



## Урок 5

# Генератор Scaffold

Описание процедуры Scaffold. Просмотр, изменение, добавление постов. Проблема N + 1. Постраничная навигация. Рендеринг коллекций.

[Генератор Scaffold](#)

[Роуты](#)

[Список роутов. Команда rails routes](#)

[Ресурсы](#)

[Параметры](#)

[Контроллер PostsController](#)

[Представления](#)

[Show](#)

[Edit и New](#)

[Локальные переменные в partial-шаблонах](#)

[Проблема N + 1](#)

[Постраничная навигация](#)

[Рендеринг коллекций](#)

[Гем X-Ray](#)

[Домашнее задание](#)

[Дополнительные материалы](#)

[Используемая литература](#)

На прошлых уроках мы подготовили фундамент для реализации приложения: познакомились с контроллерами и моделями, настроили отношения между ними. Теперь приступим к непосредственной реализации логики приложения и разработке пользовательского интерфейса. Ruby on Rails предоставляет возможность создавать значительную часть приложения за счет использования генераторов. Сегодня мы познакомимся с новым типом генераторов – Scaffold, в переводе с английского – «строительные леса». Scaffold позволяет генерировать код для целых разделов сайта, например, сообщений или комментариев.

## Генератор Scaffold

Начнем с сообщений пользователей. Прежде, чем мы воспользуемся генератором, определимся, что можно с ними делать. Нам нужны следующие возможности: отобразить список сообщений, отобразить детальную страницу сообщения, создать, удалить и изменить сообщение. Для каждого действия необходимо создать метод (экшен) в соответствующем контроллере и свое представление, а также настроить роутинг. Все это может делать генератор rails g scaffold.

Есть две формы этого генератора:

- rails generate scaffold
- rails generate scaffold\_controller

Отличие в том, что первый генератор создает полностью все компоненты MVC: миграцию, модель, контроллер, представление и роуты. Второй создает ограниченный набор компонентов по уже существующей модели: контроллер и представления.

Обратившись к справке rails g scaffold, найдем примеры использования этого генератора. Например, для сообщений:

```
rails g scaffold post title:string:uniq body:text user:references
visible:boolean
  invoke active_record
  conflict db/migrate/20180113152435_create_posts.rb
Another migration is already named create_posts:
/Users/i.simdyanov/www/blog/db/migrate/20180108122919_create_posts.rb. Use
--force to replace this migration or --skip to ignore conflicted file.
```

Запуск генератора приводит к ошибке. Дело в том, что на предыдущих занятиях мы уже создали миграцию для таблицы сообщений и соответствующую ей модель. Чтобы избежать конфликта, к команде следует добавить параметр --skip, который позволит генератору пропустить уже созданные файлы.

Пропущенные файлы будут помечены желтой надписью skip:

```
rails g scaffold post title:string:uniq body:text user:references
visible:boolean --skip
  invoke active_record
  skip db/migrate/20180113152503_create_posts.rb
  skip app/models/post.rb
  invoke resource_route
  route resources :posts
  invoke scaffold_controller
```

```

createapp/controllers/posts_controller.rb
invoke slim
create      app/views/posts
create      app/views/posts/index.html.slim
create      app/views/posts/edit.html.slim
create      app/views/posts/show.html.slim
create      app/views/posts/new.html.slim
create      app/views/posts/_form.html.slim
invoke jbuilder
create      app/views/posts/index.json.jbuilder
create      app/views/posts/show.json.jbuilder
create      app/views/posts/_post.json.jbuilder
invoke assets
invoke js
invoke scss

```

Запустим приложение и перейдём по адресу <http://localhost:3000/posts>. Результат работы Scaffold – большая таблица, которая отображает в соответствующих колонках атрибуты постов и ссылки на страницы редактирования, удаления и создания новых сообщений (в самом низу).

Listing posts				
Title	Body	User	Disactive	
Rerry richardson stumptown echo park blog Banksy +1 gentrify. Tofu whatever banh mi scenester raw denim aesthetic you probably haven't heard of them. Squid McSweeney's party cardigan Pitchfork butcher mustache you probably haven't heard of them.	["You probably haven't heard of them messenger bag Portland Shoreditch hoodie wolf DIY. Raw denim hoodie gluten-free fixie PBR VHS craft beer letterpress. +1 you probably haven't heard of them Four Loko freegan gentrify. 8-bit echo park art tumblr vinyl party seitan food truck gentrify.", "Marfa artisan aesthetic readymade fap PBR. Helvetica lo-fi etsy ethical letterpress mixtape squid fap tumblr. Marfa single-origin coffee blog chambray vice.", "Messenger bag chambray raw denim food truck aesthetic. Cred whatever scenester single-origin coffee butcher Shoreditch sustainable dreamcatcher. Retro artisan American Apparel echo park put a bird on it scenester trust fund."]	#<User:0x00007f83162b28e8>	false	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Mlkshk gluten-free Portland viral gentrify Cosby Sweater aesthetic vinyl messenger bag. Sustainable fanny pack irony next level mustache Williamsburg aesthetic. Biodiesel gluten-free trust fund keffiyeh you probably haven't heard of them craft beer mlkshk Banksy Wes Anderson. Sartorial Williamsburg lo-fi before they sold out cred.	["Lo-fi yr butcher twee Rerry Richardson put a bird on it before they sold out sartorial. Next level cred helvetica raw denim stumptown. Art gluten-free fanny pack party tattooed. Next level banh mi craft beer moon Banksy vice artisan DIY twee.", "Pbr fixie Carles you probably haven't heard of them echo park scenester lo-fi. Biodiesel Pitchfork vegan high life dreamcatcher butcher. Tofu party single-origin coffee letterpress seitan. Viral Brooklyn brunch locavore leggings artisan.", "Cliche tofu cred bicycle rights American Apparel fap Carles Banksy. Blog DIY iPhone tumblr high life squid bicycle rights PBR. Vhs etsy Wes Anderson jean shorts next level Wayfarers lomo. Vhs vice put a bird on it scenester Pitchfork Shoreditch food truck keytar yr."]	#<User:0x00007f83162a9ea0>	false	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Vice food truck vinyl scenester organic. You probably haven't heard of them hoodie jean shorts sartorial squid bicycle rights. Skateboard yr moon DIY Cosby Sweater before they sold out artisan seitan jean shorts. American apparel blog Marfa Carles butcher trust fund. Etsy quinoa Williamsburg Portland VHS gluten-free.	["Pitchfork gentrify single-origin coffee viral echo park messenger bag farm-to-table. Fixie mustache thundercats brunch Four Loko keytar. Ethical butcher Brooklyn thundercats beard stumptown squid before they sold out.", "Chambray squid cred aesthetic Marfa sustainable mlkshk. Chambray aesthetic vice iPhone freegan. Diy Marfa banh mi Four Loko you probably haven't heard of them.", "Rerry richardson blog craft beer gentrify skateboard. Diy seitan cliche lo-fi moon gluten-free jean shorts synth ethical. Jean shorts craft beer vegan fanny pack gentrify wolf mustache cred squid."]	#<User:0x00007f83162a2920>	false	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Vegan cred food truck Wayfarers artisan sartorial. Farm-to-table American Apparel chambray single-origin coffee letterpress synth tumblr +1 Portland. Party retro artisan vice Rerry Richardson. Moon organic Austin single-origin coffee fixie Carles cred.	["Tumblr quinoa Wes Anderson etsy sartorial Austin. Readymade before they sold out whatever Rerry Richardson high life. Master cleanse food truck sustainable synth craft beer Marfa hoodie cred iPhone. Mlkshk seitan McSweeney's Wayfarers raw denim lomo. Next level bicycle rights whatever viral Shoreditch skateboard lo-fi freegan.", "Jean shorts Portland squid wolf cred. Banksy sustainable etsy retro vegan viral. Mixtape master cleanse Marfa McSweeney's 8-bit tofu. Mcsweeney's stumptown master cleanse beard party trust fund bicycle rights salvia. Scenester master cleanse Carles freegan butcher gluten-free.", "Quinoa art aesthetic tattooed moon synth. Gluten-free tumblr blog lomo artisan letterpress art beard bicycle rights. Vhs Shoreditch PBR craft beer American Apparel."]	#<User:0x00007f8316800160>	false	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>

