Бесплатная электронная книга

УЧУСЬ phoenix-framework

Free unaffiliated eBook created from
Stack Overflow contributors.

#phoenix-framework

Около

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: phoenix-framework

It is an unofficial and free phoenix-framework ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official phoenix-framework.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

глава 1: Начало работы с феникс-каркасом

замечания

В этом разделе представлен обзор того, что такое феникс-инфраструктура, и почему разработчик может захотеть его использовать.

Следует также упомянуть любые крупные темы в рамках феникса и ссылки на связанные темы. Поскольку Documentation for phoenix-framework является новым, вам может потребоваться создать начальные версии этих связанных тем.

Версии

Версия	Дата выхода
0.1.1	2014-05-01
0.2.0	2014-05-01
0.2.1	2014-05-01
0.2.2	2014-06-05
0.2.3	2014-05-05
0.2.10	2014-05-22
0.2.11	2014-06-30
0.3.0	2014-07-01
0.3.1	2014-07-05
0.4.0	2014-08-31
0.4.1	2014-09-09
0.5.0	2014-10-14
0.6.0	2014-11-22
0.6.1	2014-11-30
0.6.2	2014-12-08
0.7.0	2014-12-10

Версия	Дата выхода
0.7.1	2014-12-10
0.7.2	2014-12-11
0.8.0	2015-01-11
0.9.0	2015-02-12
0.10.0	2015-03-08
0.11.0	2015-04-08
0.12.0	2015-05-01
0.13.0	2015-11-15
0.13.1	2015-05-17
0.14.0	2015-06-30
0.15.0	2015-07-27
0.16.0	2015-08-06
0.16.1	2015-08-06
0.17.1	2015-08-27
1.0.0	2015-08-28
1.0.1	2015-09-03
1.0.2	2015-09-07
1.0.3	2015-09-29
1.0.4	2015-12-15
1.1.0	2015-09-16
1.1.1	2015-09-27
1.1.2	2016-01-09
1.1.3	2016-01-20
v1.2.0-rc.0	2016-04-29
v1.2.0-RC.1	2016-05-25

Версия	Дата выхода
1.2.0	2016-06-23
1.2.2	2017-03-14
1.2.3	2017-03-15
1.2.4	2017-05-16
1.3.0-RC.1	2017-03-15
1.3.0-PK-2	2017-05-16

Examples

Монтаж

Основа Phoenix написана в Elixir , а сам Elixir основан на языке Erlang и использует Erlang VM, известную тем, что работает с низкими задержками, распределенными и отказоустойчивыми системами. Оба языка необходимы для использования феникс-фреймворка. Следуя следующему шагу, чтобы установить феникс-фреймворк:

1. Установите Elixir на свой компьютер. См. Установка Elixir и как установить руководство Elixir .

2. Установите диспетчер пакетов **Hex** . Hex - это необходимый инструмент для запуска приложения Phoenix и для установки любых дополнительных зависимостей, которые могут понадобиться нам на этом пути. В окне управления терминалом или командами введите:

```
$ mix local.hex
```

Эта команда установит или обновит Hex, если у вас уже есть.

3. Установите Erlang на свой компьютер. Без Erlang код Elixir не будет компилироваться, потому что Elixir использует VM Erlang для компиляции кода. Когда вы установите Elixir, вы, вероятно, тоже установили Erlang, но если это не так, следуйте этой инструкции в руководстве Elixir для установки Erlang. Однако, если у вас есть система на базе Debian, вам может потребоваться явно установить Erlang.

```
$ wget https://packages.erlang-solutions.com/erlang-solutions_1.0_all.deb && sudo dpkg -i
erlang-solutions_1.0_all.deb
$ sudo apt-get update
$ sudo apt-get install esl-erlang
```

4. Установите феникс-фрейм на свой компьютер. Как только у нас есть Elixir и Erlang, мы готовы установить архив Phoenix Mix. Архив Mix - это Zip-файл, который содержит

приложение, а также его скомпилированные файлы BEAM. Он привязан к конкретной версии приложения. Архив - это то, что мы будем использовать для создания нового базового приложения Phoenix, из которого мы можем построить. Вот команда установки архива Phoenix:

```
$ mix archive.install https://github.com/phoenixframework/archives/raw/master/phoenix_new.ez
```

Вы можете загружать пакеты вручную, если вышеуказанная команда не работает должным образом. Загрузите пакеты в файловую систему архивов Phoenix и выполните следующую команду

```
mix archive.install /path/to/local/phoenix_new.ez
```

5 Plug, Cowboy и Ecto являются компонентами феникс-фреймворка, они будут установлены автоматически с помощью mix, если вы позволите mix устанавливать свои зависимости, когда вы сначала создадите проекты Phoenix. Кроме того, если вы не разрешаете микшированию загружать эти компоненты, то mix расскажет вам, как это сделать позже.

