

# Modelling complex business domains with events

*Ben Smith*

# Overview

1. Introduction to domain events
2. Modelling complex business processes with events
3. Implementing our model in Elixir

What are domain  
events?

# Domain event

- Record of something that took place in a business domain
- An immutable, historical *fact*
- Named in the past tense using the language of the business
- Examples:
  - Item Added To Cart
  - Checkout Completed
  - Payment Taken
  - Order Dispatched

# Event-driven systems

- Event notification
  - Publish *notification* domain events to other systems
- Event-carried state transfer
  - Publish *enriched* domain events containing details of any changes
- Event sourcing
  - Record all changes to a system as *internal* domain events
  - Current state is (re)built from events

See Martin Fowler's [What do you mean by “Event-Driven”?](#)

How do I model  
with events?

# EVENT STORMING

External  
System

Command

Domain  
Event

Actor

Aggregate

Policy

Read  
Model



## 1. Big picture

- Knowledge or learning is the most valuable outcome
- Identify bounded contexts and/or service boundaries

## 2. Design level

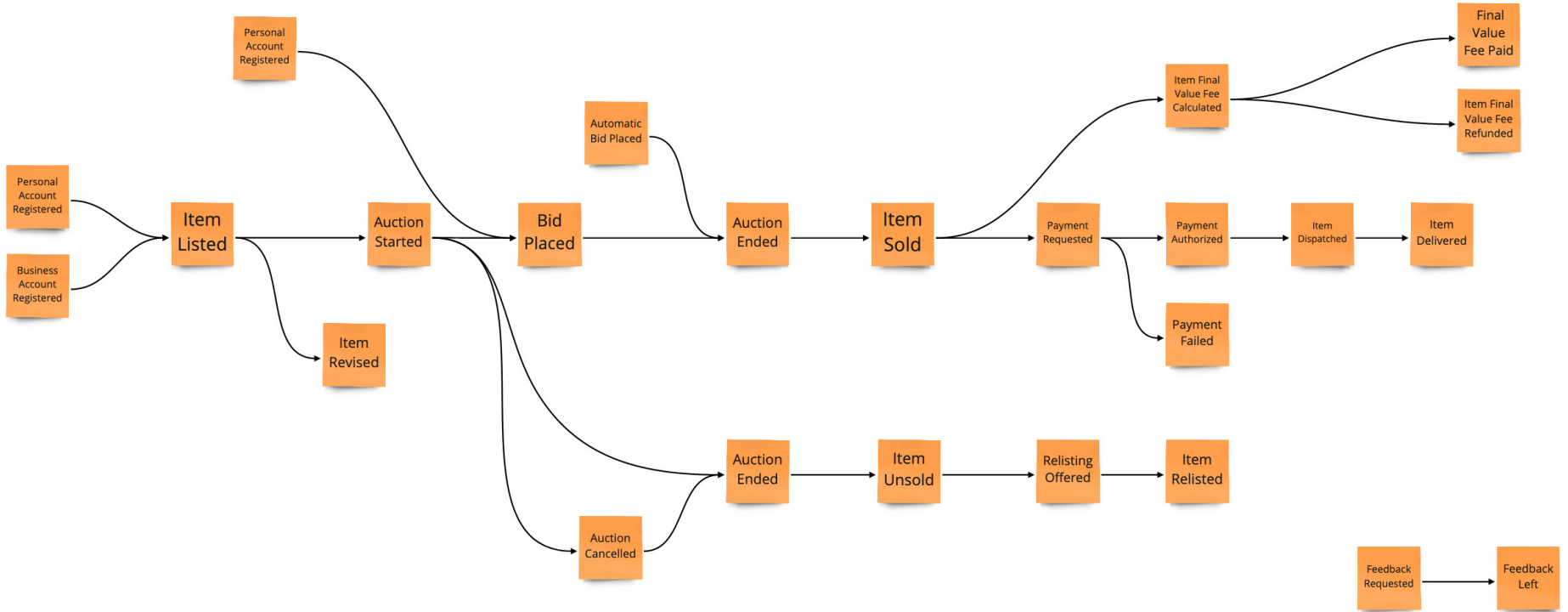
- Design a solution to a specific business problem
- Desired outcome is a robust candidate solution

