

ChartJS

Devpoint Labs - Dave Jungst / Jake Sorce

Create Rails app

```
rails new robots -T -d postgresql --skip-turbolinks
```

```
rails g model inventor name age:integer hire_date:date
```

```
rails g model robot name industry last_service:date evil:boolean durability:integer color inventor:belongs_to
```

```
bundle exec rake db:create db:migrate
```

Gemfile

```
gem 'faker'
```

```
gem 'populator'
```

```
gem 'haml-rails'
```

```
gem 'materialize-sass'
```

```
gem 'chart-js-rails'
```

```
gem 'momentjs-rails'
```

```
bundle
```

```
mv app/assets/stylesheets/application.css app/assets/stylesheets/application.css.scss
```

```
@import 'materialize';
```

```
app/assets/javascripts/application.js
```

```
//= require moment
```

```
//= require Chart
```

```
rails g controller dashboard index
```

```
app/models/inventor.rb
```

```
has_many :robots, dependent: :destroy
```

```
root 'dashboard#index'
```

app/views/dashboard/index.html.haml

Use a canvas as a placeholder for the chart. The canvas should have an ID and since the Canvas doesn't know it's size prior to being populated it's a good idea to give it a default size.

```
1 .container
2   %h2.center-align Compton Robot Bunnies
3   %hr
4   .row
5     .col.s4
6       %canvas#inventors_hired{height: 400}
```

Populate some data

```
1 #lib/tasks/populate.rake
2 namespace :populate do
3   require 'populator'
4   require 'faker'
5
6   desc "Create a bunch of records"
7   task data: :environment do
8     Inventor.populate 50 do |inventor|
9       inventor.name = Faker::Name.name
10      inventor.age = Faker::Number.between(18, 110)
11      inventor.hire_date = Faker::Date.between(10.years.ago, Date.today)
12      Robot.populate Faker::Number.between(1, 10) do |robot|
13        robot.name = Faker::Name.name
14        robot.industry = ['Service', 'War', 'Technician', 'Education', 'Science'].sample
15        robot.last_service = Faker::Number.between(Date.today.beginning_of_year, Date.today)
16        robot.evil = [true, false].sample
17        robot.durability = Faker::Number.between(1, 10)
18        robot.color = ['Gray', 'Black', 'Blue', 'Green', 'White', 'Red'].sample
19        robot.inventor_id = inventor.id
20      end
21    end
22  end
23 end
```

bundle exec rake populate:data

Build a chart with static data

1. Check the path so this code only runs on the page desired.
2. Set up a data hash consisting of labels and datasets
3. labels are the chart labels and datasets are hashes containing options for the data
4. `datasets[:data]` is an array of data points.

```
1 $(document).ready ->
2   if window.location.pathname == '/'
3     data = {
4       labels: ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sept", "Oct"]
5       datasets: [
6         {
7           label: "My First dataset",
8           fillColor: "rgba(220,220,220,0.2)",
9           strokeColor: "rgba(220,220,220,1)",
10          pointColor: "rgba(220,220,220,1)",
11          pointStrokeColor: "#fff",
12          pointHighlightFill: "#fff",
13          pointHighlightStroke: "rgba(220,220,220,1)",
14          data: [65, 59, 80, 81, 56, 55, 40]
15        }
16      ]
17    }
18
19    $('#inventors_hired').empty()
20    new Chart($('#inventors_hired').get(0).getContext('2d')).Line(data, {responsive: true})
```

Line 19 clears out the canvas

Line 20 creates a new chart by finding the chart with JQuery then getting the context finally setting the type of chart in this case Line and passing in the data followed by any chart options

Chart Types

Line: A line chart is used to show data trends and can be used to compare 2 or more datasets.

Bar: Show datasets side by side.

Radar: Show multiple datasets and the variation between them.

Polar Area: Similar to pie chart but each dataset has the same angle.

Pie / Doughnut: Used to compared proportion from datasets

Common Options

scaleShowGridLines: (true) Show grid lines on chart

scaleGridLineColor: (rgba(0,0,0,0.05)) Color of grid lines

scaleGridLineWidth: (1) Width of grid lines

scaleShowHorizontalLines: (true) Show horizontal grid lines

scaleShowVerticalLines: (true) Show vertical grid lines

bezierCurve: (true) Curved line between points

bezierCurveTension: (0.4) Tension of curve

pointDot: (true) Show dots for data points

pointDotRadius: (true) Radius of each point dot

dataFill: (true) Fill the dataset with a color

legendTemplate: A legend template as HTML

Setting Options

```
1 $(document).ready ->
2   if window.location.pathname == '/'
3     data = {
4       labels: ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sept", "Oct"]
5       datasets: [
6         {
7           label: "My First dataset",
8           fillColor: "rgba(220,220,220,0.2)",
9           strokeColor: "rgba(220,220,220,1)",
10          pointColor: "rgba(220,220,220,1)",
11          pointStrokeColor: "#fff",
12          pointHighlightFill: "#fff",
13          pointHighlightStroke: "rgba(220,220,220,1)",
14          data: [65, 59, 80, 81, 56, 55, 40]
15        }
16      ]
17    }
18
19    options = { responsive: true, scaleShowGridLines: false, pointDotRadius: 3, bezierCurve: false }
20
21    $('#inventors_hired').empty()
22    new Chart($('#inventors_hired').get(0).getContext('2d')).Line(data, options)
23
```

