

# Configuring SMTP endpoints

- [Overview](#)
- [Configuration properties](#)
  - [mail.clients](#)
  - [mail.<clientId>.host](#)
  - [mail.<clientId>.port](#)
  - [mail.<clientId>.username](#)
  - [mail.<clientId>.password](#)
  - [mail.<clientId>.protocol](#)
  - [mail.<clientId>.subject](#)
  - [mail.<clientId>.to](#)
  - [mail.<clientId>.cc](#)
  - [mail.<clientId>.bcc](#)
  - [mail.<clientId>.groups](#)
  - [mail.<clientId>.converterRef](#)
  - [mail.<clientId>.converterClass](#)
- [SMTP properties example](#)
  - [SMTP endpoint for Gmail server](#)
  - [SMTP endpoint for Gmail server with TLS](#)
- [Using templates for emails](#)
  - [Sample template for sending a notification about FIX message](#)
  - [Sample template for sending a notification about FIX session state change](#)

## Overview

The JavaMail framework is integrated into FIXEdge Java in the form of SMTP destination endpoints and gives the ability to send emails to external SMTP servers.

SMTP is a destination endpoint only.

## Configuration properties

The list of main FEJ SMTP endpoint properties:

Property	Description	Required
<b>mail.clients</b>	A comma-separated list of FEJ SMTP client IDs	Y
<b>mail.&lt;clientId&gt;.host</b>	The SMTP server to connect to	Y
<b>mail.&lt;clientId&gt;.port</b>	The SMTP server port to connect to	Y
<b>mail.&lt;clientId&gt;.username</b>	The SMTP user name to connect with	Y
<b>mail.&lt;clientId&gt;.password</b>	The SMTP user's password. Optional if SMTP supports user access without a password.	N
<b>mail.&lt;clientId&gt;.protocol</b>	Protocol for connection to the SMTP server. 'smtp' supported.	N
<b>mail.&lt;clientId&gt;.subject</b>	Default email's subject	Y
<b>mail.&lt;clientId&gt;.to</b>	Default email's 'To' comma-separated list	Y
<b>mail.&lt;clientId&gt;.cc</b>	Default email's 'Cc' comma-separated list	N
<b>mail.&lt;clientId&gt;.bcc</b>	Default email's 'Bcc' comma-separated list	N
<b>mail.&lt;clientId&gt;.groups</b>	FEJ groups for routing to this SMTP endpoint	N
<b>mail.&lt;clientId&gt;.converterRef</b>	An instance of the <a href="#">email converter interface</a> , as a reference to the Spring bean name, to convert incoming MessageEvent to email fields.	N
<b>mail.&lt;clientId&gt;.converterClass</b>	Implementation class of the <a href="#">email converter interface</a> to create a converter instance on endpoint initialization.  By default, an implementation based on <a href="#">Apache Velocity templates</a> is used.	N

The list of the main JavaMail properties, set with the `mail.<clientId>.properties` prefix:

Property	Description
mail.smtp.from	Email address to use for the SMTP MAIL command. This sets the envelope return address.
mail.smtp.auth	If <i>true</i> , attempt to authenticate the user using the AUTH command.
mail.smtp.socketFactory.port	Specifies the port to connect to when using the specified socket factory.
mail.smtp.socketFactory.class	If set, specifies the name of a class that implements the <code>javax.net.SocketFactory</code> interface. This class will be used to create SMTP sockets.
mail.smtp.starttls.enable	If <i>true</i> , enables the use of the STARTTLS command (if supported by the server) to switch the connection to a TLS-protected connection before issuing any login commands. If the server does not support STARTTLS, the connection continues without the use of TLS; see the <a href="#">mail.smtp.starttls.required</a> property to fail if STARTTLS isn't supported.
mail.smtp.starttls.required	If <i>true</i> , requires the use of the STARTTLS command. If the server doesn't support the STARTTLS command, or the command fails, the connect method will fail.
template	Path to the Velocity template file for the default <a href="#">email converter interface</a> . See more at <a href="#">Template converter properties</a> .

The full list of properties could be found [here](#).

**NOTE:** The FIX session can be started or stopped only if scheduling is applied. Otherwise, the session will be inactive. Refer to the [Configuring Scheduler](#) section where the session start procedure is specified.