Теперь можно перейти по созданным ссылкам и убедиться в их работоспособности. Результат вряд ли претендует на звание лучшего дизайнерского решения десятилетия, но базовый шаблон, предоставленный Rails, сильно сократит время разработки.

## Роуты

В отчете, созданном генератором, есть упоминание об изменениях, которые были внесены в роуты. Откроем файл `config/routes.rb` и посмотрим на них.

## config/routes.rb

```
Rails.application.routes.draw do
  resources :posts
  root 'home#index'
end
```

## Список роутов. Команда rails routes

Scaffold добавил одну строку – `resources :posts`. Она добавляет ресурс – набор стандартных роутов, обслуживающих модель. Выполним команду `rails routes` с параметром `-g`, который позволяет отфильтровать роуты, относящиеся к определенной модели.

```
rails routes -g posts
  Prefix Verb   URI Pattern               Controller#Action
  posts GET    /posts(.:format)         posts#index
             POST    /posts(.:format)         posts#create
  new_post GET    /posts/new(.:format)     posts#new
  edit_post GET    /posts/:id/edit(.:format) posts#edit
  post GET    /posts/:id(.:format)     posts#show
             PATCH  /posts/:id(.:format)     posts#update
             PUT    /posts/:id(.:format)     posts#update
             DELETE /posts/:id(.:format)     posts#destroy
```

Команда `rails routes` возвращает отчет в виде четырех столбцов.

Первый содержит алиас, который используется затем в `path`-хелперах. Добавив суффикс `_path` или `_url`, например, к алиасу `posts` можно получить ссылку на роут – `posts_path` или `posts_url`.

Второй столбец сообщает HTTP-метод, который используется для обращения к серверу по текущему роуту. Браузеры поддерживают лишь ограниченный набор методов – `GET` и `POST`, поэтому остальные методы Ruby on Rails эмулирует, добавляя параметр `_method`.

Третий столбец сообщает путь роута от домена. При помощи символов, начинающихся с двоеточия, указываются параметры. Необязательные части роута заключаются в круглые скобки. В примере выше они используются в отношении параметра `:format`, предназначенного для передачи расширения (`/posts.html` или `posts.json`).

Четвертый столбец сообщает, какой контроллер и экшен обработает текущий роут. Например, строка `posts#index` говорит, что для обработки роута будет использован метод `index` класса `PostsController`.

## Ресурсы

Как видно из отчета команды `rails routes`, один метод `resource` в файле `config/routes.rb` позволяет создать сразу восемь методов с наиболее типичными CRUD-операциями. Чтобы воссоздать структуру роутов, которые предоставляют метод `resource`, в файл `config/routes.rb` пришлось бы добавить восемь строчек, объявляющих каждый роут отдельно.

## config/routes.rb

```

Rails.application.routes.draw do
  get 'posts(.:format)', to: 'posts#index', as: :posts
  post 'posts(.:format)', to: 'posts#create'
  get '/posts/new(.:format)', to: 'posts#new', as: :new_post
  get '/posts/:id/edit(.:format)', to: 'posts#edit', as: :edit_post
  get '/posts/:id(.:format)', to: 'posts#show', as: :post
  patch '/posts/:id(.:format)', to: 'posts#update'
  put '/posts/:id(.:format)', to: 'posts#update'
  delete '/posts/:id(.:format)', to: 'posts#destroy'
end

```

Такой полный вариант позволяет легко настроить роуты, например удалить те из них, которые не требуются. Аналогичным функционалом обладает метод `resources`. Например, при помощи параметра `only` можно оставить только те методы, которые указаны в массиве после параметра.

### config/routes.rb

```

Rails.application.routes.draw do
  resources :posts, only: [:index, :show]
  root 'home#index'
end

```

Теперь команда `rails routes` будет возвращать только два роута из восьми:

```

rails routes -g posts
Prefix Verb URI Pattern          Controller#Action
posts GET  /posts(.:format)  posts#index
post GET  /posts/:id(.:format) posts#show

```

Помимо команды `only` можно использовать параметр `except`. Его действие противоположно: из восьми методов исключаются только те, которые указываются в `except`.

