

Send an email

This function has been **renamed** with the [JWT 3.0](#) release.

Find the new documentation at:

[Send email](#)

On this page

- [Purpose](#)
- [Example: Notifying issue closing to request participants and Service Desk team](#)
- [Configuration Parameters](#)
- [Usage Example](#)
- [Related Features](#)

Purpose

Post-function **Send an email** allows to send an email with custom Subject and Body, setting recipients by means for **fields**, **custom fields**, **groups**, **project roles** or simply **fixed email addresses**.

You can send with your emails all or some of the **attachments** in the issue.

Example: Notifying issue closing to request participants and Service Desk team

In this example we define an email with a custom text that will be sent to issue **reporter**, users in field **Request Participants**, and to users in **Service Desk Team** project role. Users selected repeatedly are filtered, i.e., only one email is sent per recipient.

Project leader will receive a copy of the message, but in case he is also selected as direct recipient of the email (i.e., in **To**), then he will be automatically excluded from **Cc**.

If a file called **resolution_report.pdf** is attached to the issue, then it will be sent attached to the email.

We use parameter "**Conditional execution**", so that the email will only be sent if current issue's **Security Level** is different from "**Classified**". Note that **%{00103}** is field code for **Security level** virtual field.

From: Select the sender of the message.	<div><input checked="" type="radio"/> Default <input type="radio"/> Project Email Address <input type="radio"/> User in Field <input type="radio"/> User <input type="radio"/> Email Address in Field <input type="radio"/> Email Address</div> <div>Default address this JIRA server uses to send emails from.</div>												
To: Select the destination address of the message. Repeated addresses will be automatically filtered.	<table><tr><th>Users in Field</th><th>Emails in Field</th><th>Project Roles</th><th>Groups</th></tr><tr><td><div><div>✕ Reporter</div><div>✕ Request participants</div></div></td><td><div></div></td><td><div><div>✕ Service Desk Team</div></div></td><td><div></div></td></tr><tr><td colspan="4">Comma separated list of email addresses:<div>destinatary_1@example.com, destinatary_2@example.com, ...</div></td></tr></table>	Users in Field	Emails in Field	Project Roles	Groups	<div><div>✕ Reporter</div><div>✕ Request participants</div></div>	<div></div>	<div><div>✕ Service Desk Team</div></div>	<div></div>	Comma separated list of email addresses: <div>destinatary_1@example.com, destinatary_2@example.com, ...</div>			
Users in Field	Emails in Field	Project Roles	Groups										
<div><div>✕ Reporter</div><div>✕ Request participants</div></div>	<div></div>	<div><div>✕ Service Desk Team</div></div>	<div></div>										
Comma separated list of email addresses: <div>destinatary_1@example.com, destinatary_2@example.com, ...</div>													
Cc: Select the destination address of the copy message. Repeated addresses, and those already in To will be automatically filtered.	<table><tr><th>Users in Field</th><th>Emails in Field</th><th>Project Roles</th><th>Groups</th></tr><tr><td><div><div>✕ Project leader</div></div></td><td><div></div></td><td><div></div></td><td><div></div></td></tr><tr><td colspan="4">Comma separated list of email addresses:<div>destinatary_1@example.com, destinatary_2@example.com, ...</div></td></tr></table>	Users in Field	Emails in Field	Project Roles	Groups	<div><div>✕ Project leader</div></div>	<div></div>	<div></div>	<div></div>	Comma separated list of email addresses: <div>destinatary_1@example.com, destinatary_2@example.com, ...</div>			
Users in Field	Emails in Field	Project Roles	Groups										
<div><div>✕ Project leader</div></div>	<div></div>	<div></div>	<div></div>										
Comma separated list of email addresses: <div>destinatary_1@example.com, destinatary_2@example.com, ...</div>													
Sending mode: Select recipients visibility, and shared or personalized message.	<div><input type="radio"/> Common <input checked="" type="radio"/> Personalized</div> <div>A different email is sent to each recipient in field To. <i>Subject</i> and <i>Body</i> can be personalized using the following fields: '<i>Ephemeral string 4</i>' = recipient's username, and '<i>Ephemeral string 5</i>' = recipient's full name. Warning!: Beware of the fact that recipients in Cc will receive all the emails sent.</div>												

