# Инициализация приложения

1. Версия руби 2.0
2. Установить базу данных Mysql, возможно потребуется создать тестовую базу для

DROP DATABASE IF EXISTS railstest\_development;  
CREATE DATABASE railstest\_development CHARACTER SET utf8;  
USE railstest\_development;  
set charset Utf8;

1. Сконфигурировать настройки базы для доступа в файле config/database.yml проекта
2. Нужно собрать gem который используется и реализован в рамках проекта, для этого нужно зайти в папку graph\_form\_sql и выполнить команды по установки гема:

gem build graph\_form\_sql.gemspec  
 gem push graph\_form\_sql-0.0.1.gem  
  
4) перейти в корневую директорию проекта и запустить bundle install

для инсталляции всех нужных гемов в проекте

5) запустить 'rake db:migrate' and 'rake db:seed' для инсталляции структуры базы и заполнения таблиц данными

6) запустить сервер 'rails server' command  
  
7) перейти на стартовую страницу localhost:3000

# Что было сделано

1. Сделан механизм для заполнения базы данных через веб формы CRUD <http://localhost:3000/authors>

Можно заполнять авторов и добавлять к ним книги

1. Автоматический механизм заполнения базы seed
2. Отображения данных на графиках 3мя способами, можно посмотреть в controllers\graph\_controller.rb

- первый способ использует простой find\_by\_sql и итерацию на вьюхе

- второй способ формирует все строковые данные внутри класса ChartGraph и выдает на вьюху. Которая использует хелпер для отображения по типу <%= make\_graph @booksbyyear , 'canvas1' , "Line" %>

- третий способ такой же как второй, но с использованием Gema отдельного

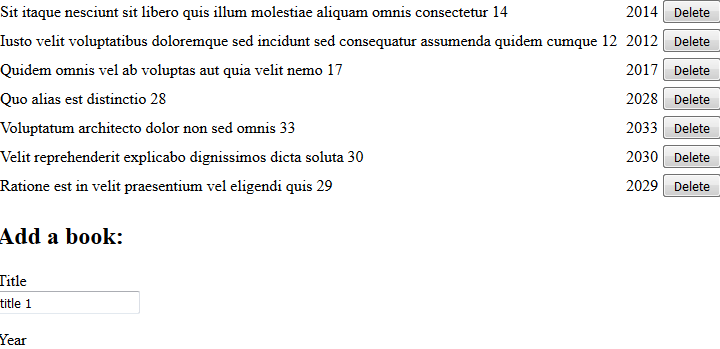
1. Сделан гем **graph\_form\_sql** для выборки данных с базы и отображения на графики страницы html

# Некоторые детали

1. Список авторов, их можно добавлять удалять просматривать, пейджинг сделал с помощью модуля will\_paginate

страница <http://localhost:3000/authors>

1. 
2. на странице просмотра автора можно добавлять книги



1. Далее сделал заполнение тестовыми данными seed.db 1000 авторов и примерно 20 тыс книг (сделал разброс чтобы графики были не равномерны) в тестах использован gem 'faker'
2. Страница с графиками:

<http://localhost:3000/graph/graph1>

<http://localhost:3000/graph/graph2>

<http://localhost:3000/graph/graph3>

для создания тоже использовал уже имеющийся gem 'chart-js-rails'

выглядит так:

