

Create Jira issue with webservice connection with files and remote links

Yes, you can create Jira issue with WebService IFTTT action (usual IFTTT to create Jira issues will not work for anonymous users as it does uses the Application link to connect to your Jira and that will not be available for anonymous users, while web service connection is available to anyone, unless you tell otherwise (setting up connection restrictions))

Here is the demo (creates the issue and stores the Jira key back)

Visual view in the editor

The screenshot displays the IFTTT editor interface for a 'ConfForms Form Definition' trigger. The trigger is titled 'formName = jj | registrationFormTitle = Create JIRA issue with ConfForms Webservice co...'. Below the trigger, there are several 'ConfForms Form Field Definition' actions for fields: 'summary', 'desc1', 'desc2', 'jk', and 'fi'. Additionally, there is a 'ConfForms Rules for Field Definition' action with the condition 'id:[empty] | ac...'. The main action is 'ConfForms IFTTT Integration Rules' with the event 'onCreated', action 'WebService Request', title '/rest/api/2/issue/', and who = PO... The JSON payload for the WebService Request is as follows:

```
{
  "fields": {
    "project": {
      "key": "JTEST"
    },
    "summary": "[entry.summary.escapeJSON]",
    "assignee": {"name": "admin"},
    "description": "[entry.desc1.escapeJSON] \n [entry.desc2.escapeJSON]",
    "issuetype": {
      "name": "Bug"
    }
  }
}
```


ConfiForms IFTTT Integration Rules | event = onCreated | action = Create ConfiForms Entry | title = entryId=[entry.id]&jk=[i...


ConfiForms IFTTT Integration Rules | event = onCreated | action = Webservice Request | title = /rest/api/latest/issue/entry...


ConfiForms Form Field | fieldName = fi


ConfiForms IFTTT Integration Rules | event = onCreated | action = Webservice Request | title = /rest/api/latest/issue/entry...

No Format

```

{
  "object": {
    "url": "https://wiki.vertuna.com/pages/viewpage.action?pageId=19104427",
    "title": "[entry._page.title.escapeJSON]"
  }
}

```

Source code for the solution

[Create Jira issue with webservice connection with files and remote links - storage format](#)

note that you need to configure webservice connection to something like this

WS Connection Configuration

Name: ^{*}

The name for your configuration, will be visible to users

WS URL

Enter with protocol (http:// or https://). This URL will be combined together with the service URL you setup. For WS-backed fields in ConfiForms you service must accept GET requests and return JSON as a result

Username

Leave empty, if the connection requires **no authentication**

Password

Restrictions

Comma separated user groups and/or users who should have access to this configuration. Blank, means no restrictions - everyone can use it

Save

[Close](#)

The table is created like this

```
<ac:structured-macro ac:macro-id="d309c641-56ef-46cb-b01f-19590d41605e" ac:name="coniform-table" ac:schema-version="1">
  <ac:parameter ac:name="formName">ji</ac:parameter>
  <ac:rich-text-body>
    <p>
      <ac:structured-macro ac:macro-id="bb4766d6-cbfe-40c6-ace4-f3dcd21dcc99" ac:name="coniform-field" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">jk.asJSON.key</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="18b7e9fe-30c3-4a04-886e-6a47446af441" ac:name="coniform-field" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">created</ac:parameter>
      </ac:structured-macro>
    </p>
  </ac:rich-text-body>
</ac:structured-macro>
```

Using asJSON function as a bridge ([Virtual functions](#)) and accessing the issue key (key property)

```
jk.asJSON.key
```

- [Create Jira issue with webservice connection with files and remote links - storage format](#)