

# Getting started with event-sourcing

...

And a little bit about tracking your customers  
Tim Huijzers



Tim Huijzers  
Dragem or Webbaard  
Developer @  
[island-bookings.com](https://island-bookings.com)  
Founder of DeventerPHP  
Usergroup



# First Try

- Limited knowledge
- No experience
- No ES framework
- Doomed from the start

# Second Try

- Limited knowledge
- Limited experience
- New Framework
- New DI manager
- New ES Framework
- Doomed To Fail

# Third Try

- Some knowledge
- Some experience
- Known framework
- known database
- known ES framework
- Still doomed



I KNOW WHAT I'M DOING!







**BeerWarehouse**

# Why use Event Sourcing

**CRUD**

Beer
Brewer: Founders
Name: KBS
Bought: 2018-03-12
Location: Fridge
Style: Imperial Russian Stout

We will save a new entry in our system because we just bought it and will store it in the fridge for later.

Beer
Brewer: Founders
Name: KBS
Bought: 2018-03-12
Location: Shelf
Style: Imperial Russian Stout

If we change the location the system only knows about that location.

Beer
Brewer:
Name:
Bought:
Location:
Style:

We drank it so it's not in the system anymore

Beer
Brewer: Founders
Name: KBS
Bought: 2018-03-12
Location: All Gone
Style: Imperial Russian Stout

We want to keep a history of everything we drank.



## Beer

Brewer: Founders

Name: KBS

Bought: 2018-03-12

Location: All Gone

Style: Imperial Russian Stout

ConsumptionDate: 2018-05-03

I want to know when I drank this in my history.

**But that's only for new beers.**

A man in a dark suit, white shirt, and patterned tie is shown from the chest up, looking back over his right shoulder with a serious, intense expression. He is in an office setting with large windows in the background showing a blurred cityscape. Another person's back is visible on the right side of the frame.

THAT'S NOT  
GOOD ENOUGH

# Events

BeerAddedToStorage
Brewer: Founders
Beer: KBS
Bought: 2018-03-12
Location: Fridge
Style: Imperial Russian Stout

**Same Information as before + Explicit action about what happened**

**Make Small Events**

BeerBought
Brewer: Founders
Beer: KBS
Bought: 2018-03-12
Style: Imperial Russian Stout

Removed Location and changed name because in the real world you might not know this yet.



BeerMoved
Identifier: BeerId
Location: Fridge

When returning home I put the beer in my fridge

BeerMoved
Identifier: BeerId
Location: Shelf

I need room in my fridge so I take it out. Using the same  
Event

BeerConsumed
Identifier: BeerId

And at last a event about when I consumed it.

CANAL+ FILM HD



# Crud

- I know what beer I have.
- I know when it was consumed.
- I know where it is.

# Event-Sourcing

- I know what beer I have.
- I know when it was consumed.
- I know where it is.
- I know where it was before.
- I know when it was moved.
- I know where it was at any point in time
- I know how many times it was moved.
- I know when it was added to the system.
- I know what else was moved in that day.

**“Every software program relates to some activity or interest of its user.”**

**Eric Evans, Domain-Driven Design: Tackling Complexity in the Heart of Software**





# When To Use Event Sourcing

- You need an audit log
- You like scalability
- You want to separate the read and write of an application
- You want to replay event on a dev machine to get an accurate situation of what the state was at a point in time.
- You want reporting but don't know what yet.
- You are done with mapping objects to tables

# When NOT To Use Event Sourcing

- You only need a simple CRUD system.
- You are processing a lot of personal data.
- You just want to query a lot of things on the DB
- You are starting on a big project for production

# Event Sourcing in code

Prooph


<http://getprooph.org/>



# THE CQRS AND EVENT SOURCING COMPONENTS FOR PHP

Crafted for Your Enterprise App

INSTALL

TAKE A TOUR 



# Let's build a Application

...

<https://github.com/webbaard/FoxAndHound>

A man with a mustache and dark hair, wearing a light blue suit jacket, a light-colored shirt, and a tie, is shown from the chest up. He is looking off to the left with a slight, knowing smirk. The background is an office setting with a window on the left, a small boat hanging on the wall behind him, and a bookshelf on the right. The text "GREAT SUCCESS!" is overlaid at the bottom in a large, bold, white font with a black outline.

**GREAT SUCCESS!**

# Pitfalls

**Refactoring is harder, think about your  
architecture**



OPEN

Open  
Mon  
Tue-Thur  
Fri-Sat  
Sunday

# Versioning

- Change an Event but support the old version
- Make a new Event
- Make the Event right from the start

**Something wrong with the event**

# Event are immutable, So don't change them

- Try solving it another way first.
- Correct errors with new events
- Try a upcaster
- Make a new stream and fill it with mutated events (and test)
- Change the events in the database



**But what if I have like 100 trillion gazillion  
events?**

**LEBRON BE LIKE..**

A photograph of LeBron James in a red Miami Heat jersey with the number 6. He is standing on a basketball court with his arms outstretched, palms facing forward, in a gesture of calm or protest. The background is a blurred crowd of spectators in a stadium.

**CALM THAT SH\*T DOWN!**

# Snapshots

**You Do Not Need Snapshots From The start**

# Trigger on Event Count

**Pure Event Sourcing Is Not A Holy Grail**

**Do Not Save Personal Data In Events**

**Make Projections For All Your Lists**



# Try It In A Hackathon First

...

Or a nice hackathon

**Most Of The Time Your DB Is Not Holy**

**What Now?**

<http://getprooph.org/>



# THE CQRS AND EVENT SOURCING COMPONENTS FOR PHP

Crafted for Your Enterprise App

INSTALL

TAKE A TOUR 

# Source

<https://github.com/prooph/proophessor-do-symfony>

<http://getprooph.org/>

# Other Tools

- Broadway
  - No Upcaster,
  - No Snapshots,
  - No Replaying
- Axon
  - Upcasting by MessageFactory,
  - Snapshots by Trigger on event count,
  - Replaying by Example code for replay
- Akka
  - Upcasting by Event Adapter,
  - Snapshots decided by actor,
  - Replaying

**Thank you.**