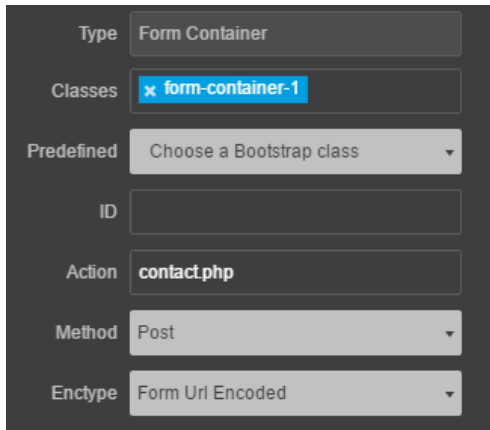# Tweaks for RSD forms

Forms can be styled and made very pretty in RSD, RFF and RBB. But there you don't have any back-end scripts, therefore they don't work. In SDv3 the interface in the programme is a bit different, but the same tweaks apply there.

With the php script provided here they do work. There are a couple of things to be aware of:

1. This is showing the Bootstrap side of RSD, but it is exactly the same in Foundation. Under Action the php file name has to appear, and also use the Post method.
2. You don't have to set anything Required, as that is taken care of in the php script (see later how to make other fields required).
3. When creating the input fields, it is important to give each of them a 'name'. That is because the php script is relying on those names to identify the various fields.
4. When the input fields are done, preview in a browser and check the source code. Look for something saying 'pattern=""', like at the end here:

```
<div class="container container1" id="name"><label class="label label1">Name
*</label><input value="" name="name" type="text" class="field" title=""
pattern="">
```

5. If you see that, remember it for later, when you are ready to save and export. After export, open the file, in my case contact.html, in the Editor (or any website builder of your choice), and remove it, unless of course, you know exactly what the pattern means and what is supposed to be put in between the "". I don't know, so I delete it. Otherwise the form doesn't go. If you create your form in Bootstrap, you apparently don't get that 'pattern' problem.
6. If you have used the form input 'dropdown', you won't be able to edit the text of the options in RSD, RFF and RBB. Therefore, while having the contact.html open, you need to add the different options manually. And each time you upload changes, this will be overwritten (also the pattern, because if left in there, the end user will get an error message when typing into the Name field), so I'm keeping a copy of my changes in a separate file so that I just have to paste them into where they belong.
7. Don't add the php file into the 'Resources', and don't link to it in the Settings. This is being taken care of in the form Action (see image above). The php file will have to be uploaded separately to the server.
8. The php file, here 'contact.php', can be used as good as untouched if you want to create the same kind of form that I have done. But most likely you don't want to ask everybody about the weather or how they got involved with Coffeecup, so on the next page are the things you need to be aware of.

```php
1   <?php
2   /* Set e-mail recipient */
3   $myemail  = "you@domain.com";
4
5   /* Check all form inputs using check_input function */
6   $name = check_input($_POST['name'], "Enter your name");
7   $phone = check_input($_POST['phone']);
8   $email    = check_input($_POST['email'], "Enter your e-mail address");
9   $website  = check_input($_POST['website']);
10  $subject  = check_input($_POST['subject'], "Write a subject");
11  $weather  = check_input($_POST['weather']);
12  $how = check_input($_POST['how']);
13  $comments = check_input($_POST['comments']);
14  $regex = check_input($_POST['regex']);
```

9. On line 3 you need to put in the recipient's mail address.

10. You can see that the names of the different input fields are being used, look at e.g. $phone which has the input name 'phone'. If you wish to have input for other things, e.g. for your car registration number plate, you give the input field the name of e.g. 'car' in RSD, and then here add a line like

    ```php
    $car    = check_input($_POST['car']);
    ```

11. Let's say you want to make that field required, you then change it to

    ```php
    $car    = check_input($_POST['car'], "Enter your car number");
    ```

    There you have made it required and also at the same time you have an error message handy in case it is forgotten.

```php
16  /* If e-mail is not valid show error message */
17  if (!preg_match("/([\w\-]+\@[\w\-]+\.[\w\-]+)/", $email))
18  {
19      show_error("E-mail address not valid");
20  }
21
22  /* If URL is not valid set $website to empty */
23  if (!preg_match("/^(https?:\/\/+[\w\-]+\.[\w\-]+)/i", $website))
24  {
25      $website = '';
26  }
27  /* Set the right answer for the regex captcha */
28  if($regex != "seven" && $regex != "Seven" )
29  {
30  show_error("You're not Human");
31  }
```

12. This image is checking if the email address and website URL are in the correct format. It also validates if the user's answer in the 'human' question is correct. So if you want to set up a different question, you need to pop the correct answers in here. **NB!** See addendum at the end of this file!

```
/* Let's prepare the message for the e-mail */
$message = "Hello!

Your contact form has been submitted by:

Name: $name
Phone: $phone
E-mail: $email
URL: $website

How's the weather? $weather
How did you find CC? $how

Comments:
$comments

End of message
";

/* Send the message using mail() function */
mail($myemail, $subject, $message);

/* Redirect visitor to the thank you page */
header('Location: thanks.html');
exit();
```

13.   This is first showing the details for the email message going to the recipient. You can add or subtract from that, e.g. Car: $car

14.   Then there is a list of parameters that are needed for the mail to the recipient: The sender's Subject and a summary of all the details submitted.

15.   At the bottom of this screenshot is the page that the visitor will be directed to after the form has been submitted. So obviously, such a page has to be created.

16.   Below that again (not in the picture here) is some security and validation stuff for the form.

I have not done much styling of this form, just made it responsive. The idea is, that those of you who want to use it, should style it to match the website it will be placed on. It is possible to save the form as a 'component', just select the whole row. But be aware that you need to save the file contact.php separately. Also, don't forget to make a 'thanks.html'.

This form does not cover the more advanced features that a form should have. I will see if I can develop it a bit further to include a few more things. But for anything regarding calculations, payments, posting to a database etc I recommend using Coffeecup's Web form Builder, in short called WFB.

Addendum:

I see that I have forgotten one thing: If you want to use the captcha I have described in this document, there is one more thing to be added after export. Find the line in the contact.html saying:

```
<div class="container"><label class="label label1">Prove that you are
human</label><input value="" name="regex" type="text" class="field"
required="required" id="regex">
```

Change it to read:
```
<div class="container"><label class="label label1">Prove that you are
human</label><input value="" placeholder="How much is 3+4? Spell it."
name="regex" type="text" class="field" required="required" id="regex">
```

You can of course change the captcha to something else. Then remember to edit the answer in contact.php too.

Good luck!

Inger