

# Что такое CQRS?

Влад Чесноков



# В начале был CQS

CQS — Command–query separation

Каждый метод:

- либо команда для выполнения действия, без возврата;
- либо запрос, который не меняет данные.

# В начале был CQS

Вы уже знакомы с ним

`array.sort()` – команда

`val first = array[0]` – запрос

# CQS + R = CQRS

CQRS – Command and Query Responsibility Segregation

- CQS + для обработки запросов и команд используются **разные** сущности (сервисы) (одна для запросов, другая для команд).

# CQS + R = CQRS

CQRS – Command and Query Responsibility Segregation

- CQS + для обработки запросов и команд используются **разные** сущности (сервисы) (одна для запросов, другая для команд).

И все!

Спасибо за внимание!

# FAQ

# FAQ

Нужны ли две базы для реализации CQRS?



# FAQ

Нужны ли две базы для реализации CQRS?

НЕТ

Главное – разделить потоки. А база может быть одна.

Нужны ли две модели для реализации CQRS?

Нужны ли две модели для реализации CQRS?

ДА, но не всегда

За сервисами чтения и записи могут скрываться общие модели, но в большинстве случаев это не будет оптимальным решением.

Команда совсем ничего не возвращает?

# FAQ

Команда совсем ничего не возвращает?

**НЕ ОБЯЗАТЕЛЬНО**

Можно возвращать результат операции, например id созданного объекта.

Обязательно иметь Event-Driven (в т.ч. Event Sourcing) для реализации CQRS?

Обязательно иметь Event-Driven (в т.ч. Event Sourcing) для реализации CQRS?

НЕТ

Часто реализуются вместе, но напрямую не связаны друг с другом.

# FAQ

Сервис для операций + сервис для отчетности или поиска – это CQRS?



# FAQ

Сервис для операций + сервис для отчетности или поиска – это CQRS?

НЕТ

Потому что сервис для операций и читает, и пишет.

На каких уровнях архитектуры нужен CQRS?

# FAQ

На каких уровнях архитектуры нужен CQRS?

IT DEPENDS

Разделение может быть на нижних слоях, а на верхних никто ничего не заметит – не обязательно кидать это до контроллеров

# Как применять

# Кейсы применения

- Неравномерная нагрузка на чтение и запись

# Кейсы применения

- Неравномерная нагрузка на чтение и запись
- Дисбаланс между чтением и записью на уровне бизнеса

# Кейсы применения

- Неравномерная нагрузка на чтение и запись
- Дисбаланс между чтением и записью на уровне бизнеса
- Проще для фронта (если нет BFF)  
(в т.ч. оптимизация запросов на чтение)

# Кейсы применения

- Неравномерная нагрузка на чтение и запись
- Дисбаланс между чтением и записью на уровне бизнеса
- Проще для фронта (если нет BFF)  
(в т.ч. оптимизация запросов на чтение)



# Extract it when it grows

- Общая база
- Чтение с реплики
- Отдельная база для чтения
- Event Sourcing

# Потенциальные проблемы

# Потенциальные проблемы

- Eventual consistency

# Потенциальные проблемы

- Eventual consistency
- По коду может быть тяжело связать команду и ее обработчик

# Потенциальные проблемы

- Eventual consistency
- По коду может быть тяжело связать команду и ее обработчик
- ORM может стать узким местом – лучше возвращать DTO

# Потенциальные проблемы

- Eventual consistency
- По коду может быть тяжело связать команду и ее обработчик
- ORM может стать узким местом – лучше возвращать DTO

# Примеры

# Примеры

- Camunda

Describes basic methods for querying.

