

# Toys on Trains

A Ruby on Rails eCommerce Website

---

Peter Murphy

January 25, 2018

Center for modern beamer themes

# Table of contents

1. Introduction
2. Implementation
3. Elements

# Introduction

---

# The Job Test

- Demonstrate the ability to make a website in *Ruby on Rails*
  - A toy store for selling and buying toys
- Demonstrate the ability to design websites as well as build them
  - What do the users want?
  - Can the problem be broken down to design a solution.
- Can the code for thee website be stored at Github?
- Can the website be implemented live?

# Deliverables

- Github project (and README) at:
  - <https://github.com/peterkmurphy/toytracks>
- Design documentation at:
  - <https://github.com/peterkmurphy/toytracks/blob/master/design.md>
  - Includes Project Summary
  - Includes User Stories
  - Includes Entity Relationship Modelling
  - Includes *some* wireframes
- Implemented website at:
  - <https://limitless-escarpment-96871.herokuapp.com/>
  - Runs on Heroku

# Project Summary

- **Toys on Tracks**
- Website for buying and selling toys
- Different registration unnecessary for both roles.
- Assumptions:
  - Each sale involves an individual item rather than multiple items
  - Each product is unique (users don't sell from a line of toys)
- Two types of transactions
  - One off fixed sales
  - Auctions sales at start price, users start bidding, until fixed period expires.
- Transactions simple: sellers have bank accounts, buyers have credit cards, money works or not.
- Doesn't handle delivery, fraud, bitcoin, etc.

# The Home Page (Wireframe)



**Figure 1:** This is a wireframe of the home page.

# The Profile Page (Wireframe)

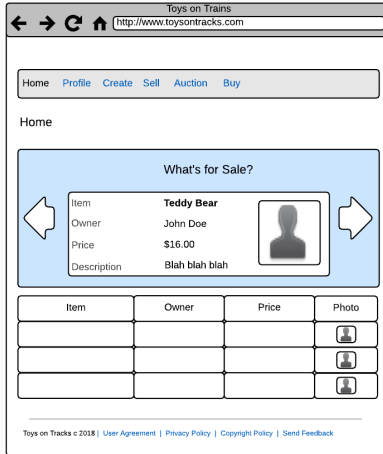


Figure 2: This is a wireframe of the profile page.



# Implementation

---

# Rolling out the website

- Tried to use a website generator to generate an example of a "working" website
- That is, with Devise and other goodies installed.
- Then make the final copy somewhere else.
- Ended up using it as base for today's submissions.
- Rails Composer broken
- Any website involving Devise had errors
- Used prelang - abandonware from a year ago that got a Rails 4.x website

**Let's see the website at**

`https://limitless-escarpment-96871.  
herokuapp.com/.`

# What was implemented

- A Ruby on Rails
- Uses Devise for Authentication
- Can be deployed to the cloud with minimum effort

# What was NOT implemented

- Code quality tools like Rspec
- APIs like Omniauth or Geocoding
- Allowing photos to be uploaded
  - Could have been implemented easily in Development
  - Need S3 to be implemented for Heroku
- Way of hiding the secret stuff in the YAML files.
- A website that came close to implementing the user stories.

## Conclusion

---

# Comparisons with Django

- Different terminology - same idea
  - What is a "view" in Rails is a "template" in Django
  - What is a "controller" in Rails is a "view" in Django
  - Both share same concept of "models" and "URL routers"
- Easy to get started in Ruby on Rails than in Django
- Easy to find routes in Ruby on Rails than in Django
- Django has authentication built in (RoR needs Devise)
- Django's model migrations are far easier in 2018.

**Questions?**



# About this presentation

This presentation was developed using *Beamer*.

`github.com/josephwright/beamer`

With the aid of the *Metropolis* theme.

`github.com/matze/mtheme`

The theme *itself* is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