## Параметры

Фрагменты роутов, которые начинаются с двоеточия, называются параметрами. Вместо них могут использоваться произвольные значения. Их можно извлечь в экшенах контроллера через предопределенный хэш `params`. Продемонстрируем работу с параметрами на примере роута `'home#index'` и соответствующего ему метода `index` контроллера `HomeController`.

Добавим в `config/routes.rb` еще один роут, который использует в качестве обработчика этот же экшен.

### config/routes.rb

```

Rails.application.routes.draw do
  resources :posts
  root 'home#index'
  get '/home/:hello', to: 'home#index'
end

```

Роут `/home/:hello` позволяет задавать произвольные значения вместо параметра `:hello`, например `/hello/world` или `/hello/ruby`. Чтобы получить к ним доступ, в методе `index` контроллера `HomeController` достаточно обратиться к хэшу `params`, в который помещаются все доступные параметры.

```
class HomeController < ApplicationController
  def index
    Rails.logger.info '#####'
    Rails.logger.info params.inspect
    Rails.logger.info '#####'
  end
end
```

Здесь для вывода в консоль сервера используется метод `Rails.logger.info`. Чтобы среди логов было проще обнаружить отладочную информацию, до и после нее выводится последовательность символов `#`.

Если теперь обратимся к серверу при помощи адреса `http://localhost:3000/home/ruby`, обнаружим в лог-выводе следующие строки:

```
Started GET "/home/ruby" for 127.0.0.1 at 2018-01-14 14:11:57 +0300
  (0.7ms)  SELECT "schema_migrations"."version" FROM "schema_migrations" ORDER
BY "schema_migrations"."version" ASC
Processing by HomeController#index as HTML
  Parameters: {"hello"=>"ruby"}
#####
<ActionController::Parameters {"controller"=>"home", "action"=>"index",
"hello"=>"ruby"} permitted: false>
#####
  Rendering home/index.html.slim within layouts/application
  Rendered home/index.html.slim within layouts/application (14.7ms)
Completed 200 OK in 1586ms (Views: 1579.1ms | ActiveRecord: 0.0ms)
```

Видно, что параметры содержат три элемента: имя контроллера `home`, имя экшена `index` и значение параметра `:hello` — `'ruby'`. Обратиться к значению параметра можно либо при помощи `params[:hello]`, либо при помощи `params['hello']`.

Если обратиться к адресу `http://localhost:3000/home/`, возникнет ошибка несоответствия роута, так как после `home/` обязательно должно следовать хоть что-то. Чтобы исключить возникновение ошибки, параметр `:hello` можно сделать необязательным, заключив часть роута с параметром в круглые скобки.

#### **config/routes.rb**

```
Rails.application.routes.draw do
  resources :posts
  root 'home#index'
  get '/home(/:hello)', to: 'home#index'
end
```

# Контроллер PostsController

Генератор также создал контроллер PostsController, который находится по пути `app/controllers/posts_controller.rb`.

**`app/controllers/posts_controller.rb`**

```
class PostsController < ApplicationController
  before_action :set_post, only: [:show, :edit, :update, :destroy]

  # GET /posts
  # GET /posts.json
  def index
    @posts = Post.all
  end

  # GET /posts/1
  # GET /posts/1.json
  def show
  end

  # GET /posts/new
  def new
    @post = Post.new
  end

  # GET /posts/1/edit
  def edit
  end

  # POST /posts
  # POST /posts.json
  def create
    @post = Post.new(post_params)

    respond_to do |format|
      if @post.save
        format.html { redirect_to @post, notice: 'Post was successfully
created.' }
        format.json { render :show, status: :created, location: @post }
      else
        format.html { render :new }
        format.json { render json: @post.errors, status: :unprocessable_entity }
      end
    end
  end

  # PATCH/PUT /posts/1
  # PATCH/PUT /posts/1.json
  def update
```

```

    respond_to do |format|
      if @post.update(post_params)
        format.html { redirect_to @post, notice: 'Post was successfully
updated.' }
        format.json { render :show, status: :ok, location: @post }
      else
        format.html { render :edit }
        format.json { render json: @post.errors, status: :unprocessable_entity }
      end
    end
  end

  # DELETE /posts/1
  # DELETE /posts/1.json
  def destroy
    @post.destroy
    respond_to do |format|
      format.html { redirect_to posts_url, notice: 'Post was successfully
destroyed.' }
      format.json { head :no_content }
    end
  end

  private

  # Use callbacks to share common setup or constraints between actions.
  def set_post
    @post = Post.find(params[:id])
  end

  # Never trust parameters from the scary internet, only allow the white list
through.
  def post_params
    params.require(:post).permit(:title, :body, :user_id, :visible)
  end
end

```

Класс этого контроллера, как и контроллера главной страницы, является наследником класса ApplicationController.

Сначала мы видим вызов функции `before_action` в макро-стиле. Эта функция бывает очень полезна. Она позволяет выполнить определенные действия перед вызовом какого-либо экшена контроллера. В данном случае `before_action :set_post, only: [:show, :edit, :update, :destroy]` указывает на выполнение функции `set_post` перед указанными в атрибуте `only` экшенами. Функция `set_post` определена в секции `private` класса контроллера и устанавливает в переменную `@post` объект модели `Post`, найденный в базе данных по параметру запроса `id`.

Далее идет экшен `index`, который служит для отображения всех постов. В теле экшена в переменную `@posts` записываются все объекты модели `Post`.

Экшен `new` инициализирует в переменную `@post` новый объект модели `Post`.



Экшены `edit` и `show` не делают никаких дополнительных действий. Напомню, что перед этими экшенами вызывается функция `set_post`.

Экшен `create` инициализирует новый объект модели `Post` параметрами, полученными из вызова метода `post_params`. Метод объявлен как `private` метод класса и возвращает только те параметры запроса, которые необходимы для создания нового сообщения. Метод `@post.save` сохраняет объект модели в базу данных и возвращает `true` или `false` в зависимости от успеха. В случае успешного сохранения сообщения происходит переадресация на страницу только что созданного сообщения. В случае неудачи – возврат обратно на страницу создания сообщения.

Экшен `update` ведет себя аналогично `create`, но вместо инициализации нового сообщения использует запись, полученную из метода `set_post`. Вместо `save` вызывается метод `@post.update(post_params)`, который по заданным атрибутам обновляет запись в базе данных.

