

JT Formulator

What is JT Formulator?

This Joomla plugin allows you to embed forms on your Joomla website. The forms can be easily integrated anywhere on your Joomla website (in posts and modules).

How does JT Formulator work?

The forms are created in XML files so that you can use one form very quickly and easily on multiple Joomla installations.

The plugin supports the CSS frameworks Bootstrap 2 to 3 and UiKit 2 - 3.

Who is JT Formulator suitable for?

The plugin is suitable for beginners and professionals who know how to build a simple XML form.

Usage

Here's what you can use:

XML form templates from the plugin

Copy and adjust XML form templates from the plugin in a template override

Create your own XML forms in a template override

Advantages:

The plugin is based on Joomla standards (jForm fields)

It is easy to use

You can customize the subject and the recipient in the plugin call code and thus act very flexible

The plugin comes with a nice file upload field

The plugin does not provide a user interface - you can assemble or edit your forms in one XML file!

The XML forms you create can then be used multiple times on the website.

The plugin supports various CSS frameworks (Bootstrap 2, Bootstrap 3, UiKit 2, UiKit 3)

Getting started

To use the extension you have to download it from

https://github.com/JoomTools/plg_content_jtf/releases and install it via the Joomla Extension Manager. Then go to Extensions " Plugins and activate the JT Formulator Plugin.

As mentioned in the beginning you have .xml files which are responsible for the structure of your form. These forms can be displayed anywhere on your Joomla website (posts, modules,...) with the help of the plugin call code {jtf} - In the plugin call code you can pass individual values with parameters.

Prefabricated forms

Standard form (default)

The standard form has the input fields:

Name
E-mail
Subject
Message
Data protection consent

The form XML is stored under plugins/content/jtf/tmpl/default/.

To use the default form you can simply use the plugin call code {jtf} anywhere on your website where plugins can be executed.

To change the default form, you can copy and paste the required framework specific XML file from plugins/content/jtf/tmpl/default/ into your template to templates/deintemplatename/html/plg_content_jtf/default/.

To use the standard form as a basis for another form, you can also copy the file into your template under templates/deintemplatename/html/plg_content_jtf/deinformularname/ and adapt it.

Extended form (extended)

The extended form is a small demo form. It shows some additional fields, which are often used for a contact request.

This form is stored under plugins/content/jtf/tmpl/extended/.

To use the extended form, you can simply use the plugin call code {jtf theme=extended} anywhere on your website where plugins can be executed.

To change the extended form, you can copy and paste the required framework specific XML file from plugins/content/jtf/tmpl/extended/ into your template at templates/deintemplatename/html/plg_content_jtf/extended/.

To use the extended form as a basis for another form, you can also copy and adapt the file in your template under `templates/deintemplatename/html/plg_content_jtf/deinformulardname/`.

Pattern Form (pattern)

The sample form is another demo form that shows different JForm field types with different settings and a bit of XML magic. You will most likely not use it, but you can have a look at what is possible with JT Formulator.

This form is stored under `plugins/content/jtf/tmpl/muster/`.

To use the sample form you can simply use the plugin call code `{jtf theme=pattern}` anywhere on your website where plugins can be executed.

To change the pattern form, you can copy and paste the required framework specific XML file from `plugins/content/jtf/tmpl/muster/` into your template at `templates/deintemplatename/html/plg_content_jtf/muster/`.

To use the sample form as a basis for another form, you can also copy the file into your template under `templates/deintemplatename/html/plg_content_jtf/deinformulardname/` and adapt it.

Plugin Settings

CSS frameworks

The JT Formulator supports various CSS frameworks like Bootstrap 2, Bootstrap 3, UIKit 2 and UIKit 3 - to activate the framework you can either run it individually `{jtf framework=bs2}` in the plugin call or set it globally in the plugin settings. The value in the plugin call overrides the global setting.

Bootstrap 2 = `{jtf framework=bs2}`

Bootstrap 3 = `{jtf framework=bs3}`

UIKit 2 = `{jtf framework=uikit2}`

UIKit 3 = `{jtf framework=uikit3}`

Subject

The subject of the mail that is delivered is defined in the XML by the field named subject, or with the parameter `subject=your subject` in the plugin call.