[www.eventstorming.com](http://www.eventstorming.com)



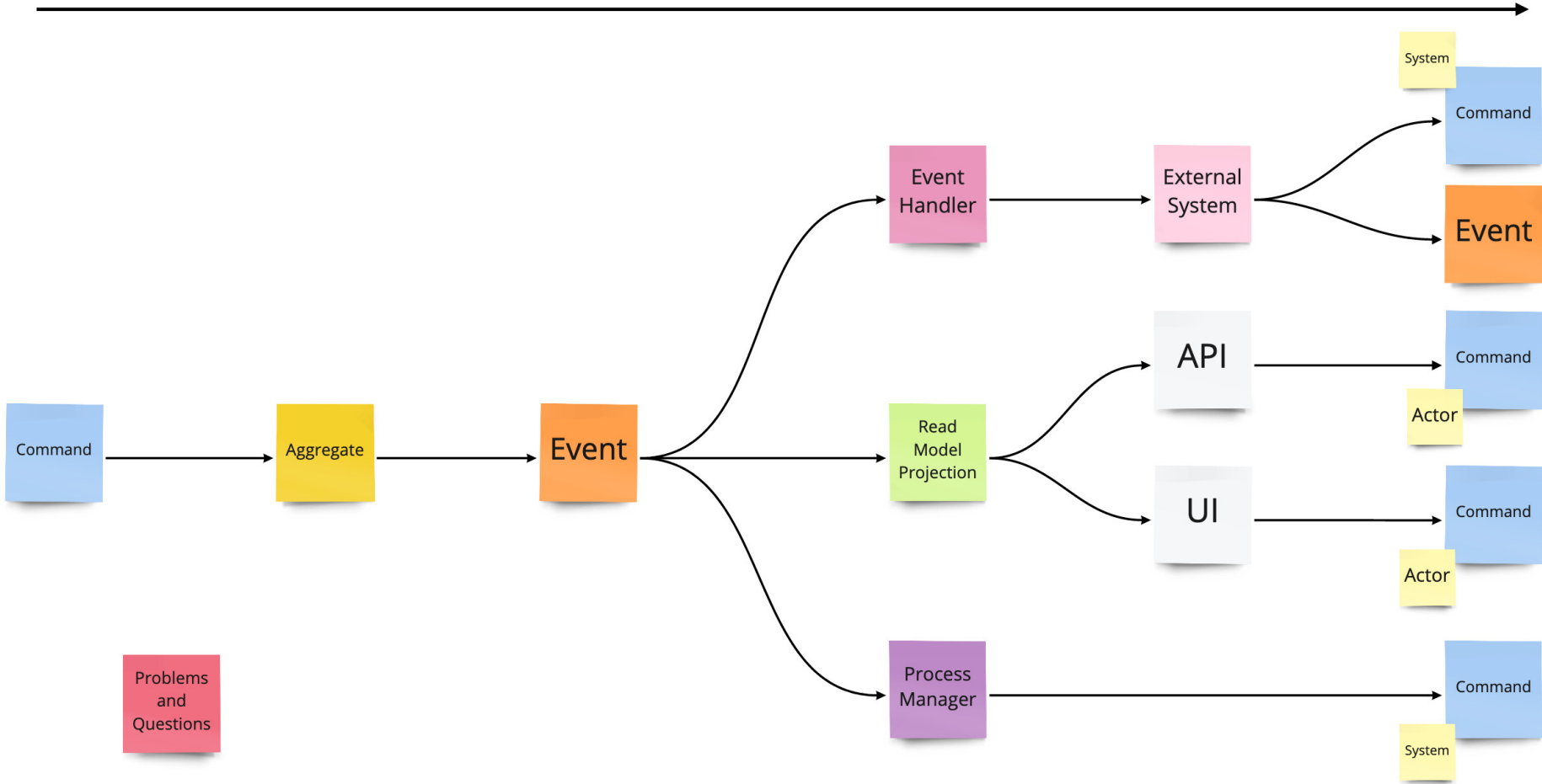
# Big picture EventStorming

Time



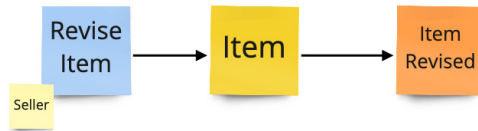
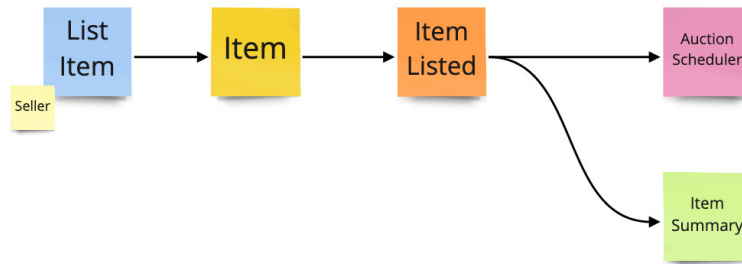
# Design level EventStorming

