Helpers

**PART 1 - JAVASCRIPT (JS) and AJAX (remote calls)**

---

### JAVASCRIPT AND AJAX HELPERS

**JavaScript Helpers**

- `link_to_function ($name, $function, $html_options=array())`
- `javascript_tag ($content)`
- `update_element_function ($element_id, $options=array())`

**Ajax Helpers**

An AJAX interaction is made up of three parts:

* a caller (a link, button or any control that the user manipulates to launch the action)
* a server action
* a zone in the page to display the result of the action to the user.

Symfony provides multiple helpers to insert AJAX interaction in your templates by putting the caller in a link, a button, a form, or a clock. These helpers output HTML code, not JavaScript.

- `link_to_remote ($name, $options=array(), $html_options=array())`
- `remote_function ($options=array())`
- `form_remote_tag ($options=array(), $options_html=array())`
- `observe_field ($field_id, $options=array())`
- `periodically_call_remote ($options=array())`

### REMOTE CALL PARAMETERS

All the AJAX helpers can take other parameters, in addition to the `update` and `url` parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>position</td>
<td>The position parameter can be defined as: Value</td>
</tr>
<tr>
<td>before</td>
<td>before the element after the element</td>
</tr>
<tr>
<td>after</td>
<td>at the top of the element bottom of the content element</td>
</tr>
</tbody>
</table>

### Conditions

- `confirm` => `'Are you sure?'`
- `condition` => `$('elementId') == true`.

Remote call can also be conditioned by a test performed on the browser side (in JavaScript).

### Callbacks

- `before` => before request is initiated
- `after` => immediately after request is initiated and before loading
- `loading` => when the remote response is being loaded
- `loaded` => when the browser has finished loading the remote response
- `interative` => when the user can interact with the remote response, even though it has not finished loading
- `success` => when the XMLHttpRequest is completed, and the HTTP status code is in the 2XX range
- `failure` => when the XMLHttpRequest is completed, and the HTTP status code is not in the 2XX range
- `404` => when the request returns a 404 status

#### Examples

- `<?php echo link_to_remote('Delete this post', array('update' => 'indicator', 'url' => 'post/delete?id='.$post->getId())) ?>`
- `<?php echo form_remote_tag(array('update' => 'myzone', 'url' => '@item_add')) ?>`
- `<?php echo observe_field('@item_add_regular') ?>`
- `<?php echo periodically_call_remote(array('frequency' => 60, 'update' => 'notification', 'url' => '@watch', 'with' => array('param' => '(mycontent).value')) ?>`

### Notes

- *Actions called as remote functions know that they are in an AJAX transaction, and therefore automatically don’t include the web debug toolbar in development. Also, they skip the decoration process (their template is not included in a layout by default). If you want an Ajax view to be decorated, you need to specify explicitly has_layout: true for this view in the module view.yml file. Actions called through Ajax, return true to the following call:*  
  ```
  $isAjax = this->isXmlHttpRequest();
  ```

* The AJAX helpers won’t work if the URL of the remote action doesn’t belong to the same domain as the current page. This restriction exists for security reasons, and relies on browsers limitations that cannot be bypassed.