Example: `{jtf subject=your subject}`

If the field is not filled in and is not queried in the form, your form will be sent with a default subject that uses the global page name in the language key JTF_EMAIL_SUBJECT.

Recipient

The recipient of the mail that is delivered is defined by the field with the parameter mailto=mail#empfaenger.de in the plugin call.

Example: {jtf mailto=head of department#empfaenger.de}

You can also name any field in the parameters from which you want to drag the recipient. For example mailto=fieldname. The value in the field must be an e-mail address and must not be empty.

CC and BCC recipient

You can also send the mails to bcc / cc recipients. To achieve this you define cc=mail#anotherrecipient.com or for bcc bcc=mail#blindrecipient.com in the plugin call, you can also name any field in the parameters from which you want to pull the other recipient. So cc=fieldname or bcc=fieldname

Example: {jtf cc=chef#empfaenger.de}

Tip: For the recipient parameters (mailto, cc, bcc) the following applies: several recipients can be specified, separated by semicolons. A combination of e-mail address and field is also possible. For example cc=cc#example.org;field name. This way you can send a form to multiple recipients.

Note: Remember to replace the @ sign in the plugin call with a hash #. Otherwise the plugin call may be destroyed by the email cloaking plugin.

Example for the transfer of field values into the mailto, cc and bcc parameter:

Field in the XML file:

```
<field name="email_list"
      type="list"
      icon="icon-list"
      description="JTF_LIST_DESC"
      label="JTF_LIST_LABEL"
      default=""
      validate="options"
```

```

        required="true"
    >

        <option value="">JTF_OPTION_NONE</option>
        <option value="empaenger1@example.org">Department 1</option>
        <option value="empaenger2@example.org">Department 2</option>

    </field>

```

Call in the plugin:

```
{jtf mailto=email_list}
```

Create your own forms

Structure

You can create as many forms as you like. You just have to insert the XML files into your template in the following structure:

```
templates/deintemplate/html/plg_content_jtf/deinformulardname/fields.xml
```

Joomla Core Fields

The JT Formulador is based on jForm standards with extended capabilities. In fact you can use any field that is possible in Joomla. Which fields Joomla offers can be found at: https://docs.joomla.org/Form_field

Extended fields of JT Formulador

The JT Formulador comes with a few more custom field types that you can use.

List of JT Formulador JForm fields here
postcode

Advanced XML Parameters / XML Magic

Joomla Core Magic:

hiddenLabel - hides the label from a field

showon: - shows the field in dependency to another field You can read how to use this at https://docs.joomla.org/Form_field#Showon.

Additional JT Formulador spells (attributes in XML field)

gridgroup

sets a class in the DIV container that encloses the label and the input field

gridlabel

sets a class in the DIV container, which only encloses the label

gridfield

sets a class in the DIV container, which only encloses the input field

icon

Icon class as given by the CSS framework, you enter uk-icon-envelope for UI-Kit for example.

optionclass/optionlabelclass

inserts a CSS class into each radio, checkbox and listbox radio radio radio radio option field

buttonclass

for field types with a button, e.g. calendar, a class can be added to the button

buttonicon

for field types with a button, e.g. calendar, you can change the icon in the button by specifying the icon class here

Display your own form in the frontend

to display your own form in the frontend call {jtf theme=deinformulardname} in your Joomla post or module.

Overrides in the plugin call code

Always remember that you can override some of the XML settings when you call the form.

To do this, write the parameters, separated by upstrokes (also called pipes |) in your plugin call.

JT Formulator

Was ist JT Formulator?

Dieses Joomla Plugin ermöglicht es dir Formulare auf deiner Joomla Webseite einzubinden.

Die Formulare können sehr einfach überall auf deiner Joomla Webseite (In Beiträgen und Modulen) eingebunden werden.

Wie funktioniert JT Formulator?

Die Formulare werden in XML Dateien angelegt so dass du ein Formular sehr schnell und einfach auf mehreren Joomla Installationen nutzen kannst.

Das Plugin unterstützt die CSS Frameworks Bootstrap 2 bis 3 und UIKit 2 - 3.

Für wen ist JT Formulator geeignet?