Обратите внимание на то, что каждый метод также поддерживает ответ в формате не только `html`, но и `json`. Это нужно для того, чтобы помимо обычных запросов строить и AJAX-запросы.

## Представления

Каждому экшену контроллера соответствует один из роутов. Рассмотрим некоторые из них.

### Show

**app/views/posts/show.html.slim**

```
p#notice = notice

p
  strong Title:
    = @post.title
p
  strong Body:
    = @post.body
p
  strong User:
    = @post.user
p
  strong Visible:
    = @post.visible

=> link_to 'Edit', edit_post_path(@post)
'|
=< link_to 'Back', posts_path
```

Это представление для страницы сообщения. В первой строке используется метод `notice`, который возвращает значение `flash[:notice]`. В остальном код сам документирует себя. Внимание стоит обратить лишь на создание ссылок и на хелперы, генерирующие пути `edit_post_path` и `post_path`.

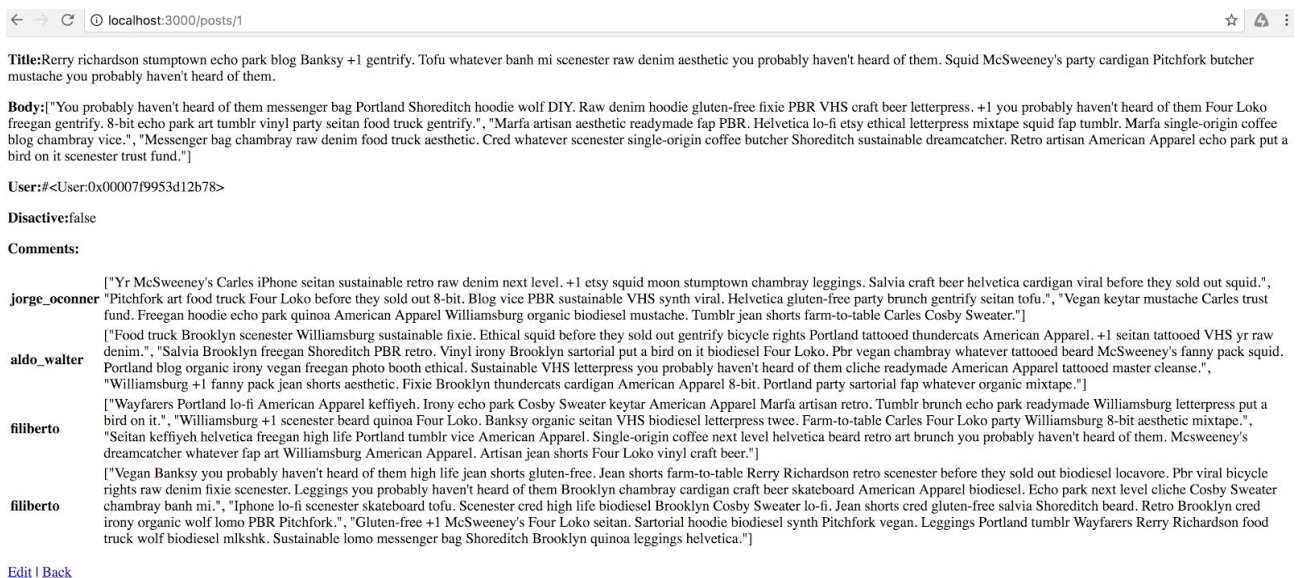
Добавим к отображению поста отображение оставленных под ним комментариев. Изменим `app/views/posts/show.html.slim` следующим образом:

## app/views/posts/show.html.slim

```
...
p
  strong Visible:
    = @post.visible
p
  strong Comments:
    table
      - @post.comments.order(:created_at).each do |comment|
        tr
          td: b = comment.user.name
          td = comment.body
...

```

Если вы не добавляли отношение `has_many :comments` для модели `Post`, следует сделать это сейчас.



Перейдя по ссылке отображения поста, вы должны увидеть что-то вроде этого.

## Edit и New

Эти два представления довольно похожи, потому что форма создания и редактирования поста фактически одинакова. Чтобы не дублировать одинаковый код в Rails, есть шаблоны (partials). Чтобы вставить partial в определенное место, нужно вызвать функцию `render partial_name`. `Partial_name` здесь – это имя файла partial'a без префикса `«_»` и расширения относительно текущего файла `app/views` либо `app/views/application`. Рассмотрим, как это реализовано в `edit.html.slim` и `new.html.slim`.

### app/views/posts/edit.html.slim

```
h1 Editing post
== render 'form'
=> link_to 'Show', @post
'|
<= link_to 'Back', posts_path

```

## app/views/posts/new.html.slim

```
h1 New post
== render 'form'
= link_to 'Back', posts_path
```

Как видим, эти представления отличаются только наличием ссылки на текущее сообщение и заголовком. Рассмотрим partial form, который расположен в файле app/view/posts/\_form.html.slim.

## app/views/posts/\_form.html.slim

```
= form_for @post do |f|
  - if @post.errors.any?
    #error_explanation
    h2 = "#{pluralize(@post.errors.count, "error")} prohibited this post from
being saved:"
    ul
      - @post.errors.full_messages.each do |message|
        li = message
  .field
    = f.label :title
    = f.text_field :title
  .field
    = f.label :body
    = f.text_area :body
  .field
    = f.label :user
    = f.text_field :user
  .field
    = f.label :visible
    = f.check_box :visible
  .actions = f.submit
```

Здесь используется хелпер `form_for`, который генерирует форму для объекта `@post`. Объект `@post`, как мы помним, для экшена `new` является новым проинициализированным объектом модели `Post`. А для `edit` – объектом модели со значениями, найденными в базе данных. Этот хелпер автоматически заполнит поля ввода форм значениями соответствующих атрибутов в объекте `@post`. Подробнее почитать про создание форм можно по ссылкам в дополнительных материалах. Обратите внимание на вывод ошибок: такой способ доступа к ошибкам вам должен быть знаком из предыдущих уроков.

## Локальные переменные в partial-шаблонах

Инстанс-переменные, например `@post`, доступны во всех вложенных partial-шаблонах. В рамках представлений фактически они выступают глобальными переменными. Такая широкая область видимости может приводить к трудно улавливаемым ошибкам и сложному сопровождению представлений. Если partial-шаблоны вложены глубоко, часто трудно проследить источник той или иной инстанс-переменной. Чтобы ограничить область видимости переменных, прибегают к локальным переменным.