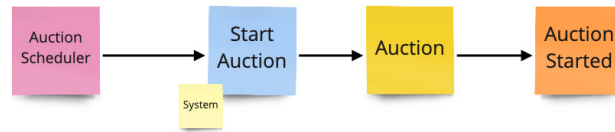
Time

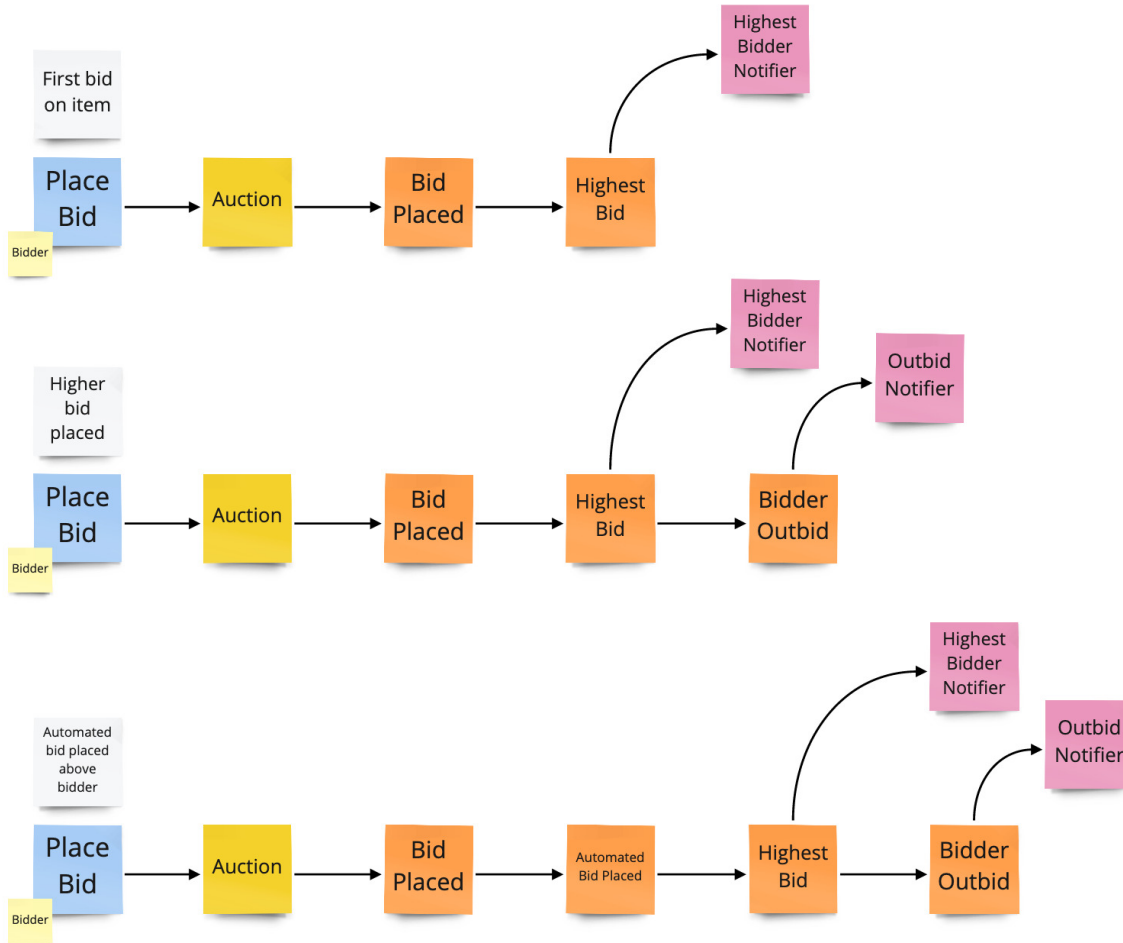


Problems and Questions