Das Plugin ist für Programmieranfänger und Profis geeignet, die wissen wie ein einfaches XML-Formular aufgebaut wird.

Nutzung

Du kannst folgendes nutzen:

- XML Formular-Vorlagen aus dem Plugin
- XML Formular-Vorlagen aus dem Plugin in einem Template Override kopieren und anpassen
- Eigene XML Formulare in einem Template Override erstellen

Vorteile:

- Das Plugin basiert auf Joomla Standards (jForm Felder)
- Es ist leicht zu nutzen
- Man kann den Betreff und den Empfänger im Plugin Aufruf Code anpassen und somit sehr flexibel agieren
- Das Plugin kommt mit einem schönen File Upload Feld
- Das Plugin bietet keine Benutzeroberfläche - du kannst deine Formulare in einer XML Datei zusammenstellen oder bearbeiten!
- Die von dir angelegten XML-Formulare kannst du dann einfach auf der Webseite mehrfach nutzen.
- Das Plugin unterstützt von Haus aus verschiedene CSS Frameworks (Bootstrap 2, Bootstrap 3, UIKit 2, UIKit 3)

Erste Schritte

Um die Erweiterung zu verwenden musst du diese von

https://github.com/JoomTools/plg_content_jtf/releases herunterladen und über den **Joomla Erweiterungs Manager** installieren. Danach gehst du in **Erweiterungen » Plugins** und aktivierst das JT Formulator Plugin.

Wie eingangs erwähnt hast du .xml Dateien, die für den Aufbau deines Formulars zuständig sind. Diese Formulare können mit Hilfe vom Plugin Aufruf Code `{jtf}` überall auf deiner Joomla Webseite dargestellt werden (Beiträge, Module,...) - Im Plugin Aufruf Code kann man mittels Parametern noch individuelle Werte übergeben.

Vorgefertigte Formulare

Standard-Formular (default)

Das Standard-Formular hat die Eingabefelder:

- Name
- E-Mail
- Betreff
- Nachricht
- Datenschutz-Zustimmung

Die Formular XML ist unter **plugins/content/jtf/tmpl/default/** gespeichert.

Um das Standard-Formular zu nutzen kannst du einfach den Plugin Aufruf Code **{jtf}** überall auf deiner Webseite einsetzen wo Plugins ausgeführt werden können.

- Um das Standard-Formular zu ändern, kannst du die benötigte framework spezifische XML-Datei aus **plugins/content/jtf/tmpl/default/** in dein Template nach **templates/deintemplatename/html/plg_content_jtf/default/** kopieren und anpassen.
- Um das Standard-Formular als Basis für ein anderes Formular zu nutzen, kannst du die Datei auch in dein Template unter **templates/deintemplatename/html/plg_content_jtf/deinformulardname/** kopieren und anpassen.

Erweiterte Formular (extended)

Das erweiterte Formular ist ein kleines Demo Formular. Es zeigt einige zusätzliche Felder an, welche für eine Kontaktanfrage häufig verwendet werden.

Dieses Formular ist unter **plugins/content/jtf/tmpl/extended/** gespeichert.

Um das erweiterte Formular zu nutzen, kannst du einfach den Plugin Aufruf Code **{jtf theme=extended}** überall auf deiner Webseite einsetzen wo Plugins ausgeführt werden können.

- Um das erweiterte Formular zu ändern, kannst du die benötigte framework spezifische XML-Datei aus **plugins/content/jtf/tmpl/extended/** in dein Template unter **templates/deintemplatename/html/plg_content_jtf/extended/** kopieren und anpassen.
- Um das erweiterte Formular als Basis für ein anderes Formular zu nutzen, kannst du die Datei auch in dein Template unter **templates/deintemplatename/html/plg_content_jtf/deinformulardname/** kopieren und anpassen.

Muster Form (muster)

Das Musterformular ist ein weiteres Demo Formular, welches verschiedene JForm Feldtypen mit verschiedenen Einstellungsmöglichkeiten und ein bisschen XML Magie anzeigt. Du wirst es mit größter Wahrscheinlichkeit nicht nutzen, aber einen Blick darauf werfen können was mit JT Formulador so möglich ist.