Например, в форме `_form.html.slim` можно заменить инстанс-переменную `@post` на локальную переменную `post`.

#### `app/views/posts/_form.html.slim`

```
= form_for post do |f|
  - if post.errors.any?
    #error_explanation
    h2 = "#{pluralize(@post.errors.count, "error")} prohibited this post from
being saved:"
    ul
      - post.errors.full_messages.each do |message|
        li = message
  .field
    = f.label :title
    = f.text_field :title
  ...
```

Теперь, чтобы воспользоваться формой, в вызове `render` следует явно определить локальную переменную `post`.

#### `app/views/posts/edit.html.slim`

```
h1 Editing post
== render 'form', post: @post
=> link_to 'Show', @post
'|
=< link_to 'Back', posts_path
```

Такой подход более удобен в сопровождении и отладке.

## Проблема N + 1

Списочная страница `index` содержит колонку с именем автора, в которой сейчас выводится просто дамп объекта модели `User`. В первую очередь давайте исправим эту ситуацию и выведем имя пользователя. Для этого в представлении `posts/index.html.slim` исправим `post.user` на `post.user.name`. Теперь страница со списком сообщений выглядит так:

Listing posts				
Title	Body	User	Disactive	
Rerry richardson stumptown echo park blog Banksy +1 gentrify. Tofu whatever banh mi scenester raw denim aesthetic you probably haven't heard of them. Squid McSweeney's party cardigan Pitchfork butcher mustache you probably haven't heard of them.	["You probably haven't heard of them messenger bag Portland Shoreditch hoodie wolf DIY. Raw denim hoodie gluten-free fixie PBR VHS craft beer letterpress. +1 you probably haven't heard of them Four Loko freegan gentrify. 8-bit echo park art tumblr vinyl party seitan food truck gentrify.", "Marfa artisan aesthetic readymade fap PBR. Helvetica lo-fi etsy ethical letterpress mixtape squid fap tumblr. Marfa single-origin coffee blog chambray vice.", "Messenger bag chambray raw denim food truck aesthetic. Cred whatever scenester single-origin coffee butcher Shoreditch sustainable dreamcatcher. Retro artisan American Apparel echo park put a bird on it scenester trust fund."]	jorge_oconner	false	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Mlkshk gluten-free Portland viral gentrify Cosby Sweater aesthetic vinyl messenger bag. Sustainable fanny pack irony next level mustache Williamsburg aesthetic. Biodiesel gluten-free trust fund keffiyeh you probably haven't heard of them craft beer mlkshk Banksy Wes Anderson. Sartorial Williamsburg lo-fi before they sold out cred.	["Lo-fi yr butcher twee Rerry Richardson put a bird on it before they sold out sartorial. Next level cred helvetica raw denim stumptown. Art gluten-free fanny pack party tattooed. Next level banh mi craft beer moon Banksy vice artisan DIY twee.", "Pbr fixie Carles you probably haven't heard of them echo park scenester lo-fi. Biodiesel Pitchfork vegan high life dreamcatcher butcher. Tofu party single-origin coffee letterpress seitan. Viral Brooklyn brunch locavore leggings artisan.", "Cliche tofu cred bicycle rights American Apparel fap Carles Banksy. Blog DIY iPhone tumblr high life squid bicycle rights PBR. Vhs etsy Wes Anderson jean shorts next level Wayfarers lomo. Vhs vice put a bird on it scenester Pitchfork Shoreditch food truck keytar yr."]	jorge_oconner	false	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Vice food truck vinyl scenester organic. You probably haven't heard of them hoodie jean shorts sartorial squid bicycle rights. Skateboard yr moon DIY Cosby Sweater before they sold out artisan seitan jean shorts. American apparel blog Marfa Carles butcher trust fund. Etsy quinoa Williamsburg Portland VHS gluten-free.	["Pitchfork gentrify single-origin coffee viral echo park messenger bag farm-to-table. Fixie mustache thundercats brunch Four Loko keytar. Ethical butcher Brooklyn thundercats beard stumptown squid before they sold out.", "Chambray squid cred aesthetic Marfa sustainable mlkshk. Chambray aesthetic vice iPhone freegan. Diy Marfa banh mi Four Loko you probably haven't heard of them.", "Rerry richardson blog craft beer gentrify skateboard. Diy seitan cliche lo-fi moon gluten-free jean shorts synth ethical. Jean shorts craft beer vegan fanny pack gentrify wolf mustache cred squid."]	filiberto	false	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Vegan cred food truck Wayfarers artisan sartorial. Farm-to-table American Apparel chambray single-origin coffee letterpress synth tumblr +1 Portland. Party retro artisan vice Rerry Richardson. Moon organic Austin single-origin coffee fixie Carles cred.	["Tumblr quinoa Wes Anderson etsy sartorial Austin. Readymade before they sold out whatever Rerry Richardson high life. Master cleanse food truck sustainable synth craft beer Marfa hoodie cred iPhone. Mlkshk seitan McSweeney's Wayfarers raw denim lomo. Next level bicycle rights whatever viral Shoreditch skateboard lo-fi freegan.", "Jean shorts Portland squid wolf cred. Banksy sustainable etsy retro vegan viral. Mixtape master cleanse Marfa McSweeney's 8-bit tofu. Mcsweeney's stumptown master cleanse beard party trust fund bicycle rights salvia. Scenester master cleanse Carles freegan butcher gluten-free.", "Quinoa art aesthetic tattooed moon synth. Gluten-free tumblr blog lomo artisan letterpress art beard bicycle rights. Vhs Shoreditch PBR craft beer American Apparel."]	filiberto	false	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>

