



JangoSMTP Tutorial

Sending Transactional Emails Using the SendTransactionalEmail API method

Overview:

JangoSMTP provides a robust **mechanism for sending single-recipient transactional emails**. Examples of transactional emails are **order confirmations, appointment reminders, “be my friend” requests on social networking sites**, and other single-recipient email messages that result to facilitate a transaction between the sending organization and the recipient.

Transactional emails can be sent via the **API/Web Service method, SendTransactionalEmail**. There is a [test form for this method](#) that can be used to send test transactional emails, and there is also the [technical specification](#) for the method. There is also the option to enable Marketers to dictate the content of transactional emails. To do this, sign up for a [JangoMail account](#) and use the [SendTransactionalEmailFromTemplate](#) API method.

JangoSMTP's transactional messaging reporting allows a user to **view reporting data, such as Opens, Clicks, Unsubscribes, Bounces, and Complaints** organized by groups of transactional emails. The reporting data can be retrieved in the JangoSMTP web interface or with the Web Service's reporting methods.

How do I use this feature?

To send transactional emails through the API, you must have some programming experience using a Web-based API. Here is a [tutorial on how to use the JangoMail API](#). Calling the SendTransactionalEmail API method is similar to calling the the SendMassEmail API method in JangoMail. While SendMassEmail is meant for sending email campaigns to multiple recipients (from two recipients to millions of recipients), the SendTransactionalEmail method is designed for sending single recipient email messages.

Benefits of sending transactional emails through JangoSMTP:

Typically, organizations send transactional emails either manually, using a desktop or web-based email program or programatically through server-side scripting on a web page. The email message is routed through an SMTP server to the recipient.

There are many advantages to using JangoMail's API methods or the JangoSMTP Relay Service:

1. **Authentication:** The transactional emails are **signed with DomainKeys and DomainKeys Identified Mail (DKIM)**.
2. **Reporting:** The transactional emails are tagged such that **Open Tracking and Click Tracking data is available** on the email messages.
3. **Compliance:** The transactional emails **are checked against your account's unsubscribe and bounce lists**, ensuring that emails are not sent to recipients that

do not want emails from your organization. This saves you money and makes sure that you are following CAN-SPAM requirements.

4. **Deliverability:** The email messages can be **routed through our high-reputation sending IP addresses**. Additionally, if your account qualifies, your email messages may be routed through servers whitelisted with the [Return Path Certification Program](#) (additional cost may apply).

Calling the SendTransactionalEmail method:

The following are the **required input parameters** for the method:

```
Username  
Password  
FromEmail  
FromName  
ToEmailAddress  
Subject  
MessagePlain  
MessageHTML  
Options
```

Each input parameters is explained in detail in the [technical specification](#). The "Options" parameter allows the user to set tracking and other details of the email message, such as Open Tracking, Click Tracking, a CC address, a BCC address, a custom ReplyTo address, Attachments, a Transactional Group in which to categorize this email message, and other details.

Return Value: When successfully called, the method will return a string such as:

```
"0  
SUCCESS  
90293283"
```

The return value is a 0 followed by a newline character followed by the word "SUCCESS" followed by another newline character followed by a unique ID number representing this transactional email message.

The SendTransactionalEmail method can be called via an HTTP Form Post, an HTTP GET, or via a SOAP call. A [test form is available that calls the method via an HTTP Form Post](#).

Screenshot of Test Form

SendTransactionalEmail	
Parameter	Value
Username:	<input type="text" value="TestUser"/>
Password:	<input type="text" value="Jk3#98!fj"/>
FromEmail:	<input type="text" value="baker@browniekitchen.com"/>
FromName:	<input type="text" value="The Brownie Kitchen"/>
ToEmailAddress:	<input type="text" value="joesmith@yahoo.com"/>
Subject:	<input type="text" value="Thank you for your cookie order"/>
MessagePlain:	<pre>Dear Joe, Thank you for your order. Attached is a detailed PDF with your order information. Thanks, The Brownie Kitchen</pre>
MessageHTML:	<pre><HTML> <P> Dear Joe,

 Thank you for your order. Attached is a detailed PDF with your order information.

 Thanks,
 The Brownie Kitchen </P> </HTML></pre>
Options:	<pre>OpenTrack=True, ClickTrack=True, Attachment1=Order192.pdf, TransactionalGroupID=902482</pre>
<input type="button" value="Invoke"/>	

Creating Transactional Email Groups:

Different kinds of transactional emails may be organized into Transactional Groups. A particular Transactional Group can then be assigned to the email message upon each call of the SendTransactionalEmail method. The Reporting interface will then show reporting data organized by Transactional Group.

Transactional Email Groups can be created and managed in your account under **Settings** → **Transactional Groups**.

Group ID	Group Name	Create Date	Action
27738	Account Expiration	12/2/2009 11:48:00 AM	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
25384	Default Transactional Group	9/30/2009 10:53:00 AM	
27739	Order Confirmations	12/2/2009 11:49:00 AM	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
27740	Shipping Status	12/2/2009 11:49:00 AM	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

In this account, three Transactional Groups have been created. All accounts come with a "Default Transactional Group" that cannot be modified or deleted. Next to each Transactional Group is its unique numeric identifier. This numeric identifier can be specified when calling the SendTransactionalEmail method to assign that particular email message to the TransactionalGroup. Specifically, within the Options input parameter, specify the parameter "TransactionalGroupID". For example, the Options input parameter may be:

"OpenTrack=True, ClickTrack=True, TransactionalGroupID=27739"

for a transactional email message that is an order confirmation for the sending organization.

If a TransactionalGroupID is not specified in the Options input parameter, the email message is grouped into the "Default Transactional Group".

Additional API Methods for Reporting:

Reporting data is accessible from the JangoSMTP web interface under **Reports** → **Transactional Emails**. Reporting data is also accessible through the following API methods:

Reports_Transactional_GetRecipients_DataSet
Reports_Transactional_GetRecipients_XML
Reports_Transactional_GetRecipients_String

Reports_Transactional_GetOpens_DataSet
Reports_Transactional_GetOpens_XML
Reports_Transactional_GetOpens_String

Reports_Transactional_GetClicks_DataSet
Reports_Transactional_GetClicks_XML
Reports_Transactional_GetClicks_String

Reports_Transactional_GetSingleEmailStats_Dataset
Reports_Transactional_GetSingleEmailStats_String
Reports_Transactional_GetSingleEmailStats_XML

Each method is available in three return types, a .Net DataSet, an XML document, and a String.

More information on these methods is available on the [API home page](#) and in the [API technical documentation](#).

Summary:

JangoSMTP's SendTransactionalEmail Web Service method is a powerful mechanism for sending single-recipient transactional emails. Email messages are signed DomainKeys, DKIM, tagged for open and click tracking, and are sent immediately through high-deliverability SMTP servers.