Dieses Formular ist unter **plugins/content/jtf/tmpl/muster/** gespeichert.

Um das Musterformular zu nutzen kannst du einfach den Plugin Aufruf Code **{jtf theme=muster}** überall auf deiner Webseite einsetzen wo Plugins ausgeführt werden können.

- Um das Musterformular zu ändern, kannst du die benötigte framework-spezifische XML-Datei aus **plugins/content/jtf/tmpl/muster/** in dein Template unter **templates/deintemplatename/html/plg_content_jtf/muster/** kopieren und anpassen.
- Um das Musterformular als Basis für ein anderes Formular zu nutzen, kannst du die Datei auch in dein Template unter **templates/deintemplatename/html/plg_content_jtf/deinformulardname/** kopieren und anpassen.

Plugin Einstellungen

CSS Frameworks

Der JT Formulador unterstützt verschiedene CSS-Frameworks wie Bootstrap 2, Bootstrap 3, UIKit 2 und UIKit 3 - um das Framework zu aktivieren, kannst du es entweder im Plugin-Aufruf individuell **{jtf framework=bs2}** ausführen oder global in den **Plugin Einstellungen** festlegen. Der Wert im Plugin-Aufruf überschreibt die globale Einstellung.

- Bootstrap 2 = {jtf framework=bs2}
- Bootstrap 3 = {jtf framework=bs3}
- UIKit 2 = {jtf framework=uiKit2}
- UIKit 3 = {jtf framework=uiKit3}

Betreff/Subject

Der Betreff der Mail, die zugestellt wird, wird in der XML durch das Feld mit dem Namen subject, oder mit dem Parameter **subject=deinbetreff** im Plugin-Aufruf festgelegt.

Beispiel: **{jtf subject=deinbetreff}**

Wenn das Feld nicht ausgefüllt ist und auch nicht im Formular abgefragt wird, wird dein Formular mit einem Standard-Betreff gesendet, der den globalen Seitennamen im Sprach-Schlüssel **JTF_EMAIL_SUBJECT** verwendet.

Empfänger

Der Empfänger der Mail, die zugestellt wird, wird durch das Feld mit dem Parameter **mailto=mail#empfaenger.de** im Plugin-Aufruf festgelegt.

Beispiel: `{jtf mailto=abteilungsleiter#empfaenger.de}`

Du kannst aber auch ein beliebiges Feld in den Parametern benennen, aus dem du den Empfänger ziehen möchtest. Zum Beispiel **mailto=feldname**. Der Wert im Feld muss dazu zwingend eine E-Mail-Adresse sein und darf auf keinen Fall leer sein.

CC- und BCC-Empfänger

Du kannst die Mails auch an bcc / cc-Empfänger versenden. Um dies zu erreichen definierst du **cc=mail#anotherrecipient.com** oder für bcc **bcc=mail#blindrecipient.com** im Plugin-Aufruf. Du kannst auch ein beliebiges Feld in den Parametern benennen, aus dem du den anderen Empfänger ziehen möchtest. Also **cc=feldname** bzw. **bcc=feldname**

Beispiel: `{jtf cc=chef#empfaenger.de}`

Tipp: Für die Empfänger-Parameter (mailto, cc, bcc) gilt: es können mehrere Empfänger, durch Semikolon getrennt, angegeben werden. Auch eine Kombination aus E-Mail-Adresse und Feld ist möglich. Zum Beispiel **cc=cc#example.org;feldname**. Auf diese Art kann man ein Formular an mehrere Empfänger senden.

Hinweis: Denke daran das @-Zeichen im Plugin Aufruf durch eine Raute # zu ersetzen. Das kann sonst dazu führen, das der Plugin Aufruf durch das E-Mail-Cloaking-Plugin zerstört wird.

