

Read: **¿Qué es una Query?**

09:07



Qué es una Query (en CQRS)

Normalmente si alguien nos dice Query nos viene a la mente SQL. En el contexto de CQRS no es así.

En CQRS, una Query representa la intención de pedir unos datos a nuestro sistema sin que ello acabe alterando el estado de tal.

Cuando nos viene bien utilizarlo

A la que tengamos un sistema con un volumen alto de tráfico ya nos podremos aprovechar de las ventajas que nos aporta.

Al no tener side-effects nos viene muy bien para poder cachearlas.

Ejemplo

Fuera de la teoría, la Query en sí no es más que un DTO (objeto de transferencia de datos) el cuál representa la petición que queremos consultar. Por ejemplo, si quisiéramos ver el detalle de un vídeo:

```
final class FindVideoQuery implements Query
{
    private $id;
    public function __construct(string $id)
    {
        $this->id = $id;
    }
    public function id(): string
    {
        return $this->id;
    }
}
```

Esta Query nos llegaría a nuestro QueryHandler, el cuál crearía el **Value Object** de **VideoId** (de esta forma estaríamos validando que la Query cumpla con los requisitos de nuestro dominio).

Luego el QueryHandler enviaría estos datos al caso de uso (**VideoFinder**), el cuál ya se encargaría de ir al Repositorio para recuperar el vídeo, devolverlo y transformarlo en un **VideoResponse**.

Enlaces relacionados

- [Ejemplo FindVideoQuery](#).

Conclusión

El trabajar con Queries nos ayuda a separar mejor nuestro dominio, también nos permite cachear muy fácilmente (al no tener side-effects).

En el curso de Read Model\Write Model veremos como podemos tener *proyecciones* las cuales nos ayudarán a que nuestras Queries sean más simples, veloces y eficaces.