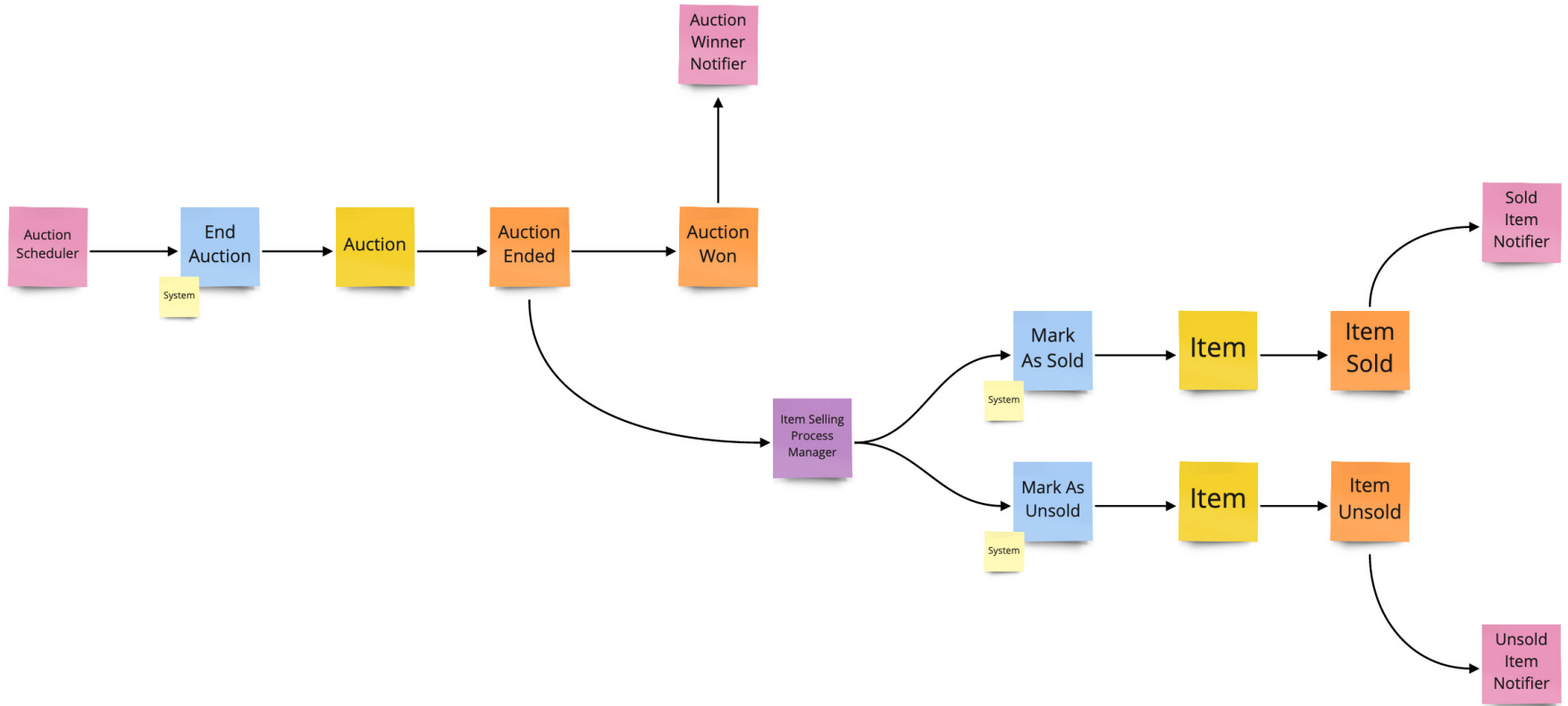
# Modelling an online auction system

