

JIRA Feedback Plugin documentation

- 1. Installation
- 2. Configuration
- 3. Setting up language in form
- 4. Use Cases
- 5. Using Mail Handler

1. Installation

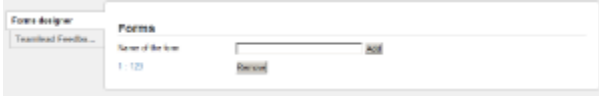


You can download plugin from [this page](#).
Install with Universal Plugin Manager.

1. The first step is license key request. In the administrative part of JIRA Plugins menu the *Teamlead Feedback -> Settings* _item appears from the drop down list. You have to insert key in the _license key window and confirm.



2. After getting the key go to the same menu and add new / edit an existing form.



2. Configuration



User must have at least administrator permission and maximum system administrator to access the plugin configuration.

* Name of the form	Filled arbitrarily.
* Project	In this project the issues will be created.
* Issue type	The issues with this type will be created in a project referred in Project field.
* User to create users	The issues will be created on behalf of this user. ⚠ Please note, this user must have create users permission.
* User for search	JIRA will search issues (the fields that will be search indicated below) on behalf of this user .
* Jira URL	The form will refer to JIRA according to this URL form to create issue.
Groups	Created user will be automatically added to this group.
Administrator email	The user can contact the specified address, if it the forms works incorrect.
The output matches	The form will display a list of matches on the current screen.
Fields for search	Summary, Description, Comment, Environment.
Field type for component	• none; • select; • checkbox; Issue will be created with the selected values of the components. The values are loaded into the form at the time of generation form code. If after the publication of the form you have created a new component, you must update the form to upload another meaning.

Field type for affect version	• none; • select; • checkbox; Issue will be created with the selected values of the versions. The values are loaded into the form when the form code is generated. If you have created a new version after the form is generated, you must update the form to upload new values.
Allow file upload	Ability to attach a file to the issue (Ability to add multiple files will be added in the next verisons).
Send a notification after user creation	Does not appear on the screen form. Notification will be sent with the authorization data to the new user.
Access level to the issue after creating	Does not appear on the screen form. Enter the required level, if your JIRA has issue security levels.
Field name on the form	The name of the custom field displayed on the screen.
Field type	• string • simple selectlist • multi-check • cascading-select
Custom field	The name of the custom that will be displayed.
Message about the successful registration of the issue	You can use %ISSUE_LINK% variable in your text about successful issue registration.
Form code	Generated foem code for inserting.

* required fields.

3. Setting up language in form

To select which language to use in the form you need to add this tags to the code of the form:

```
<div style='display:none;'><input type='text' id='feedback-language' name='feedback-language' class='feedback-language' placeholder=' (en/ru)' value='en'/></div>
```

If you need English - then leave `value='en'`. If you need Russian - change value to `'ru'`.

4. Use Cases

1. Creating issue from a site page

Select the component

Select a version

Summary

You can specify the details ...

E-mail

Choose File No file chosen

Ask

2. Reviewing of your issues from a site page

Hide issues history

my@my.ru

Issue key

Issue summary

Search

Open issues

Resolved issues

Closed issues

All issues

Key	Issue type	Summary	Status
TEST-6	Bug	Create interface	Open
TEST-5	Bug	This is a bug	Open
TEST-4	Bug	I would like to improvement this	Reopened
TEST-3	Bug	Test of code	Resolved
TEST-2	Bug	Mama mia!	Open

3. Configurator

Forms / Form 2

* Name of the form:

* Project:

* Issue type:

* User to create issue:
Start typing a username to match

* User for search:
Start typing a username to match

* Use GPT:

Groups: [Add group](#)

Administrator email:

The output matches: ☐

Fields for search: ☐ Summary
☐ Description
☐ Comment
☐ Environment

Field type for component:

Field type for affect version:

☐ Add file upload

☐ Send a notification after user creation

Access level after creating issue:

Field name on the form:

Field type:

Custom field:

[Add field](#) [Remove all fields](#)

4. Code of creating form for insert into a site page:

```

<STYLE TYPE='text/css'>BODY, input, select, table {
font-size: 12px;font-family: Arial,Helvetica,FreeSans,'sans - serif';}
</STYLE>
<div id='feedback-main-div' style='font-size: 12px; font-family: Arial,Helvetica,FreeSans,sans-serif'>
<div id='feedback-error'></div>
<form enctype='multipart/form-data' id='feedback-form' name='feedback-form' target='feedback-response'
method='post' action='http://localhost:8080/plugins/servlet/feedback/create'>
<input type='hidden' name='form-index' id='form-index' value='1'/>
<div><input type='text' id='feedback-summary' name='feedback-summary' size='60' placeholder='Summary' value=''
/></div>
<div><TEXTAREA id='feedback-body' name='feedback-body' COLS='62' ROWS='5' placeholder='You can specify the
details ...'></TEXTAREA></div>
<div><select id='feedback-component' name='feedback-component'>
<option value='' selected >Select the component</option>
</select>
</div>
<input type='text' id='feedback-email' name='feedback-email' placeholder='E-mail' size='38' value='' />
<div><input type='file' id='feedback-attachment' name='feedback-attachment' size='38' /></div>
<input type='button' value=' Ask ' onclick='addFeedback()' />
</form></div>
<iframe id='feedback-response' name='feedback-response' frameborder='no' style='display: none' width='450'
height='200'></iframe><div id='feedback-new-issue-div'><input type='button' id='feedback-new-issue-button'
value='Create more issue'></div><script type='text/javascript'>
jQuery(window).load(function()
{
    jQuery('#feedback-main-div').show();
    jQuery('#feedback-new-issue-div').hide();
    jQuery('#feedback-response').hide();
    jQuery('#feedback-new-issue-button').click(function ()
    {
        jQuery('#feedback-form')[0].reset();
        jQuery('#feedback-main-div').show();
        jQuery('#feedback-new-issue-div').hide();
        jQuery('#feedback-response').hide();
    });
});

function addFeedback() {
    var summary = jQuery('#feedback-summary').val();
    var email = jQuery('#feedback-email').val();
    if (summary.trim() == '' || email.trim() == '')
    {
        var message = '';
        if (summary.trim() == '')
        {
            message = message + '<div>Enter the summary</div>';
        }
        if (email.trim() == '')
        {
            message = message + '<div>Enter E-mail</div>';
        }
        jQuery('#feedback-error').html(message);
        return false;
    }
    jQuery('#feedback-error').html('');
    jQuery('#feedback-main-div').hide();
    jQuery('#feedback-new-issue-div').show();
    jQuery('#feedback-response').show();
    jQuery('#feedback-form').submit();
    return false;
}
</script>

```

5. Using Mail Handler

After the plugin is installed new Teamlead Jira Email Handler will appear. It allows you:

- to specify the group which newly created users will be added to;
- to select what group new users will be added to;
- to create issues from emails with empty subject.

To specify the group which newly created users will be added to you need to set up parameter:

userGroup=<usergroup_name>

You can see examples of other mail handler parameters [on the Atlassian website](#).

- [JIRA Core](#)
- [Confluence](#)
- [Crucible](#)
- [Bamboo](#)
- [Stash](#)

-
-
-
-
-

- [CRM](#)
- [HelpDesk](#)
- [Timesheet](#)
- [Project Management](#)

Teamlead

-
-
-
- [Work at Teamlead](#)
-