



Desarrollo de aplicaciones con Symfony 4

Otros temas importantes



Symfony 4: Bundles

Uno de los principales cambios de Symfony 4 respecto a las versiones anteriores ha sido la eliminación por defecto de bundles dentro de la carpeta src. ¡Pero aún se pueden crear!

<https://symfony.com/doc/current/bundles.html>

Symfony 4: Migrations

Las migraciones de base de datos son una forma de actualizar de forma segura el esquema de su base de datos tanto localmente como en producción. En lugar de ejecutar el comando `doctrine:schema:update` o aplicar los cambios de la base de datos manualmente con sentencias de SQL, las migraciones permiten replicar los cambios en el esquema de su base de datos de una manera segura.

<https://symfony.com/doc/master/bundles/DoctrineMigrationsBundle/index.html>

Symfony 4: Param converters

La anotación `@ParamConverter` transforma parámetros de la url en objetos. Estos objetos se almacenan como atributos de solicitud y, por lo tanto, se pueden inyectar como argumentos de método de controlador:

```
/**
 * @Route("/blog/{id}")
 * @ParamConverter("post", class="SensioBlogBundle:Post")
 */
public function show(Post $post)
{
}
```

<https://symfony.com/doc/master/bundles/SensioFrameworkExtraBundle/annotations/converters.html>

Symfony 4: Form Events

El componente Formulario proporciona un proceso estructurado que le permite personalizar sus formularios, haciendo uso del componente EventDispatcher. Mediante el uso de eventos de formulario, puede modificar la información o los campos en diferentes pasos del flujo de trabajo: desde la población del formulario hasta el envío de los datos de la solicitud.

<https://symfony.com/doc/current/form/events.html>

Como vas avanzado tenemos los EventSubscriber en formularios.

https://symfony.com/doc/current/form/dynamic_form_modification.html

Symfony 4: Data Transformers

Los data transformers se utilizan para traducir los datos de un campo a un formato que se puede mostrar en un formulario y viceversa.

https://symfony.com/doc/current/form/data_transformers.html

Symfony 4: Swift Mailer

Symfony proporciona una función de correo electrónico basada en la popular biblioteca Swift Mailer a través de SwiftMailerBundle.

Esta aplicación de correo admite el envío de mensajes con nuestros propios servidores de correo, así como el uso de proveedores de correo populares como Gmail, Mandrill, Amazon SES...

<https://symfony.com/doc/current/email.html>

Importante durante el desarrollo:

https://symfony.com/doc/current/email/dev_environment.html

Symfony 4: Bundles de terceros interesantes

beberlei/**DoctrineExtensions**

<https://github.com/beberlei/DoctrineExtensions>

Knplabs/**KnnpaginatorBundle**

<https://github.com/Knplabs/KnpPaginatorBundle>

cocur/**slugify**

<https://github.com/cocur/slugify>

craue/**CraueFormFlowBundle**

<https://github.com/craue/CraueFormFlowBundle>

Symfony 4: Mis recomendaciones

Primera y más importante **RTFM**; la documentación de Symfony es muy buena (al igual que la de PHP) y antes de reinventar la rueda mira si Symfony ya ha contemplado el problema al que enfrentas.

Mira el código de symfony y el bundles de terceros.

Aplica buenas prácticas de programación y standards

PSR: <https://www.php-fig.org/psr/>

SOLID: <https://profile.es/blog/principios-solid-desarrollo-software-calidad/>

Sigue a la comunidad

¡Espero que el curso os
haya sido de utilidad!