Dynamic Data

```
1 $(document).ready ->
2   if window.location.pathname == '/'
3     labels = []
4     dataPoints = []
5     $.ajax '/inventors_hired',
6       type: 'GET'
7     success: (data) ->
8       labels = Object.keys(data.records)
9       for key of data.records
10         dataPoints.push(data.records[key])
11       data = {
12         labels: labels
13         datasets: [
14           {
15             label: "My First dataset",
16             fillColor: "rgba(220,220,220,0.2)",
17             strokeColor: "rgba(220,220,220,1)",
18             pointColor: "rgba(220,220,220,1)",
19             pointStrokeColor: "#fff",
20             pointHighlightFill: "#fff",
21             pointHighlightStroke: "rgba(220,220,220,1)",
22             data: dataPoints
23           }
24         ]
25       }
26
27       options = { responsive: true, scaleShowGridLines: false, pointDotRadius: 3, bezierCurve: false }
28
29       $('#inventors_hired').empty()
30       new Chart($('#inventors_hired').get(0).getContext('2d')).Line(data, options)
```

Controller / Route

rails g controller inventors

get 'inventors_hired', to: 'inventors#hired'

```
1 class InventorsController < ApplicationController
2   def hired
3     render json: { records: Inventor.hired }
4   end
5 end
```

The query

```
1 class Inventor < ActiveRecord::Base
2   def self.hired
3     months = {}
4     counts = select("count(*) AS count, date_trunc( 'month', hire_date) AS hire_date")
5               .where("EXTRACT(year FROM hire_date) = 2014")
6               .group("date_trunc( 'month', hire_date)")
7               .order("date_trunc('month', hire_date)")
8     counts.each do |record|
9       months[record.hire_date.strftime("%b")] = record.count
10    end
11    months
12  end
13 end
```

The query returns a count by month of inventors hired

And then a hash is set to the month as the key and count as value { "Jan" => 3 }

Pie / Doughnut / Polar Area

These type of charts are set up a little differently

```
32 industryData = []
33 $.ajax '/by_industry',
34   type: 'GET'
35   success: (data) ->
36     for type, count of data.robots
37       switch type
38         when 'Service' then set = {value: count, color: "#517EA6", highlight: 'rgba(81,126,166,.1)', label: type}
39         when 'War' then set = {value: count, color: "#AB2116", highlight: 'rgba(171,33,22,.1)', label: type}
40         when 'Technician' then set = {value: count, color: "#5A6B75", highlight: 'rgba(90,107,117,.1)', label: type}
41         when 'Education' then set = {value: count, color: "#BC774C", highlight: 'rgba(188,119,76,.1)', label: type}
42         when 'Science' then set = {value: count, color: "#FDB45C", highlight: 'rgba(253,180,92,.1)', label: type}
43       industryData.push(set)
44
45   $('#robots_by_industry').empty()
46   new Chart($("#robots_by_industry").get(0).getContext("2d")).PolarArea(industryData)
```

Route & View

```
1 Rails.application.routes.draw do
2   root 'dashboard#index'
3
4   get 'inventors_hired', to: 'inventors#hired'
5   get 'by_industry', to: 'robots#by_industry'
6 end
```

```
1 .container
2   %h2.center-align Compton Robot Bunnies
3   %hr
4   .row
5     .col.s4
6       %canvas#inventors_hired{height: 400}
7     .col.s4
8       %canvas#robots_by_industry{height: 400}
```

Controller & Model

rails g controller robots

```
1 class RobotsController < ApplicationController
2   def by_industry
3     render json: { robots: Robot.by_industry }
4   end
5 end
```

```
1 class Robot < ActiveRecord::Base
2   belongs_to :inventor
3
4   def self.by_industry
5     robots = {}
6     results = select('count(*) AS count, industry')
7               .group(:industry)
8     results.each do |r|
9       robots[r[:industry]] = r[:count]
10    end
11    robots
12  end
13 end
```

Project

1. On the Dashboard create a chart that shows a count of robots by color
2. Create an Inventor index route and some way to get there from the Dashboard
 - a. # of robots by inventor
 - b. # robot counts by industry for most recently hired inventor
3. Create an inventor show page
 - a. % of Good VS Evil robots for Inventor
4. Create a Robot Index Page
 - a. # robots serviced by month this year

BONUS:

- Avg durability by color
- Avg durability by industry
- Color counts by industry