



Symfony

Data is Valid	Data is invalid
Form is bound and you can for e.g. persist data to DB before redirecting the user to other page	Form is bound and rendered, displaying all validation errors

Entity

represent and store the data

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;
    }

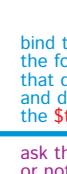
    public function setDueDate(\DateTime $dueDate = null) {
        $this->dueDate = $dueDate;
    }
}
```

Validation
Applied to the underlying object (class) by adding a set of rules (constraints). These rules can be specified in YAML, XML, annotations, or PHP

bind the submitted data to the form, which translates that data back to the task and dueDate properties of the \$task object

ask the \$task object whether or not it has valid data

Unless a property is public, it must have a "getter" and "setter" method so that the form component can get and put data onto the property



Built-in Field Types

Text Fields	Choice Fields	Date and Time Fields	Field Groups
text	choice	date	collection
textarea	entity	datetime	repeated
email	country	time	
integer	language	birthday	
money	locale		
number	timezone		
password			
percent			
search			
url			

Hidden Fields	Other Fields	Base Fields	
hidden	checkbox	field	
csrf	file	form	
	radio		

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(),
    ));
}
```

bindRequest

translate user-submitted data back to the properties of an object

isValid

check for valid data in the object

How Symfony2 Forms Works?

Basic usage

Fragments Templates

The fragments are defined as blocks in Twig and as template files in PHP.

label (e.g. field_label) renders the field's label

widget (e.g. field_widget) renders the field's HTML representation

errors (e.g. field_errors) renders the field's errors

row (e.g. field_row) renders the field's entire row

label, widget & errors

rows (e.g. field_rows) renders all form field rows

rest (e.g. form_rest) renders any fields that have not yet been rendered

enctype (e.g. form_enctype) if at least one field is a file upload field, this renders the enctype="multipart/form-data"

Twig

The Form

Task

Write a blog post
Duedate Jul 24, 2011
Submit



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

```
<form action="{{ <?php echo $view['router']->generate(task_new) ?> }}"
      method="post" <?php echo $view['form']->enctype($form) ?>>
<?php echo $view['form']->widget($form) ?>
```

<input type="submit" />

Twig

PHP