

Practical Event Sourcing

Alfredo Motta, @mottalrd

https://github.com/mottalrd/event_sourcing_workshop

Ruby Unconference, London 6th October 2018

Agenda

- 10 minutes intro to Event sourcing
- Build a blog in 1 hour using event sourcing

Step 1: Setup Rails app

- As a user
 - Given that I can visit the home page of the blog
 - And that I click on the new article link
 - I expect to see a page with two text fields (article title, article text)

Step 2: Create entities

- As a user
 - Given that I visit the new articles page
 - And I provide title and text
 - And I hit submit
 - I expect to be redirected to the newly created article page (but only see dummy text there)
- As a developer
 - Given that the user submitted the new article
 - I expect a new event <entity_type: article, event_type: created> to be added to the database

Step 3: Validation

- As a user
 - Given that I visit the new articles page
 - And I DO NOT provide title and text
 - And I hit submit
 - I expect NOT TO be redirected to the newly created article
 - I expect to see some errors on the page
- As a developer
 - Given that the user submitted an invalid article
 - I DO NOT expect to see any new event in my events table

Step 4: Show entity

- As a user
 - Given that I visit the new articles page
 - And I provide title and text
 - And I hit submit
 - I expect to be redirected to the newly created article with the appropriate text
- As a developer
 - When the user visits the show page of a given article
 - I expect to retrieve such article using the events in my table (re-hydrate my entity)

Step 5: Build relation views

- As a user
 - Given that I visit the home page
 - And I click the “All articles” link
 - I expect to see all the articles created so far

Step 6: Edit a blog post

- As a user
 - Given that I visit the home page
 - And I click the “All articles” link
 - I expect to see an Edit link for each article
 - Given that I click such link on the first article
 - I expect to see a page with the article edit form
 - Given that there is a checkbox to mark an article as published
 - And I tick the checkbox, plus submit
 - I expect to be redirected to the show page of such article
 - I expect to see the article marked as published

Step 7: Send an email when an article is published

- As a user
- Given that I go to the edit article page
- And I edit such article with some new text
- I expect to receive an email to event_sourcing_workshop@mailinator.com confirming my operation

Step 8: Add a report

- As a business analyst
- Given that I want to keep the count of all the articles published
- I expect to be able to have an sql table updated with the latest published articles count

Step 9: Add comments

- As a user
- Given that I go to the page of an article
- I expect to see a comments text area
- Given that I provide my comment and my name
- And I hit submit
- I expect to see my comment in the list of comments for such page



That's all Folks!