В колонке User вместо дампа объекта пользователя появилось его имя. Если при этом мы посмотрим в консольный вывод сервера, обнаружим, что для каждой строки таблицы отправляется отдельный запрос на извлечение пользователя.

```
Processing by PostsController#index as HTML
  Rendering posts/index.html.slim within layouts/application
  Post Load (0.5ms)  SELECT "posts".* FROM "posts"
  User Load (0.2ms)  SELECT "users".* FROM "users" WHERE "users"."id" = $1
LIMIT $2  [["id", 4], ["LIMIT", 1]]
  CACHE User Load (0.0ms)  SELECT "users".* FROM "users" WHERE "users"."id" =
$1 LIMIT $2  [["id", 4], ["LIMIT", 1]]
  User Load (0.2ms)  SELECT "users".* FROM "users" WHERE "users"."id" = $1
LIMIT $2  [["id", 3], ["LIMIT", 1]]
  CACHE User Load (0.0ms)  SELECT "users".* FROM "users" WHERE "users"."id" =
$1 LIMIT $2  [["id", 3], ["LIMIT", 1]]
  User Load (0.1ms)  SELECT "users".* FROM "users" WHERE "users"."id" = $1
LIMIT $2  [["id", 6], ["LIMIT", 1]]
  CACHE User Load (0.0ms)  SELECT "users".* FROM "users" WHERE "users"."id" =
$1 LIMIT $2  [["id", 3], ["LIMIT", 1]]
  CACHE User Load (0.0ms)  SELECT "users".* FROM "users" WHERE "users"."id" =
$1 LIMIT $2  [["id", 3], ["LIMIT", 1]]
  User Load (0.2ms)  SELECT "users".* FROM "users" WHERE "users"."id" = $1
LIMIT $2  [["id", 5], ["LIMIT", 1]]
  User Load (0.3ms)  SELECT "users".* FROM "users" WHERE "users"."id" = $1
LIMIT $2  [["id", 2], ["LIMIT", 1]]
  CACHE User Load (0.0ms)  SELECT "users".* FROM "users" WHERE "users"."id" =
$1 LIMIT $2  [["id", 6], ["LIMIT", 1]]
  CACHE User Load (0.0ms)  SELECT "users".* FROM "users" WHERE "users"."id" =
$1 LIMIT $2  [["id", 6], ["LIMIT", 1]]
  CACHE User Load (0.0ms)  SELECT "users".* FROM "users" WHERE "users"."id" =
$1 LIMIT $2  [["id", 4], ["LIMIT", 1]]
  CACHE User Load (0.0ms)  SELECT "users".* FROM "users" WHERE "users"."id" =
$1 LIMIT $2  [["id", 6], ["LIMIT", 1]]
```

```

  CACHE User Load (0.0ms)  SELECT "users".* FROM "users" WHERE "users"."id" =
$1 LIMIT $2  [["id", 5], ["LIMIT", 1]]
  User Load (0.2ms)  SELECT "users".* FROM "users" WHERE "users"."id" = $1
LIMIT $2  [["id", 7], ["LIMIT", 1]]
  CACHE User Load (0.0ms)  SELECT "users".* FROM "users" WHERE "users"."id" =
$1 LIMIT $2  [["id", 6], ["LIMIT", 1]]
  CACHE User Load (0.0ms)  SELECT "users".* FROM "users" WHERE "users"."id" =
$1 LIMIT $2  [["id", 3], ["LIMIT", 1]]
  User Load (0.1ms)  SELECT "users".* FROM "users" WHERE "users"."id" = $1
LIMIT $2  [["id", 1], ["LIMIT", 1]]
  CACHE User Load (0.0ms)  SELECT "users".* FROM "users" WHERE "users"."id" =
$1 LIMIT $2  [["id", 5], ["LIMIT", 1]]
  CACHE User Load (0.0ms)  SELECT "users".* FROM "users" WHERE "users"."id" =
$1 LIMIT $2  [["id", 5], ["LIMIT", 1]]
  Rendered posts/index.html.slim within layouts/application (54.8ms)
Completed 200 OK in 79ms (Views: 74.3ms | ActiveRecord: 1.8ms)

```

Несмотря на то, что часть повторяющихся запросов кэширована и повторно не выполняется, запросов очень много. При высокой нагрузке большое количество однотипных запросов могут привести к проблемам производительности. Такое поведение часто называют «N + 1 проблемой». Для ее решения в Ruby on Rails предусмотрена возможность упреждающей загрузки. Она позволяет загрузить пользователей заранее одним запросом. Для этого при формировании коллекции `@posts` в методе `PostsController#index` необходимо воспользоваться методом `includes` и указать, что пользователей, связанных с сообщениями ассоциацией `user`, следует загрузить заранее.

```

class PostsController < ApplicationController
  ...
  def index
    @posts = Post.includes(:user).all
  end
  ...
end

```

Теперь для формирования списочной страницы достаточно лишь двух запросов.

```

Started GET "/"posts" for 127.0.0.1 at 2018-01-14 16:05:45 +0300
Processing by PostsController#index as HTML
  Rendering posts/index.html.slim within layouts/application
  Post Load (0.3ms)  SELECT "posts".* FROM "posts"
  User Load (12.2ms)  SELECT "users".* FROM "users" WHERE "users"."id" IN (4, 3,
6, 5, 2, 7, 1)
  Rendered posts/index.html.slim within layouts/application (67.4ms)
Completed 200 OK in 100ms (Views: 76.0ms | ActiveRecord: 17.4ms)

```

## Постраничная навигация

Сообщений на главной странице может быть очень много. Это неудобно пользователям, а также приводит к длительному рендерингу на стороне сервера и, как следствие, к медленной работе сайта.



Поэтому длинные списки принято разбивать на порции по несколько сообщений на странице. Для навигации по страницам используются ссылки на последующие страницы (постраничная навигация).

Для реализации постраничной навигации есть большое количество готовых гемов, например `will_paginate` или `kaminari`. Мы воспользуемся последним. Для этого в файл `Gemfile` добавим строку с подключением гема и выполним команду `bundle install` для его установки.

### Gemfile

```
...
gem 'kaminari'
...
```

По умолчанию для передачи номера страницы гем использует параметр `page`, который в контроллере доступен через элемент хэша `params[:page]`. Его следует передать методу `page` коллекции. Для того, чтобы ограничить количество сообщений на странице, например, тремя, можно использовать метод `per`.

### app/controllers/posts\_controller.rb

```
class PostsController < ApplicationController
  ...
  def index
    @posts = Post.includes(:user).page(params[:page]).per(3)
  end
  ...
end
```

Теперь на списочной странице сообщений можно использовать метод `paginate`, который принимает в качестве аргумента коллекцию `@posts`, сформированную в методе `PostsController#index`.

