к генерации ошибки.

**Параметры процедуры:**

@TemplateID – Id шаблона из таблицы \_STMailTemplates

@XML – результирующая таблица в формате xml.

**Поля результирующей таблицы:**

ObjID – ID записи таблицы, к которой привязан шаблон (ObjectID, которые были выбраны в SourceDataSet). Id класса можно извлечь из таблицы \_STMailTemplates с помощью параметра процедуры @TemplateID (см. пример).

Sended – результат отправки. 1 – письмо принято сервером, 0 – письмо отвергнуто сервером, NULL – отправка не производилась (такая ситуация может быть из-за сбоя SMTP-сервера в процессе рассылки).

ErrDesc – описание ошибки возвращённое сервером SMTP. Заполнено только в том случае, когда Sended = 0

Пример:

CREATE procedure [dbo].[TestSendMail] @TemplateID int, @XML varchar(max)

AS

DECLARE @ClassID int, @idoc int

DECLARE @SendResult TABLE([ObjID] int,[Sended] bit,[ErrDesc] varchar(255))

SELECT @ClassID = ClassID FROM dbo.\_STMailTemplates WHERE ID = @TemplateID

EXEC sp\_xml\_preparedocument @idoc OUTPUT, @XML

INSERT @SendResult

SELECT \* FROM OPENXML (@idoc, '/ROOT/SENDRESULT',1)

WITH ([ObjID] int, [Sended] bit,[ErrDesc] varchar(255))

EXEC sp\_xml\_removedocument @idoc