## SMTP properties example

### SMTP endpoint for Gmail server

#### SMTP properties example

```
# comma separated list of SMTP clients
mail.clients=smtpl

# SMTP server connection details
mail.smtpl.host=smtp.gmail.com
mail.smtpl.port=587
mail.smtpl.username=smtp-user1
mail.smtpl.password=pass1

# default Subject/To/CC/BCC values
mail.smtpl.subject=Test subject
mail.smtpl.to=<John_Doe@outlook.com, Jane_Doe@outlook.com, etc.>
mail.smtpl.cc=<cc list>
mail.smtpl.bcc=<bcc list>
mail.smtpl.protocol=smtp

# FEJ routing groups
mail.smtpl.groups=maill-group

# Optional. Converter class or Spring reference bean name
#mail.smtpl.converterClass=com.epam.fej.smtp.TestEmailConverter
#mail.smtpl.converterRef=testEmailConverter

# additional SMTP properties
mail.smtpl.properties.mail.smtp.auth=true
mail.smtpl.properties.mail.smtp.starttls.enable=true

# Velocity template for email content
mail.smtpl.properties.template=mail_message_template.vm
```

## SMTP endpoint for Gmail server with TLS

### SMTP properties to send emails from Gmail

```
mail.clients=smtp-gmail

mail.smtp-gmail.host=smtp.gmail.com
mail.smtp-gmail.port=587
mail.smtp-gmail.username=<user_name>
mail.smtp-gmail.password=<pass>

mail.smtp-gmail.subject=Test subject
mail.smtp-gmail.to=<John_Doe@outlook.com>
mail.smtp-gmail.cc=<cc>
mail.smtp-gmail.bcc=<bcc>
mail.smtp-gmail.protocol=smtp

mail.smtp-gmail.groups=gmail-group
#mail.smtp-gmail.converterClass=com.epam.fej.smtp.TestEmailConverter
#mail.smtp-gmail.converterRef=testEmailConverter

mail.smtp-gmail.properties.mail.smtp.auth=true
mail.smtp-gmail.properties.mail.smtp.starttls.enable=true
mail.smtp-gmail.properties.mail.smtp.host=smtp.gmail.com
mail.smtp-gmail.properties.mail.smtp.port=587
mail.smtp-gmail.properties.mail.smtp.socketFactory.port=587
mail.smtp-gmail.properties.mail.smtp.socketFactory.class=javax.net.ssl.SSLSocketFactory
#mail.smtp-gmail.properties.mail.debug=true

# Velocity template for email content
mail.smtp.properties.template=mail_message_template.vm
```

## Using templates for emails

If custom implementation for the [email converter interface](#) is not defined, the SMTP endpoint uses the default one, which is based on [Apache Velocity](#). Velocity is a Java template engine with simple but powerful syntax.

The converter puts the received `MessageEvent` object and all its header fields to the template context as shown in the table below:

Variable name	Variable type	Description
event	MessageEvent	Original <code>MessageEvent</code> for converting to email.
source	AppEvent	Predefined header field. The source <code>AppEvent</code> instance in case, if it was routed to SMTP endpoint.
email.template	String	Predefined header field. Custom template name. Can be set via the <code>MessageEvent#setHeader()</code> field.
email.subject	String	Predefined header field. Custom email subject. Can be set via the <code>MessageEvent#setHeader()</code> field.

The path to the template should be defined with the `template` property in the SMTP endpoint configuration. If the template is placed into the **conf** directory, the path can be relative:

### default Velocity template definition

```
mail.smtp-adaptor.properties.template=mail_message_template.vm
```

## Sample template for sending a notification about FIX message

#### default Velocity template definition

```
*** FIXEdge SMTP Notification System ***<br/>
<br/>
Notification Type: FIX<br/>
<br/>
FIX: <b>${event.getMessage().toPrintableString()}</b><br/>
```

## Sample template for sending a notification about FIX session state change

#### default Velocity template definition

```
*** FIXEdge SMTP Notification System ***<br/>
<br/>
FIX Session ID: ${source.getSessionId()}<br/>
New State: ${source.getSessionState()}<br/>
```