Author: Frederik Heremans

```
public interface Query<T extends Query< ? , ? >, U extends Object> {
```

```
public abstract class AbstractModificationCmd <T> implements Command<T> {
```



# Примеры

- Простое REST API

## Catalog

- GET /catalog/products/{sku}
- PUT /catalog/products/{sku}

## Sales

- GET /sales/products/{sku}
- POST /sales/products/{sku}/increasePrice
- POST /sales/products/{sku}/decreasePrice
- POST /sales/products/{sku}/available
- POST /sales/products/{sku}/unavailable

## Purchasing

- POST /purchasing/products/{sku}/increaseCost
- POST /purchasing/products/{sku}/decreaseCost

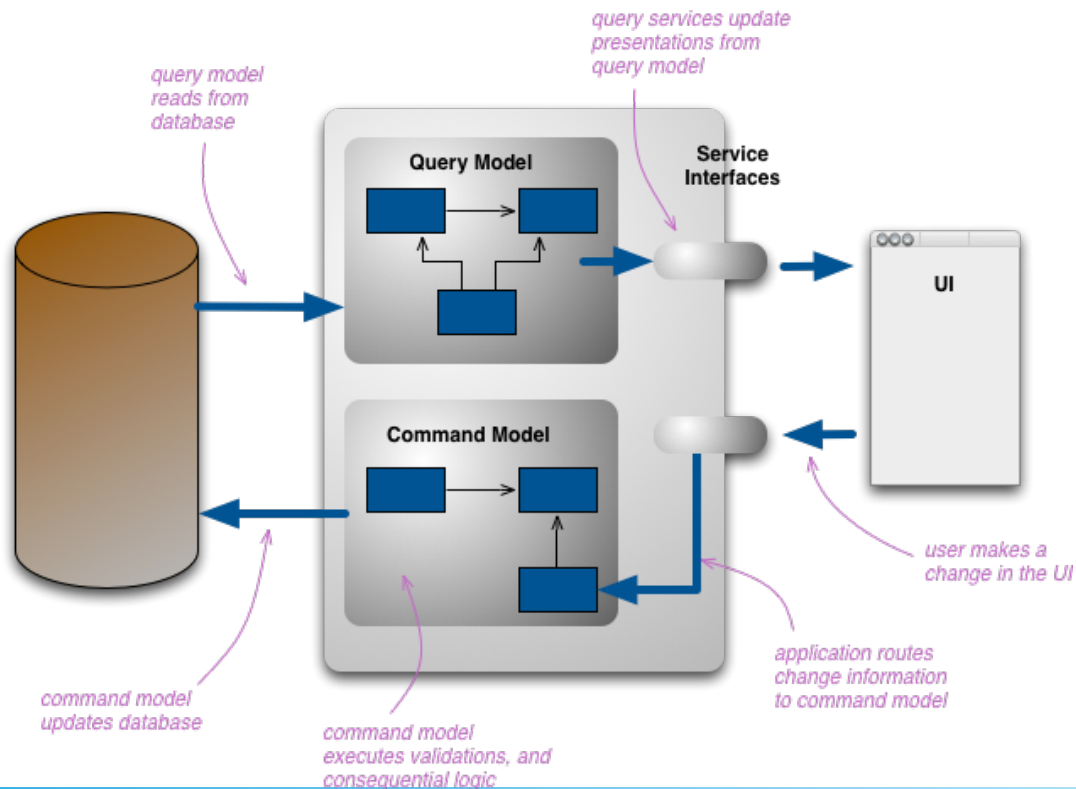
## Warehouse

- GET /warehouse/products/{sku}
- POST /warehouse/products/{sku}/inventoryAdjustment
- POST /warehouse/products/{sku}/shipProduct
- POST /warehouse/products/{sku}/receiveProduct

ИСТОЧНИК

# Примеры

- Взаимодействие с UI



ИСТОЧНИК

# Примеры

- Сбор данных для поиска/аналитики – много записи, мало чтения (мой диплом, гыгы)



## Greg Young's Blog

Good Enough Software is By Definition Good Enough

[Class Videos](#)   [Where am I?](#)

[« StackOverflow](#)

[Free Phone This](#)

### **CQRS**

CQRS is not a silver bullet

CQRS is not a top level architecture

CQRS is not new

CQRS is not shiny

CQRS will not make your jump shot any better

CQRS is not intrinsically linked to DDD

CQRS is not Event Sourcing

CQRS does not require a message bus

CQRS is not a guiding principle / CQS is

CQRS is not a good wife

CQRS is learnable in 5 minutes

CQRS is a small tactical pattern

CQRS can open many doors.

And yes you can support [RFC 2549](#) using CQRS

Спасибо за внимание!