

How to sync Table Grid Editor data

Last Modified on 01/15/2026 3:42 pm EST

The Table Grid Editor is a JIRA add-on that allows you to store tabular data in the context of an issue.

It is available in [Atlassian Marketplace](#).

Type:

Priority:

Affects Version/s:

Component/s:

Labels:

grid checklist:

GRID Checklist

P3 - Major

None

None

None

Status:

Resolution:

Fix Version/s:

Open

Unresolved

None

	Summary	Responsible	Status
1	Unit test added/adapted	Kevin Spacey (team lead)	Open
2	Knowledge base article updated	Robert Duvall (Customer)	Open
3	Bug Impact Reviewed	Guy Pearce (IT)	Closed
4	Approved by the Customer		Open
5	Tests added to cover the improvment	unassigned	Closed
6	Impact on UI reviewed	Brad Pitt	Open
7	Documentation Adapted		Open
8	Support is aware of improvement	Testuser - developer	Open

Page 1 of 1

View 1 - 8 of 8

It is possible to synchronize the data contained in such grids between the two instances by using [the Table Grid API](#) in the Incoming sync.

The following script example shows how this can be done.

Incoming sync

```

1 import com.atlassian.crowd.embedded.api.User
2 import com.atlassian.jira.component.ComponentAccessor
3 import com.atlassian.jira.issue.CustomFieldManager
4 import com.atlassian.jira.issue.fields.CustomField
5 import com.atlassian.jira.security.JiraAuthenticationContext
6 import com.atlassian.plugin.PluginAccessor
7 import com.atlassian.jira.user.ApplicationUser
8 import com.atlassian.jira.user.util.UserManager
9
10
11
12
13 // get TGE custom field
14 // replace 'mytable' below with the name of the table grid
15//
16 CustomFieldManager customFieldManager = ComponentAccessor.getOSGiComponentInstanceOfType(CustomFieldManager.class);
17 CustomField tgeCustomField = customFieldManager.getCustomFieldObjectsByName("mytable").get(0);
18 Long tgeCustomFieldId = tgeCustomField.getIdAsLong();
19
20 // get exalate user - replace 'exalate' below with the user name of your proxy user
21 UserManager userManager = ComponentAccessor.userManager
22 User exalateUser = userManager.getUserByKey("exalate")?.directoryUser
23
24 // Get access to the table grid api
25 PluginAccessor pluginAccessor = ComponentAccessor.getPluginAccessor();
26 Class dataManagerClass = pluginAccessor.getClassLoader().findClass("com.idalko.jira.plugins.igrid.api.data.TGEGridTableDataManager");
27 def tgeGridDataManager = ComponentAccessor.getOSGiComponentInstanceOfType(dataManagerClass);
28
29
30
31 // Provide the data which needs to be added
32
33
34 Map<String, Object> row = new HashMap<String, Object>();
35 row.put("isummary", "Some summary");
36 row.put("iassignee", "admin");
37 row.put("istatus", "Done");
38 row.put("idue", new Date().getTime());
39
40
41 try {
42 List<Long> rowIds = tgeGridDataManager.addRow(Long.valueOf(issue.id), tgeCustomFieldId, Arrays.asList(row), exalateUser);
43 } catch (Exception e) {
44 // log an error if things go wrong, but continue with the synchronisation.
45 log.error("** Failed to add data due to " + e)
46 }
47
48
49
50 // do the normal sync stuff.
51
52 issue.summary = replica.summary
53 issue.description = replica.description
54 issue.comments = commentHelper.mergeComments(issue, replica)
55 issue.attachments = attachmentHelper.mergeAttachments(issue, replica)

```

[Pricing and Licensing](#)

Resources

[Subscribe for a weekly Exalate hack](#)

[Academy](#)

[Blog](#)

[YouTube Channel](#)

[Ebooks](#)

[Still need help?](#)

Still need help?

[Join our Community](#) 

[Visit our Service Desk](#) 

[Find a Partner](#) 