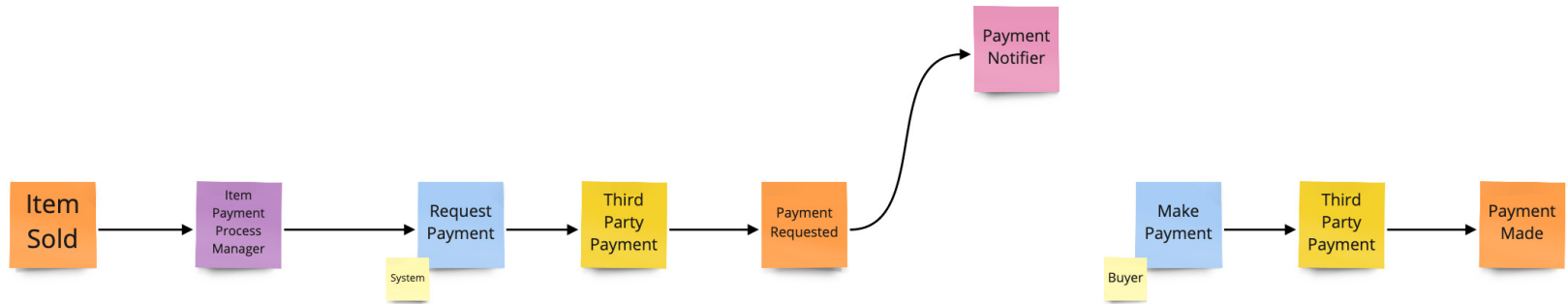


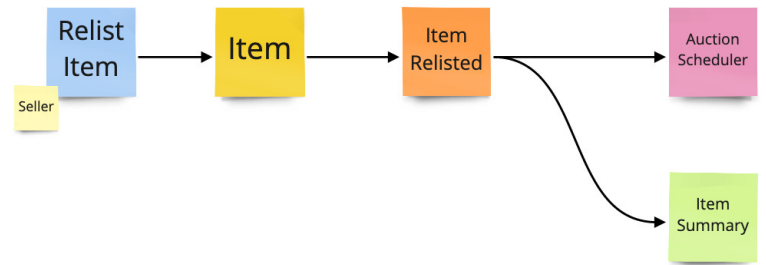
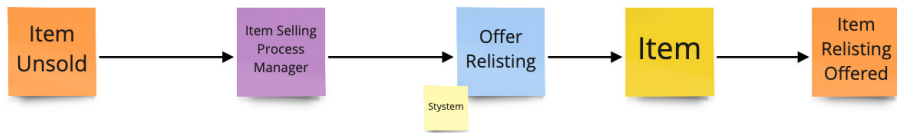


