



Desarrollo de aplicaciones con Symfony 4

Validaciones



Validaciones

La validación es una tarea muy común en aplicaciones web. Los datos ingresados en los formularios deben ser validados. Los datos también deben validarse antes de que se escriban en una base de datos o se pasen a un servicio web.

Symfony proporciona un componente Validator que hace que esta tarea sea fácil y transparente.

Validaciones

En caso de haber optado por la instalación básica de symfony, para poder instalar el componente deberemos ejecutar:

```
$ composer require symfony/validator doctrine/annotations
```

Validaciones

Partimos de nuestra entidad:

```
<?php
```

```
....
```

```
/**
```

```
 * @ORM\Column(type="string", length=255)
```

```
 */
```

```
private $title;
```

```
.....
```

Validaciones

```
<?php
```

```
...
```

```
+ use Symfony\Component\Validator\Constraints as Assert;
```

```
...
```

```
/**
```

```
* @ORM\Column(type="string", length=255)
```

```
* @Assert\NotBlank
```

```
*/
```

```
private $title;
```

```
.....
```

Validaciones

¿Cómo podemos personalizar los mensajes de error?

* `@Assert\NotBlank(message="El título no puede estar vacío")`

Validaciones

Echemos un vistazo a las diferentes validaciones que trae Syfmony

<https://symfony.com/doc/current/validation.html#supported-constraints>

****También podremos crear nuestros propios validators**

Usando el validator service

No siempre vamos a querer validar nuestra entidad utilizando formularios. Por ejemplo, podemos tener un api que cree objetos y querer validarlos antes de persistirlos.

Para ello tenemos el validator service. Para usarlo en nuestro controlador o servicio, inyectamos el ValidatorInterface:

...

```
public function new(ValidatorInterface $validator)
```

....

Y podremos utilizarlo:

```
$errors = $validator->validate($task);
```


Usando el validator service

Antes de usar el validador de Symfony, asegúrate de que esté habilitado en el archivo de configuración principal:

```
# config/packages/framework.yaml  
framework:  
    validation: { enable_annotations: true }
```

Validation Groups

Por defecto, al validar un objeto, todas las restricciones de esta clase serán verificadas si realmente pasan o no. En algunos casos, sin embargo, tendrá que validar un objeto contra solo algunas restricciones en esa clase. Para hacer esto, puede organizar cada restricción en uno o más "grupos de validación" y luego aplicar la validación a solo un grupo de restricciones.

Validation Groups

```
class User implements UserInterface{  
    /**  
     * @Assert\Email(groups={"registration"})  
     */  
    private $email;  
    /**  
     * @Assert\NotBlank(groups={"registration"})  
     * @Assert\Length(min=7, groups={"registration"})  
     */  
    private $password;  
    /**  
     * @Assert\Length(min=2)  
     */  
    private $city;  
}
```

Validation Groups

Default

Contiene las restricciones en la clase actual y todas las clases referenciadas que no pertenecen a ningún otro grupo. En este ejemplo, solo contiene el atributo de la city.

User

Equivalente a todas las restricciones del objeto Usuario en el grupo Default.

registration

Este es un grupo de validación personalizado, por lo que solo contiene las restricciones explícitamente asociadas a él. En este ejemplo, solo los campos de correo electrónico y contraseña.

*Las restricciones en el grupo Default de una clase son las restricciones que no tienen un grupo explícito configurado o que están configuradas en un grupo igual al nombre de la clase o la cadena Default.

Validation Groups

¿Como los aplicamos?

```
$errors = $validator->validate($author, null, ['registration']);
```

o en el formulario creado en un controller:

```
$form = $this->createFormBuilder($user, [  
    'validation_groups' => ['registration']  
]);
```

Validation Groups

Y en la clase Type:

```
public function configureOptions(OptionsResolver $resolver)
{
    $resolver->setDefaults([
        'validation_groups' => ['registration'],
    ]);
}
```

Validation Groups

**Es posible utilizar más de un grupo

```
'validation_groups' => ['Default', 'registration']
```

¿Preguntas?