Email's Subject: Enter a title for your email.	<div>1 Issue {00015} has been closed as {00028}.</div> <div>Field code injector: [Line 1 / Col 44]<div>Start typing to get the list of availabe fields</div><div>Field codes with format {nnnnn} will be replaced with the corresponding values. Specific levels of Cascading Select fields can be referenced with {nnnnn.0} for parent level, and {nnnnn.1} for child level.<div>Syntax Specification Check Syntax</div></div></div>
String and math-time expressions can be inserted between 3 curly brackets, i.e., {{expression}} .	
Email's Body: Write the content of your email.	<div>Format of the message: <div><input type="radio"/> Plain Text <input checked="" type="radio"/> HTML</div><div>Syntax Specification Check Syntax</div></div> <div><div>1 <p>Dear {00067},</p> 2 3 <p>We send this email to inform you that issue {00015} has been closed as {00028} on {00057} by {00020}.</p> 4 5 <p>Attached to this email you will find the Resolution report.</p> 6 7 <p>Best regards,
 8 9 Your Service Desk</p></div><div>Field code injector: [Line 9 / Col 22]<div>Start typing to get the list of availabe fields</div><div>Field codes with format {nnnnn} will be replaced with the corresponding values. Specific levels of Cascading Select fields can be referenced with {nnnnn.0} for parent level, and {nnnnn.1} for child level.</div></div><div>Special values injector:<div>Recipient's user name</div><div>Recipient's user name and Recipient's full name can only be injected with personalized sending mode.</div></div></div>
String and math-time expressions can be inserted between 3 curly brackets, i.e., {{expression}} . Example 1: Number of subtasks: {{count(subtasks())}} Example 2: Tomorrow is: {{dateToString({00057} + 1 * {DAY}, LOCAL, SERVER_LANG)}} Example 3: Description from WIKI to HTML: {{wikiToHTML({00001})}} Example 4: Formatted summary: {{capitalizeWordsFully(trim({00000})}}	

Email's Body Preview:

Dear %00067,

We send this email to inform you that issue %00015, has been closed as %00028 on %00057 by %00020.

Attached to this email you will find the *Resolution report*.

Best regards,
Your Service Desk

Attachments:

Files attached to the issue can be sent as email attachments.

☐ No Attachments
☒ All files attached to the issue
☐ Only files attached in current transition

Size limit per file in KB:

4096

Attach files by extension:

jpg png gif txt doc

Files with input extensions will be attached.

Attach files by name:

report.doc[sales.xls]present

Files matching input **regular expression** will be attached.

Input a blank space separated list of field extensions. Leave "Attach files by extension" and "Attach files by name" parameters empty in order to attach all the files of the issue.

Input a **regular expression**. If you are not familiarized with regular expressions, you can simply use a list of file names separated by | character, like in the example.

Additional actions:

Optional actions that will be executed along with email sending.

☐ Add **To** recipients as **watchers** of current issue (only those who are Jira users).
☐ Add **Cc** recipients as **watchers** of current issue (only those who are Jira users).
☐ Save email information into *Ephemeral string* fields (*ES1...ES5*), to be used in subsequent post-functions:

ES1=*From*, ES2=*To*, ES3=*Cc*, ES4=*Mail subject*, ES5=*Mail body (text format)*.

☐ Exclude prefix from email's subject.

Conditional execution:

Optional boolean expression that should be satisfied in order to actually execute the post-function.
([Syntax specification](#))

1 %00103 != "Classified"

Leave the field empty for executing the post-function unconditionally.

Collection of examples

[Line 1 / Col 25]

Logical connectives:

and, or and not. Alternatively you can also use &, | and !.

Comparison operators:

=, !=, >, >=, < and <=. Operators in, not in, any in, none in, ~ and != can be used with *strings*, *multi-valued fields* and *lists*.

Logical literals:

true and false. Literal null is used with = and != to check whether a field is initialized, e.g. {00012} != null checks whether *Due date* is initialized.

String field code injector:

Security level - (Security level) - [00103]

Numeric/Date field code injector:

Start typing to get the list of availabe fields

Check syntax

Once configured, the transition will look like this:

IN PROGRESS

Close Issue

CLOSED

Screen: [Resolve Issue Screen](#)

Triggers 0

Conditions 0

Validators 0

Post Functions 6

The following will be processed after the transition occurs

Add post function

- Send an email with the following parameters:

From: **default sender** for current JIRA instance

To: users in fields **Reporter** and **Request participants** | users in project roles **Service Desk Team**

Cc: users in fields **Project leader**

Sending mode: **Personalized**

Email's subject:

Issue %{Issue key} has been closed as %{Resolution}.

Email's body:

```
<p>Dear %{Ephemeral string 5},</p>

<p>We send this email to inform you that issue <a href='%{JIRA base URL}/browse/%{Issue key}' target='_blank'>%{Issue key}</a>
has been closed as <em>%{Resolution}</em> on <em>%{Current date and time}</em> by <em>%{Current user}</em>.</p>

<p>Attached to this emails you will find the <em>Resolution Report</em>.</p>

<p>Best regards,<br>
```

Body format: **HTML**

Attachments: all issue attachments, with less than **5000 KB**, and with file names matching **resolution_report.doc** regular expression.

Post-function will only be executed if the following boolean expression is satisfied: **%{Security level} != "Classified"**

Configuration Parameters

From

Specifies the sender of the message. Four options are available for selecting the sender:

- **Default:** the default sender used by Jira when doing notifications by email.
- **Project Email Address:** it's possible to configure a [project's email address](#), which is the email address that notifications are sent from.
- **User in Field:** user selected in a **User Picker** field, or user whose **user name** is in a text field.
- **Email Address in Field:** email address appearing in a text field. In case the field contains more than one email address, the first one found will be selected.
- **Email address:** a literal email address entered.