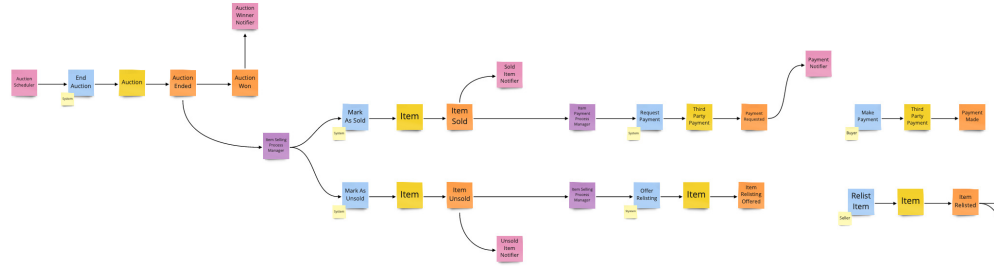
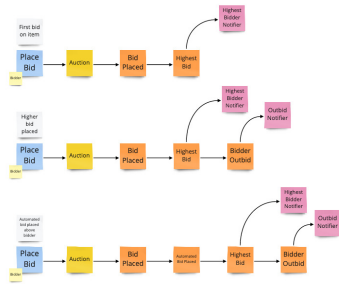
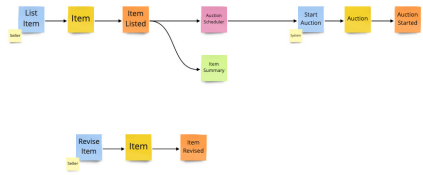












How do I  
implement the  
model in Elixir?

Commanded

# Commanded

Use Commanded to build Elixir CQRS/ES applications

📍 UK [🔗 http://commanded.io/](http://commanded.io/)

 **Repositories** 13  Packages  People 1  Teams 1  Settings

 **commanded** ⋮

Use Commanded to build Elixir CQRS/ES applications

● Elixir ☆ 1.3k 🍴 177

 **eventstore** ⋮

Event store using PostgreSQL for persistence

● Elixir ☆ 759 🍴 105

 **commanded-ecto-projections** ⋮

Read model projections for Commanded using Ecto

● Elixir ☆ 60 🍴 24

 **commanded-audit-middleware** ⋮

Command auditing middleware for Commanded CQRS/ES applications

● Elixir ☆ 26 🍴 12

 **recipes** ⋮

Commanded recipes

☆ 6

 **generator** ⋮

Commanded application generator

● Elixir

 [github.com/commanded](https://github.com/commanded)

# Commanded generator

*Mix task to create a new Commanded project.*

```
mix commanded.new auctioneer --miro o9J_1Dnyq6U=
```

