Formulários - sfForm (Parte I)

Um formulário do symfony (**sfForm**) é uma classe composta de campos. Cada campo possui um nome, um validador e um widget.

FORMULÁRIO HTML

Nome
E-mail
Mensagem
Enviar

Os campos do form são configurados no método **configure()** usando os métodos **setValidators()** e **setWidgets()**

1 FORMULÁRIO DO SYMFONY: Definindo Widgets e Validadores

```
class ContactForm extends sfForm {
  public function configure() {
    $this->setWidgets(array(
      'nome'     => new sfWidgetFormInputText(),
      'email'    => new sfWidgetFormInputText(),
      'mensagem' => new sfWidgetFormTextarea(),
    ));

    $this->setValidators(array(
      'email'    => new sfValidatorEmail(),
      'mensagem' => new sfValidatorString(array('max_length' => 255)),
    ));
  }
}
```

lib/form/ContactForm.class.php

sfWidgetSchema processa o form em XHTML, por padrão. Para trocar para HTML usar: sfWidget::setXhtml(false);

sfValidatorSchema valida os dados brutos informados limpando/convertendo eles para dados válidos e consistentes.

2 ACTION

```
function executeIndex($request){
  $this->form = new ContactForm(array(/* initial values */));

  if ($request->isMethod('post')) {
    $this->form->bind($request->getParameter('contact'));
    if ($this->form->isValid()) {
      // do something with cleaned values
      $values = $this->form->getValues();
      $this->redirect('@algumlugar');
    }
  }
}
```

apps/frontend/modules/mymodule/actions/actions.class.php

Cria o formulário e passa ele para a visão (usando $this->).

Verifica se existe uma solicitação POST.

Vincula os dados da requisição e dispara a validação.
A validação é gerenciada pelo esquema de validação e não pelo formulário.

Verifica se o formulário é válido.

3 TEMPLATE

```
<?php echo $form->renderFormTag('foo/contact') ?>
  <?php echo $form->renderHiddenFields() ?>
  <?php echo $form->renderGlobalErrors() ?>
  <table>
    <?php echo $form['nome']->renderRow() ?>
    <?php echo $form['email']->renderRow() ?>
    <?php echo $form['mensagem']->renderRow() ?>
    <tr>
      <td colspan="2">
        <input type="submit" />
      </td>
    </tr>
  </table>
</form>
```

apps/frontend/modules/mymodule/templates/indexSuccess.php

LOCALIZAÇÃO DAS CLASSES DOS FORMULÁRIOS

- lib
  - form — suas classes personalizadas e classes do propel (geradas usando a tarefa: **$ symfony propel:build-forms**)
    - doctrine — as classes do doctrine (geradas usando a tarefa **$ php symfony doctrine:build --forms**)

O QUE POSSO FAZER NO MÉTODO configure()?

RETIRAR CAMPOS: unset

```
unset($this['created_at'], $this['updated_at'], $this['expires_at'] );
```

LISTAR OS CAMPOS QUE DESEJA EXIBIR: useFields

O método useFields() faz duas coisas automaticamente: adiciona os campos ocultos e o array dos campos é usado para alterar a ordem dos mesmos.

```
$this->useFields(array('id', 'type', 'description', 'token', 'is_public', 'email'));
```

DEFINIR A POSIÇÃO DOS CAMPOS: setPositions

```
$this->widgetSchema->setPositions(array('bodywork_id', 'mark_id',
'price', 'fuel_id', 'model_id'));
```

DEFINIR WIDGETS: setWidget and setWidgets

```
$this->setWidgets(array('password' => new sfWidgetFormInputPassword()));
```

DEFINIR UMA MENSAGEM DE AJUDA PARA OS CAMPOS: setHelp and setHelps

```
$this->widgetSchema->setHelp(
    'phone', 'Only numbers'
);
```

or

```
$this->widgetSchema->setHelps(array(
  'phone'     => 'Only numbers',
  'is_public' => 'Public?',
));
```

EMBUTIR OBJETOS I18N (doctrine) (propel)

```
$this->embedI18n(array('pt', 'en', 'es'));
```

VALIDAR CAMPOS: setValidator and setValidators

```
$this->setValidator('filename', new sfValidatorFile(array('mime_types'=>'web_images')));
```

DEFINIR UMA LABEL PERSONALIZADA PARA OS CAMPOS: setLabel and setLabels

```
$this->widgetSchema->setLabel(
    'pt', 'Português'
);
```

or

```
$this->widgetSchema->setLabels(array(
  'category_id'    => 'Category',
  'is_public'      => 'Public?',
  'how_to_apply'   => 'How to apply?',
));
```

DEFINIR VALORES DEFAULT PARA OS CAMPOS: setDefault and setDeafults

```
$this->setDefault(
    'email', 'Your Email Here'
);
```

or

```
$this->setDefaults(array(
    'email' => 'Your Email Here',
    'name' => 'Your Name Here'
));
```

MODIFICAR O NOME DO ATRIBUTO HTML PARA TODOS OS WIDGETS: setNameFormat

```
$this->widgetSchema->setNameFormat('search[%s]');
```

EMBUTIR RELAÇÕES: embedRelation (doctrine) (propel)

```
$this->embedRelation('Games');
```

EMBUTIR OUTROS FORMULÁRIOS NO FORMULÁRIO ATUAL: embedForm

```
$authorForm = new AuthorForm($this->getObject()->getAuthor());
$this->embedForm('Author', $authorForm);
```

MESCLAR FORMULÁRIOS: mergeForm

```
$this->mergeForm(new RegisterForm());
```