



Como Funcionam os Formulários no Symfony2?

Uso básico

Symfony

Dados são válidos	Dados são inválidos
Formulário passa para o estado "bound" e você pode, por exemplo, persistir os dados no BD antes de redirecionar o usuário para outra página	Formulário passa para o estado "bound" e é renderizado, exibindo todos os erros de validação

Tipos de campos nativos

Campos Texto text textarea email integer money number password percent search url	Campos Escolha choice entity country language locale timezone	Campos Data e Hora date datetime time birthday	Campos Grupos collection repeated	Campos Ocultos hidden csrf	Outros Campos checkbox file radio	Campos Base field form
--	--	---	--	---	---	-------------------------------------

Fragments Templates

Os fragments são definidos como blocos no Twig e arquivos template no PHP.

label (ex. field_label)	renderiza a label do campo
widget (ex. field_widget)	renderiza a representação HTML do campo
errors (ex. field_errors)	renderiza os erros do campo
row (ex. field_row)	renderiza a linha inteira do campo (label, widget e erros)
rows (ex. field_rows)	renderiza todos os campos do formulário
rest (ex. form_rest)	renderiza quaisquer campos que ainda não tenham sido renderizados
 enctype (ex. form enctype)	se pelo menos um campo for para upload de arquivos, irá renderizar: enctype="multipart/form-data"

```
{% block field_row %}
<div class="form_row">
  {{ form_label(form) }}
  {{ form_widget(form) }}
  {{ form_errors(form) }}
</div>
{% endblock field_row %}
```

Twig

Entity

representa e armazena os dados

src/Acme/TaskBundle/Entity/Task.php

```
namespace Acme\TaskBundle\Entity;

use Symfony\Component\Validator\Constraints as Assert;

class Task{
    /**
     * @Assert\NotBlank()
     */
    public $task;

    /**
     * @Assert\NotBlank()
     * @Assert\Type("DateTime")
     */
    protected $dueDate;

    public function getTask() {
        return $this->task;
    }

    public function setTask($task) {
        $this->task = $task;
    }

    public function getDueDate() {
        return $this->dueDate;
    }

    public function setDueDate(DateTime $dueDate = null) {
        $this->dueDate = $dueDate;
    }
}
```

Validação

Aplicada ao objeto (classe) adicionando um conjunto de regras (constraints). Essas regras podem ser especificadas em YAML, XML, anotações ou PHP

usando anotações para as constraints

vincula os dados submetidos ao formulário, que traduz os dados de volta as propriedades task e dueDate do objeto \$task

pergunta ao objeto \$task se ele tem ou não dados válidos

A menos que a propriedade seja pública, ela deve ter um método "getter" e "setter" para que o componente de formulário possa obter e colocar dados na propriedade

Controller

src/Acme/TaskBundle/Controller/DefaultController.php

```
$task = new Task();
$task->setTask('Write a blog post');
$task->setDueDate(new DateTime('tomorrow'));

$form = $this->createFormBuilder($task)
    ->add('task', 'text')
    ->add('dueDate', 'date')
    ->getForm();

if ($request->getMethod() == 'POST') {
    $form->bindRequest($request);

    if ($form->isValid()) {
        // perform some action, such as saving the task to DB
        return $this->redirect($this->generateUrl('task_success'));
    }
}

return $this->render('AcmeTaskBundle:Default:new.html.twig',
    array(
        'form' => $form->createView(),
    ));
```

Cria o formulário

Tipos de campos nativos

submetendo o formulário (POST)

renderiza o formulário

bindRequest

traduz os dados submetidos pelo usuário de volta as propriedades de um objeto

isValid

verifica se há dados válidos no objeto

createView

renderiza o formulário

O Formulário

Task

Due date Jul 24, 2011

renderiza o formulário

View

src/Acme/TaskBundle/Resources/views/Default/new.html.twig

```
<form action="{{ path('task_new') }}" method="post"
  {{ form_enctype(form) }}>
  {{ form_widget(form) }}
  <input type="submit" />
</form>
```

OR

src/Acme/TaskBundle/Resources/views/Default/new.html.php

```
<form action="php echo $view['router']-&gt;generate('task_new') ?"
  method="post" <?php echo $view['form']->enctype($form) ?>
  <?php echo $view['form']->widget($form) ?>
  <input type="submit" />
</form>
```

Twig

PHP