 [github.com/commanded/generator](https://github.com/commanded/generator)

```
commanded-generator -- -bash -- 119x54
~/src/commanded/commanded-generator -- -bash
Bens-MacBook-Pro:commanded-generator ben$ mix commanded.new /tmp/auctioneer --miro o9J_lDnyq6U=
* creating /tmp/auctioneer/config/config.exs
* creating /tmp/auctioneer/config/dev.exs
* creating /tmp/auctioneer/config/prod.exs
* creating /tmp/auctioneer/config/runtime.exs
* creating /tmp/auctioneer/config/test.exs
* creating /tmp/auctioneer/lib/auctioneer/app.ex
* creating /tmp/auctioneer/lib/auctioneer/application.ex
* creating /tmp/auctioneer/lib/auctioneer/event_store.ex
* creating /tmp/auctioneer/lib/auctioneer/repo.ex
* creating /tmp/auctioneer/lib/auctioneer/router.ex
* creating /tmp/auctioneer/lib/auctioneer.ex
* creating /tmp/auctioneer/test/test_helper.exs
* creating /tmp/auctioneer/.formatter.exs
* creating /tmp/auctioneer/mix.exs
* creating /tmp/auctioneer/README.md
* creating /tmp/auctioneer/lib/auctioneer/auction/auction.ex
* creating /tmp/auctioneer/lib/auctioneer/auction/commands/end_auction.ex
* creating /tmp/auctioneer/lib/auctioneer/auction/commands/place_bid.ex
* creating /tmp/auctioneer/lib/auctioneer/auction/commands/start_auction.ex
* creating /tmp/auctioneer/lib/auctioneer/auction/events/auction_ended.ex
* creating /tmp/auctioneer/lib/auctioneer/auction/events/auction_started.ex
* creating /tmp/auctioneer/lib/auctioneer/auction/events/auction_won.ex
* creating /tmp/auctioneer/lib/auctioneer/auction/events/automated_bid_placed.ex
* creating /tmp/auctioneer/lib/auctioneer/auction/events/bid_placed.ex
* creating /tmp/auctioneer/lib/auctioneer/auction/events/bidder_outbid.ex
* creating /tmp/auctioneer/lib/auctioneer/auction/events/highest_bid.ex
* creating /tmp/auctioneer/lib/auctioneer/item/item.ex
* creating /tmp/auctioneer/lib/auctioneer/item/commands/list_item.ex
* creating /tmp/auctioneer/lib/auctioneer/item/commands/mark_as_sold.ex
* creating /tmp/auctioneer/lib/auctioneer/item/commands/mark_as_unsold.ex
* creating /tmp/auctioneer/lib/auctioneer/item/commands/relist_item.ex
* creating /tmp/auctioneer/lib/auctioneer/item/commands/revise_item.ex
* creating /tmp/auctioneer/lib/auctioneer/item/events/item_listed.ex
* creating /tmp/auctioneer/lib/auctioneer/item/events/item_relisted.ex
* creating /tmp/auctioneer/lib/auctioneer/item/events/item_revised.ex
* creating /tmp/auctioneer/lib/auctioneer/item/events/item_sold.ex
* creating /tmp/auctioneer/lib/auctioneer/item/events/item_unsold.ex
* creating /tmp/auctioneer/lib/auctioneer/third_party_payment/third_party_payment.ex
* creating /tmp/auctioneer/lib/auctioneer/third_party_payment/commands/make_payment.ex
* creating /tmp/auctioneer/lib/auctioneer/third_party_payment/commands/request_payment.ex
* creating /tmp/auctioneer/lib/auctioneer/third_party_payment/events/payment_made.ex
* creating /tmp/auctioneer/lib/auctioneer/third_party_payment/events/payment_requested.ex
* creating /tmp/auctioneer/lib/auctioneer/handlers/auction_scheduler.ex
* creating /tmp/auctioneer/lib/auctioneer/handlers/auction_winner_notifier.ex
* creating /tmp/auctioneer/lib/auctioneer/handlers/highest_bidder_notifier.ex
* creating /tmp/auctioneer/lib/auctioneer/handlers/outbid_notifier.ex
* creating /tmp/auctioneer/lib/auctioneer/handlers/payment_notifier.ex
* creating /tmp/auctioneer/lib/auctioneer/handlers/sold_item_notifier.ex
* creating /tmp/auctioneer/lib/auctioneer/handlers/unsold_item_notifier.ex
* creating /tmp/auctioneer/lib/auctioneer/processes/item_payment_process_manager.ex
* creating /tmp/auctioneer/lib/auctioneer/processes/item_selling_process_manager.ex
* creating /tmp/auctioneer/lib/auctioneer/projections/item_summary_projector.ex
```



# Awesome Elixir and CQRS

---

A curated list of awesome Elixir, Command Query Responsibility Segregation (CQRS), and event sourcing resources.

## Contents

- [Libraries](#)
- [Community resources](#)
- [Books](#)
- [Articles](#)
- [Presentations](#)
- [Screencasts](#)
- [Example applications](#)

# Get in touch

- [ben@10consulting.com](mailto:ben@10consulting.com)
- [10consulting.com](http://10consulting.com)