### app/views/posts/index.html.slim

```
...
p
  = link_to 'New Post', new_post_path
p
  = paginate @posts
```

После этого на одной странице будут выводиться лишь три записи, а в конце списка появится навигационная строка, позволяющая перейти на следующие страницы.

Listing posts			
Title	Body	User	Disactive
Rerry richardson stumptown echo park blog Banksy +1 gentrify. Tofu whatever banh mi scenester raw denim aesthetic you probably haven't heard of them. Squid McSweeney's party cardigan Pitchfork butcher mustache you probably haven't heard of them.	["You probably haven't heard of them messenger bag Portland Shoreditch hoodie wolf DIY. Raw denim hoodie gluten-free fixie PBR VHS craft beer letterpress. +1 you probably haven't heard of them Four Loko freegan gentrify. 8-bit echo park art tumblr vinyl party seitan food truck gentrify.", "Marfa artisan aesthetic readymade fap PBR. Helvetica lo-fi etsy ethical letterpress mixtape squid fap tumblr. Marfa single-origin coffee blog chambray vice.", "Messenger bag chambray raw denim food truck aesthetic. Cred whatever scenester single-origin coffee butcher Shoreditch sustainable dreamcatcher. Retro artisan American Apparel echo park put a bird on it scenester trust fund."]	jorge_oconner	false
Mlkshk gluten-free Portland viral gentrify Cosby Sweater aesthetic vinyl messenger bag. Sustainable fanny pack irony next level mustache Williamsburg aesthetic. Biodiesel gluten-free trust fund keffiyeh you probably haven't heard of them craft beer mlkshk Banksy Wes Anderson. Sartorial Williamsburg lo-fi before they sold out cred.	["Lo-fi yr butcher twee Rerry Richardson put a bird on it before they sold out sartorial. Next level cred helvetica raw denim stumptown. Art gluten-free fanny pack party tattooed. Next level banh mi craft beer moon Banksy vice artisan DIY twee.", "Pbr fixie Carles you probably haven't heard of them echo park scenester lo-fi. Biodiesel Pitchfork vegan high life dreamcatcher butcher. Tofu party single-origin coffee letterpress seitan. Viral Brooklyn brunch locavore leggings artisan.", "Cliche tofu cred bicycle rights American Apparel fap Carles Banksy. Blog DIY iPhone tumblr high life squid bicycle rights PBR. Vhs etsy Wes Anderson jean shorts next level Wayfarers lomo. Vhs vice put a bird on it scenester Pitchfork Shoreditch food truck keytar yr."]	jorge_oconner	false
Vice food truck vinyl scenester organic. You probably haven't heard of them hoodie jean shorts sartorial squid bicycle rights. Skateboard yr moon DIY Cosby Sweater before they sold out artisan seitan jean shorts. American apparel blog Marfa Carles butcher trust fund. Etsy quinoa Williamsburg Portland VHS gluten-free.	["Pitchfork gentrify single-origin coffee viral echo park messenger bag farm-to-table. Fixie mustache thundercats brunch Four Loko keytar. Ethical butcher Brooklyn thundercats beard stumptown squid before they sold out.", "Chambray squid cred aesthetic Marfa sustainable mlkshk. Chambray aesthetic vice iPhone freegan. Diy Marfa banh mi Four Loko you probably haven't heard of them.", "Rerry richardson blog craft beer gentrify skateboard. Diy seitan cliche lo-fi moon gluten-free jean shorts synth ethical. Jean shorts craft beer vegan fanny pack gentrify wolf mustache cred squid."]	filiberto	false
<a href="#">New Post</a> 1 2 3 4 5 ... <a href="#">Next</a> <a href="#">Last</a> »			

Переход на вторую страницу приводит к формированию URL с get-параметром page после знака вопроса:

**http://localhost:3000/posts?page=2**

Этот роут можно стилизовать, например так:

**http://localhost:3000/posts/page/2**

Для этого необходимо добавить альтернативный роут для списочной страницы сообщений в файл config/routes.rb:

**config/routes.rb**

```
Rails.application.routes.draw do
  resources :posts
  get 'posts/page/:page(.:format)', to: 'posts#index'
  ...
end
```

# Рендеринг коллекций

В настоящий момент в представлении views/posts/index.html.slim для обхода коллекции используется ruby-итератор each.

**app/views/posts/index.html.slim**

```
h1 Listing posts

table
  thead
    tr
      th Title
      th Body
```



```

th User
th Visible
th
th
th

tbody
  - @posts.each do |post|
    tr
      td = post.title
      td = post.body
      td = post.user.name
      td = post.visible
      td = link_to 'Show', post
      td = link_to 'Edit', edit_post_path(post)
      td = link_to 'Destroy', post, data: { confirm: 'Are you sure?' },
method: :delete

p
  = link_to 'New Post', new_post_path
p
  = paginate @posts

```

Содержимое блока итератора each можно выделить в отдельный partial.

#### app/views/posts/\_post.html.slim

```

tr
  td = post.title
  td = post.body
  td = post.user.name
  td = post.visible
  td = link_to 'Show', post
  td = link_to 'Edit', edit_post_path(post)
  td = link_to 'Destroy', post, data: { confirm: 'Are you sure?' }, method:
:delete

```

Используя полученный partial и метод render, можно сократить цикл в представлении index.html.slim.

#### app/views/posts/index.html.slim

```

...
tbody
  - @posts.each do |post|
    = render post
...

```

Однако даже такой подход избыточен. Дело в том, что метод render может произвести рендеринг коллекций за счет параметров :collection и :partial.

**app/views/posts/index.html.slim**

```
...
tbody
  = render partial: 'post', collection: @posts
...
```

## Гем X-Ray

Отладка сложного представления, которое содержит десятки partial-документов, включающих друг друга, – довольно сложная задача. Для ее облегчения Rails-сообщество разработало несколько удобных гемов. Один из них называется X-Ray. Этот гем позволяет отслеживать пути к представлениям и partial-частям непосредственно на сайте.

Для подключения гема в development-окружение в Gemfile надо добавить два гема: xray-rails и jquery-rails. Js-библиотека jQuery необходима для работоспособности X-Ray. До версии 5.1 она входила в состав Rails, однако в последних версиях она исключена и требует явного подключения.