To

Specifies the direct recipients of the message. Five options are available:

- **Users in Field:** users selected in User Picker or Multi-User Picker fields, or users whose **user name** is contained in text fields. Comma, semicolon, or space characters can be user to separate the user names. Don't confuse **user name** with user's **full name**.

- **Emails in Field:** email addresses contained in text fields. All the email addresses found will be used as email's recipients. The email addresses may appear among fragments other text fragments, e.g., if you select field Description, any email address appearing in the description of the issue will be used as recipient of the email.
- **Project Roles:** all the users in selected project roles will receive the email.
- **Groups:** all the users in selected groups will receive the email.
- **Comma separated list of email addresses:** all the email addresses input will receive the email.

Even though an email address may appear more than once in the selection, only one email will be sent per email address.

Cc

Specifies secondary recipients of email that receive a copy of the email. Same five options as in parameter To are available for selecting this kind of recipients.

If an email address has been previously selected as direct recipient (i.e., in To), it will be excluded in runtime from Cc, i.e., a same email address will not receive the same message as To and Cc at the same time.

Sending mode

To options are available:

- **Common:** only one email will be sent to all the selected recipients in To and Cc parameters. All the users in To will be visible to all the recipients of the email.
- **Personalized:** each email address in To will receive a unique message, i.e., only a recipient will appear in field To in the email. In this mode **Ephemeral string 4** contains the recipient's **user name**, and **Ephemeral string 5** contains the recipient's **full user name**. This way you will be able to mention each individual recipient in the Subject and the Body of your custom messages. These two fields will not be filled up for recipients selected using **Emails in Field**, and **Comma separated list of email addresses** options, since no JIRA user is associated in those cases.

Email's subject

The title of the email message. You can inject field codes that will be replaced with their corresponding field values at runtime.

Email's body

The main text of the email message. You can inject field codes that will be replaced with their corresponding field values at runtime.

Two **formats** are available:

- **Plain text**
- **HTML:** you can use html tags to set the format and layout of the message.

Field codes are supported by both formats of email's body.

Attachments

Files attached to the issue can be sent as email attachments. There are three options available:

- **No Attachments:** no files are attached to the email.
- **All files attached to the issue:** all the issue attachments will be attached to the email, except those filtered by next parameters.
- **Only files attached in current transition:** files attached to the issue using the screen associated to current transition will be also attached to the message, except those filtered by next parameters.

We can filter the files to be attached using the following parameters:

- **Size limit per file in KB:** files exceeding the size limit will not be attached to the message. Leave this parameter empty if you don't want to set size limit.
- **Attach files by extension:** enter a list of extensions separated by spaces. Issue attachments with selected extensions will also be attached to the email.
- **Attach files by name:** enter a **regular expression**, so that issue attachments whose names match the entered regular expression will be attached to the email. If you are not familiarized with regular expression, you can simply use a list of file names separated by | character.

Parameters "**Attach files by extension**" and "**Attach files by name**" work as a **union** set operation, i.e., issue attachments matching **any** of the two parameters, will be attached to the email. If you leave both parameters empty, all the attachments will be attached to the email, i.e., no filtering is applied.

Additional actions

Optional actions that will be executed along with sending emails. Available actions are:

- Add **To** recipients as **watchers** of current issue.
- Add **Cc** recipients as **watchers** of current issue.

Only those recipients who are Jira users will be effectively added as watchers.

- **Save email information** into **Ephemeral String** fields (ES1...ES5), to be used in subsequent post-functions:
 - Ephemeral string 1: Contains the **sender's full name** (only in the case of Jira users) and its **email address**, i.e. the **From** field.
 - Ephemeral string 2: Contains the **user's full names** (only in the case of Jira users) and the **email addresses** of the **To** recipients.
 - Ephemeral string 3: Contains the **user's full names** (only in the case of Jira users) and the **email addresses** of the **Cc** recipients.
 - Ephemeral string 4: Contains the **Subject** of the email.
 - Ephemeral string 5: Contains the **Body** of the email in text format. If the email is composed in HTML, it's transformed into text removing tags, and inserting line returns where it's convenient in order to keep a readable aspect.

When using **Personalized** sending mode, fields Ephemeral string 4 and Ephemeral string 5 will contain the email's Subject and Body of a random email of the several ones possibly sent.

Conditional execution

If you want to send the email depending on a condition, you can enter a boolean expression in this parameter, and only when the boolean expression is satisfied the email is sent, otherwise nothing happens. You can make your boolean expression depend on the values of one or more fields, issue links, sub-tasks, etc. Use the syntax defined by the [Expression Parser](#).

i

Insert before event firing post-function

This post-function should be executed **before** the one that fires the event at the end of a transition, since the event triggers email delivery.

Usage Example

Related Features