

Read: **Ejemplo de implementación de Command**

09:27

Ya hemos visto la teoría de Commands, Queries, y sus correspondientes buses. En este vídeo veremos un ejemplo de implementación al respecto de los comandos.

Creando un nuevo Vídeo

Petición

En el curso de testing veremos más en detalle, pero por empezar por el principio (la petición POST para crear el vídeo), [aquí tenéis el test de Behat](#) (para gente no proveniente de PHP, es un framework de test de comportamiento como lo podría ser Cucumber, implementa un lenguaje Gerkin).

Punto de entrada

El punto de entrada de nuestra aplicación será el [Controlador encargado de recibir la petición POST](#). Este controlador lo único que hará será obtener los

datos de la petición, y encapsularlos en [el CreateVideoCommand](#) para poder lanzarlo al CommandBus.

CommandBus

Como decíamos, el controlador lanza el comando a la implementación concreta de [la interface de dominio CommandBus](#) que se le haya inyectado. En este caso, la implementación será la del [CommandBusSync](#).

Este CommandBus tendrá el mapeo del handler correspondiente al que debe trasladar el comando.

CommandHandler

En este caso, ejecutaremos el [CreateVideoCommandHandler](#). Responsable como sabemos de recibir el comando con primitivos e instanciar objetos del dominio en forma de value objects como por ejemplo el [VideoUrl](#). Estos Value Objects, como se ve en el ejemplo del VideoUrl son los encargados de encapsular la lógica correspondiente a cada uno de los elementos. En este caso concreto, validar que es una URL correcta.

Una vez ha instanciado todos los objetos de dominio, el CommandHandler sabe qué ApplicationService/UseCase/Action ejecutar ya que la tiene inyectada. Con lo cuál, invoca este servicio pasándole todos los parámetros necesarios con su tipo correspondiente.

Caso de uso

Una vez el caso de uso recibe la petición, simplemente ejecuta la lógica correspondiente. En el caso del [VideoCreator](#), instanciará [la entidad/agregado](#)

[Video](#), la guardará en el repositorio, y publicará el evento de dominio correspondiente.

El publicar el evento de dominio es un paso crítico ya que es lo que permitirá que luego nos subscribamos a este evento para poder almacenar estos datos como más nos convenga para el modelo de lectura (queries). Esto lo veremos en el curso de Read Model/Write Model.