

Helpers

PARTE 1 - JAVASCRIPT(JS) e AJAX (chamadas remotas) <?php echo use_helper('Javascript') ?>

HELPERS JAVASCRIPT E AJAX

Helpers JavaScript

link_to_function (\$name, \$function, \$html_options=array())
 <?php echo **link_to_function**("Click me!", "alert('foobar')") ?> // irá gerar:
 Click me!

javascript_tag (\$content)
 <?php echo **javascript_tag**("document.getElementById('indicator').innerHTML='Data processing complete';") ?>

update_element_function (\$element_id, \$options=array())
 <?php echo **javascript_tag**(**update_element_function**('indicator', array("position"=>"after", "content" =>"Data processing complete")))?>

Helpers Ajax

Uma interação AJAX é composta de três partes:
 * um **caller** (link, botão ou qualquer controle, através do qual o usuário dispara a ação)
 * uma **ação do servidor**
 * uma **região na página para exibir o resultado da ação ao usuário**.
 O Symfony disponibiliza múltiplos helpers para inserir interações AJAX em seus templates, adicionando o **caller** em um link, um botão, um formulário ou uma chamada periódica. O resultado da execução destes helpers é código HTML, não JavaScript.

link_to_remote (\$name, \$options=array(), \$html_options=array())
 <?php echo **link_to_remote**('Delete this post', array('update' => 'indicator', 'url' => 'post/delete?id='.\$post->getId())) ?>

remote_function (\$options=array())
 <?php echo **javascript_tag**(**remote_function**(array('update' => 'myzone', 'url' => 'mymodule/myaction'))) ?>

modifica parte da página de acordo com a resposta do servidor

form_remote_tag (\$options=array(), \$options_html=array())
 <?php echo **form_remote_tag**(array('update' => 'item_list', 'url' => '@item_add')) ?>
 <label for="item">Item:</label>
 <?php echo **input_tag**('item') ?>
 <?php echo **submit_tag**('Add') ?>
 </form>

abre uma tag <form>, exatamente como faz o helper form_tag()

observe_field (\$field_id, \$options=array())
 <?php echo **form_tag**('@item_add_regular') ?>
 <label for="item">Item:</label>
 <?php echo **input_tag**('item') ?>
 <?php echo **submit_tag**('Add') ?>
 <?php echo **observe_field**('item', array('update' => 'item_suggestion', 'url' => '@item_being_typed')) ?>
 </form>

o módulo/ação escrito na regra @item_being_typed rule será chamado sempre que o campo field for modificado, e a ação será capaz de obter o valor atual do item vindo do parâmetro do pedido (request). Se desejar passar outro valor além do campo observado é possível especificar como uma expressão JavaScript no parâmetro 'with'

periodically_call_remote (\$options=array())
 <?php echo **periodically_call_remote**(array('frequency' => 60, 'update' => 'notification', 'url' => '@watch', 'with' => "param=' + \$(mycontent).value')) ?>

este helper é uma interação ajax ativada a cada x segundos. Ela não está associada a um controle HTML, mas executa transparentemente em background, como um comportamento da página inteira.

Outras funções

- submit_to_remote (\$name, \$value, \$options=array())
- evaluate_remote_response ()
- observe_form (\$form_id, \$options=array())
- visual_effect (\$name, \$element_id=false, \$js_options=array())
- sortable_element (\$element_id, \$options=array())
- draggable_element (\$element_id, \$options=array())
- drop_receiving_element (\$element_id, \$options=array())
- javascript_cdata_section (\$content)
- input_auto_complete_tag (\$name, \$value, \$url, \$tag_options=array(), \$completion_options=array())
- input_in_place_editor_tag (\$name, \$url, \$editor_options=array())

PARÂMETROS PARA CHAMADAS REMOTAS

É possível informar outros parâmetros, em adição ao **update** e **url**, para todos os helpers AJAX:

position

O parâmetro **position** pode ser definido como:

Valor	Posição
before	antes do elemento
after	após o elemento
top	acima do conteúdo do elemento
bottom	abaixo do conteúdo do elemento

condições

confirm

'confirm' => 'Are you sure?'

Uma caixa de diálogo **JS** exibindo 'Are you sure?' será apresentada quando o usuário clicar no **caller**, e o módulo/ação será chamado somente se o usuário confirmar sua escolha clicando no botão 'Ok'.

condition

'condition' => "\$('elementID') == true",

A chamada remota também pode ser condicionada por um teste executado pelo browser (em JavaScript).

execução de script

'script' => true

Se o código de resposta da chamada AJAX (o código enviado pelo servidor, inserido no elemento **update**) contém **JS**, estes scripts, por padrão, não serão executados, para prevenir riscos de ataque remotos. A habilidade de executar scripts nas respostas remotas é explícita com a opção **script**.

callbacks

Callback	Evento
before	antes do pedido ser iniciado
after	imediatamente após o pedido ser iniciado e antes de carregar
loading	quando a resposta remota está sendo carregada
loaded	quando o browser terminou de carregar a resposta remota
interactive	quando o usuário pode interagir com a resposta remota, embora não terminou de carregar
success	quando o XMLHttpRequest está completo, e o código do status HTTP está na faixa 2XX
failure	quando o XMLHttpRequest está completo, e o código do status HTTP não está na faixa 2XX
404	quando o pedido retorna o status 404
complete	quando o XMLHttpRequest está completo (executa após o success ou failure , se presente)

ex.: <?php echo **link_to_remote**('Delete this post', array('update' => 'indicator', 'url' => 'post/delete?id='.\$post->getId(), 'position' => 'after', 'confirm' => 'Are you sure?', 'script' => true, 'loading' => "Element.show('indicator')", 'complete' => "Element.hide('indicator')")) ?>

NOTAS

* As **ações** chamadas como **funções remotas** sabem que estão em uma transação AJAX e, automaticamente, **não incluem a barra de ferramenta de debug** em desenvolvimento. Elas também saltam o processo de decoração (por default, seu template não é incluído no layout). Se desejar que uma view Ajax seja decorada, é necessário especificar explicitamente **has_layout: true** para esta view no arquivo **view.yml** do módulo. As ações chamadas através de AJAX, retornam true para a seguinte chamada:

```
$isAjax = $this->isXmlHttpRequest();
```

* Os helpers AJAX não funcionarão se a URL da ação remota não pertencer ao mesmo domínio da página atual. Esta restrição existe por razões de segurança.