Anwendungsbeispiel für die Übernahme von Feldwerten in den mailto, cc und bcc parameter:

Feld in der XML Datei:

```
<field name="email_list"
      type="list"
      icon="icon-list"
      description="JTF_LIST_DESC"
      label="JTF_LIST_LABEL"
      default=""
      validate="options"
      required="true"
      >

      <option value="">JTF_OPTION_NONE</option>
      <option value="empaenger1@example.org">Abteilung 1</option>
      <option value="empaenger2@example.org">Abteilung 2</option>

</field>
```

Aufruf im Plugin:

{jtf mailto=email_list}

Eigene Formulare erstellen

Struktur

Du kannst so viele Formulare erstellen, wie du möchtest. Du musst die XML Dateien einfach nur in dein Template in folgende Struktur einfügen:

templates/**deintemplate**/html/plg_content_jtf/**deinformulardname**/fields.xml

Joomla Core Felder

Der JT Formulador basiert auf jForm Standards mit erweiterten Möglichkeiten. Tatsächlich kannst du jedes Feld verwenden, das in Joomla möglich ist. Welche Felder Joomla anbietet findest du unter: https://docs.joomla.org/Form_field

Erweiterte Felder von JT Formulador

Der JT Formulador bringt noch ein paar eigene Feldtypen mit, die du nutzen kannst.

**Liste der JT Formulador JForm Felder hier
plz**

Erweiterte XML Parameter / XML Magic

Joomla Core Magic:

- hiddenLabel - versteckt das Label von einem Feld
- showon: - zeigt das Feld in abhängigkeit zu einem anderen Feld. Wie man das verwendet kannst du unter https://docs.joomla.org/Form_field#Showon nachlesen.

Zusätzliche JT Formulator Zaubereien (Attribute im XML-Feld)

- gridgroup
setzt eine Klasse in dem DIV-Container, welches das Label und das Input-Feld umschließt
- gridlabel
setzt eine Klasse in dem DIV-Container, welches nur das Label umschließt
- gridfield
setzt eine Klasse in dem DIV-Container, welches nur das Input-Feld umschließt
- icon
Icon-Klasse wie vom CSS-Framework vorgegeben, du gibst bei UI-Kit zum Beispiel uk-icon-envelope ein.
- optionclass/optionlabelclass
fügt eine CSS-Klasse in jedes Optionsfeld von Radio-, Checkboxes und Listenfelder ein
- buttonclass
bei Feldtypen mit einem Button, z.B. calendar, kann dem Button zusätzlich eine Klasse mitgegeben werden
- buttonicon
bei Feldtypen mit einem Button, z.B. calendar, kann man das Icon im Button ändern, indem man hier die Icon-Klasse angibt.

Eigenes Formular im Frontend darstellen

um dein eigenes Formular im Frontend darzustellen rufe **{jtf theme=deinformularname}** in deinem Joomla Beitrag oder Modul auf.

Overrides im Plugin Aufruf Code

Denke immer daran, dass du einige der XML Einstellungen beim Formularaufruf überschreiben kannst. Um das zu machen, schreibe die Parameter, getrennt durch Hochstriche (auch Pipes genannt |) in deinen Plugin Aufruf.

Beispiel: {jtf mailto=empfaenger@example.org | theme=meinformular | subject=Mein Betreff | fillouttime=10 | framework=uikit }

Multilanguage

Label Beschriftung überschreiben

Die globalen Sprach Dateien findest du unter **plugins/content/jtf/assets/language**
Du kannst diese überschreiben, indem du den language Ordner in deinen eigenen Template Ordner kopierst templates/**deintemplate**/html/plg_content_jtf/**deinformulardname**/language/

Dann kannst du die Sprach Strings dann ändern oder eigene hinzufügen.

Liste der Parameter für den Plugin Aufruf

- mailto (string)
Hauptempfänger
- cc (string)
Empfänger als Kopie
- bcc (string)
Empfänger als Blindkopie
- subject (string)
Betreff
- theme (string)
Vorlagenname (Templatename)
- framework (string)
CSS-Framework
- fillouttime (integer)
Zeit in Sek. bis der Submit Button zum Senden freigegeben wird
- message_article (integer)
ID des Beitrags, dessen Inhalt als Bestätigungstest bei erfolgreichem Absenden, angezeigt werden soll
- redirect_menuid (integer)
ID des Menüpunktes, wohin nach erfolgreichem Absenden, weitergeleitet werden soll

Infos

- JT - Formulatur wird nicht zusammen mit 'com_contact' ausgeführt.
- E-Mail Anhänge werden in der Email als Download Link mitgesendet