

EmailSupport

```
mail_import_secret = u"do_not_choose_a_too_short_secret_or_it_wont_work"
```

The target URL

```
mail_import_url = u"http://localhost:81/?action=xmlrpc2" }}}
```

After you have configured the settings explained above, you should be able sending yourself mails. The actual usage is explained below.

From :: We check the From: header if it is from a valid wiki user. Of course this is no real security, but maybe better than nothing.

:: When we create a mail overview table, we use the email address part of From: to try to lookup a wiki user account with that email address. In case we find some, we will link to his homepage from the From: column of the table. If we don't find a matching user account, we just list the realname part in the From: column.

To / Cc :: To: is (as usual) made of 2 parts, like `TEMPLATE <wiki@example.org>`. We use the "realname" part to make up the target page name template (can contain special chars, blanks etc.) and the email address part to get it to the right destination. The simplest form of a target page name template is just a pagename like WikiSandBox.

:: See the example for the 2 different modes of operation.

:: When we create a mail overview table, we process To: and Cc: for the To: column in the same way we do it for the From: column (see above). We do not include the address of the wiki server (and if we only have the wiki server email, we just display "-").

Subject :: The page name template can also be extracted (and also removed) from the first occurrence of `mail_import_pagename_regex` within the subject.

:: Examples: `{{{ Subj: [[TargetPageName]] will work if you put it either at the beginning or the end of the subject. Subject: bla bla bla [[PageName/$date]] }}}`

:: The text of mails with such subject will be added to the pages `{{{ TargetPageName or PageName/2006-07-20T14:48:27 }}}`

:: The full subject will also be used to build the change comment, prefixed with "`Mail:`"; variables are not replaced there.

Mail body :: The new page content is made from old page content (if any) plus the content of the email appended at the end. Page content will never get overwritten. Mails that are sent as `alternative` are shown with their HTML body on the page which is processed by a HTML parser.

:: E-Mail attachments will get put into page attachments under the page where the content will be put. Attachments will never get overwritten. The attachments will be linked from the content page (e.g. from a list of attachments at the end).

Generated content :: If requested by a leading `+`-sign (e.g. `+ PageName/`), Moin adds a table item to the parent page listing details about the e-mail.

Pagename generation :: The placeholders `$from` (containing the name of the sender), `$date` (the date) and `$subject` (subject line) are supported.

:: The pagename is generated like explained below:

1. Check if one of the mail target addresses (to, cc, ...) is in `mail_import_wiki_addrs`
2. Process TO/SUBJECT in order as given by `mail_import_pagename_search` until we have found a pagename or have to give up:

- Try to get the pagename template from the target addrs, or
- Try to extract the pagename from the subject (explained above).

1. If we didn't find a pagename, use `mail_import_subpage_template` as last resort.
2. Apply `mail_import_pagename_envelope` to (optionally) extend the pagename.
3. If the pagename ends with a slash, append the configuration setting `email_subpage_template` (see above) to the pagename.
4. Expand the pagename template to get the actual pagename.

Version #1

Erstellt: 2025-10-27 22:19:55 UTC von ESC-IT Migration Bot

Zuletzt aktualisiert: 2025-10-27 22:19:56 UTC von ESC-IT Migration Bot