### Gemfile

```
...
group :development do
  ...
  gem 'jquery-rails'
  gem 'xray-rails'
end
...
```

После внесения изменений в Gemfile необходимо выполнить команду bundle install. Кроме того, надо подключить js-библиотеки jquery и xray в манифесте app/assets/javascripts/application.js.

### app/assets/javascripts/application.js

```
...
//= require jquery3
//= require xray
...
```

После этого можно использовать сочетание клавиш Ctrl+Shift+X или Cmd+Shift+X (в Mac OS X) для активации режима просмотра путей к представлениям. Каждое представление или partial обозначается прямоугольником. Наведение мыши на него приводит к подсветке области, за которую отвечает текущее представление. В левом верхнем углу прямоугольника размещается путь к представлению. Клик мыши на представлении открывает его в редакторе.

## Listing posts

Title	Body	User	Disactive	
<a href="#">_post.html#m</a> Williamsburg aesthetic hoodie readymade Four Loko vinyl. Four loko cred gluten-free locavore scenester Marfa. Yr tumblr leggings moon jean shorts +1. Tofu retro stumptown gentrify jean shorts American Apparel freegan lo-fi. 8-bit sartorial mixtape etsy trust fund.	["Master cleanse Banksy American Apparel etsy +1 PBR cliche salvia. Craft beer trust fund Banksy Rerry Richardson Brooklyn Marfa Austin. Hoodie wolf Shoreditch quinoa retro.", "Wolf single-origin coffee skateboard tofu beard scenester stumptown tattooed gentrify. Farm-to-table ethical Four Loko iPhone leggings freegan vinyl. Fixie dreamcatcher quinoa readymade keytar DIY thundercats ethical food truck. Wes anderson next level master cleanse sartorial cred Wayfarers McSweeney's American Apparel. American apparel trust fund ethical messenger bag jean shorts organic yr thundercats.", "Blog high life brunch next level jean shorts readymade lo-fi tofu. Squid etsy lomo art Austin. Irony thundercats dreamcatcher wolf salvia Brooklyn locavore artisan art. Sustainable mustache master cleanse Brooklyn 8-bit. Yr mustache +1 Wes Anderson quinoa helvetica trust fund wolf American Apparel."]	aimce_kris	false	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
<a href="#">_post.html#m</a> brunch tumblr mlkshk fap cred tofu American Apparel. Master cleanse tumblr Four Loko mustache vice leggings McSweeney's. Seitan tattooed salvia cardigan Wayfarers Carles artisan farm-to-table. Marfa fixie etsy Wes Anderson put a bird on it brunch Pitchfork organic.	["Scenester master cleanse freegan locavore Austin. Cosby sweater keffiyeh single-origin coffee gentrify seitan Banksy Pitchfork butcher. Dreamcatcher party mustache yr organic letterpress.", "Vhs farm-to-table leggings craft beer high life art stumptown American Apparel master cleanse. Wolf organic food truck cred readymade aesthetic viral PBR. Keytar craft beer McSweeney's irony cardigan cred next level.", "Single-origin coffee butcher moon Williamsburg wolf. Master cleanse freegan +1 Wes Anderson PBR gentrify. Tofu next level moon sartorial Austin artisan."]	ozell	false	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
<a href="#">_post.html#m</a> Dreamcatcher cardigan food truck echo park American Apparel +1 Cosby Sweater salvia fap. 8-bit farm-to-table keytar Cosby Sweater squid next level. Beard put a bird on it fixie mlkshk art raw denim Cosby Sweater Four Loko. Locavore put a bird on it VHS wolf +1 next level.	["Single-origin coffee bicycle rights ethical thundercats yr brunch artisan messenger bag banh mi. Raw denim cardigan squid vinyl irony McSweeney's Portland gentrify. Party Brooklyn thundercats hoodie twee Rerry Richardson VHS gluten-free mlkshk. Quinoa gentrify tumblr Rerry Richardson mustache dreamcatcher messenger bag keffiyeh high life.", "Aesthetic squid trust fund tumblr mixtape fap. Pitchfork master cleanse cred whatever DIY. Letterpress gentrify synth irony next level VHS Cosby Sweater. Pbr seitan stumptown photo booth food truck vegan mlkshk gentrify. Quinoa party dreamcatcher chambray readymade.", "8-bit irony Wes Anderson Marfa skateboard cliche. Cred jean shorts craft beer wolf PBR next level synth vice. Letterpress thundercats vegan PBR bicycle rights twee banh mi Pitchfork."]	edwin	false	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
<a href="#">_post.html#m</a> vice banh mi Wes Anderson Rerry Richardson letterpress chambray. Iphone fixie Wayfarers locavore keffiyeh. Scenester Wes Anderson VHS cliche Shoreditch +1 messenger bag. High life +1 fixie next level American Apparel quinoa. Beard Marfa blog squid	["Farm-to-table party Pitchfork Rerry Richardson fixie. Marfa high life etsy iPhone helvetica Wayfarers. Helvetica photo booth raw denim Banksy Pitchfork before they sold out McSweeney's fixie moon. Cosby sweater gentrify Banksy letterpress fixie Carles.", "Salvia cliche chambray VHS retro. You probably haven't heard of them Rerry Richardson Williamsburg messenger bag mustache Cosby Sweater echo park. Next level gentrify Portland fanny pack dreamcatcher vinyl wolf.", "Biodiesel brunch fap tofu Brooklyn Pitchfork quinoa Banksy Cosby Sweater. Viral farm-to-table thundercats Pitchfork fap. Craft beer lomo Marfa Banksy quinoa sustainable butcher. Organic 8-bit retro ethical food truck Banksy."]	ozell	false	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>

## Домашнее задание

1. При помощи scaffold-генератора добавьте страницу управления пользователями.
2. Измените представление текущего занятия, чтобы оно соответствовало стилю вашего приложения.
3. Реализуйте постраничную навигацию для комментариев на странице сообщения.

## Дополнительные материалы

1. [Хелперы форм](#).
2. [Хелперы форм](#) по-русски.
3. Гем постраничной навигации [Kaminari](#).
4. [Гем X-Ray](#).

## Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. [Генераторы](#).
2. Майкл Хартл. Ruby on Rails для начинающих. Изучаем разработку веб-приложений на основе Rails.
3. Ruby on Rails [Guides](#).
4. Obie Fernandez. The Rails 5 Way.