6. Установите Node.js (не менее v5.0.0) на ваш компьютер. Это **необязательная** зависимость. Node.js требуется установить brunch.io зависимости. Brunch.io используется Phoenix для компиляции статических активов (javascript, css и т. Д.) По умолчанию.

Мы можем получить node.js со страницы загрузки . При выборе пакета для загрузки важно отметить, что Phoenix требуется версия 5.0.0 или выше.

Пользователи Mac OS X также могут установить node.js через homebrew .

Примечание. Io.js, которая является совместимой с npm платформой, основанной на Node.js, не работает с Phoenix.

Пользователи Debian / Ubuntu могут видеть ошибку, которая выглядит так:

```
sh: 1: node: not found
npm WARN This failure might be due to the use of legacy binary "node"
```

Это связано с тем, что Debian имеет конфликтующие двоичные файлы для узла: см. Обсуждение по следующему запросу SO

Не удается установить пакеты с помощью диспетчера пакетов узлов в Ubuntu

Есть две возможности решить эту проблему:

install nodejs-legacy:

```
$ apt-get install nodejs-legacy
```

или создать символическую ссылку

```
$ ln -s /usr/bin/nodejs /usr/bin/node
```

7 Установите базу данных (PostgreSQL) на свой компьютер. Phoenix настраивает приложения для использования по умолчанию, но мы можем переключиться на MySQL , передав флаг `--database mysql` при создании нового приложения. В вики PostgreSQL есть руководства по установке для нескольких различных систем.

Postgrex является прямой зависимостью от Phoenix, и он будет использоваться для создания моделей. **Postgrex** будет автоматически установлен вместе с остальными зависимостями, когда вы создадите и запустите проект Phoenix.

8 inotify-tools (для пользователей linux) Это наблюдаемый файловой системой Linux, который Phoenix использует для перезагрузки в реальном времени. (Пользователи Mac OS X или Windows могут спокойно игнорировать его.)

Пользователям Linux необходимо установить эту зависимость. Пожалуйста, ознакомьтесь с вики-программой inotify-tools для инструкций по установке для конкретного дистрибутива.

Установка скелета

Иногда вам нужна установка без каких-либо изменений, кроме минимальной установки phoenix. Эта команда даст вам это.

```
mix phoenix.new web --no-brunch --no-ecto
```

Примечание. Вы должны были установить Elixir, Erlang, Hex, Mix и архив Phoenix для установки скелета

Создание проекта Phoenix

Для создания вашего первого проекта в феникс-каркасе на данный момент у **вас должны быть установлены** Elixir, Erlang, Hex и Phoenix. У вас также должны быть установлены PostgreSQL и node.js для создания приложения по умолчанию.

Откройте терминал или командную строку и перейдите в папку в вашей файловой системе, где вы хотите **создать приложение** . `phoenix.new` - это команда mix, которая создаст для вас новый проект. Предполагая, что именем нашего приложения является `hello_phoenix_world` , тогда введите

```
$ mix phoenix.new hello_phoenix_world
```

В качестве альтернативы , мы можем запустить mix phoenix.new из любого каталога, чтобы загрузить приложение Phoenix. Phoenix примет абсолютный или относительный путь

для каталога нашего нового проекта

```
$ mix phoenix.new /Users/username/work/elixir-projects/hello_phoenix_world
```

Выход

```
mix phoenix.new hello_phoenix_world
* creating hello_phoenix_world/config/config.exs
* creating hello_phoenix_world/config/dev.exs
* creating hello_phoenix_world/config/prod.exs
...
* creating hello_phoenix_world/web/views/layout_view.ex
* creating hello_phoenix_world/web/views/page_view.ex

Fetch and install dependencies? [Yn]
```

Phoenix создаст структуру каталогов для вашего проекта и создаст все файлы, необходимые для приложения. Mix спросит вас, хотите ли вы **установить другие необходимые зависимости** . Скажем так.

```
Fetch and install dependencies? [Yn] Y
* running mix deps.get
* running npm install && node node_modules/brunch/bin/brunch build
```

После установки **зависимостей** задача предложит вам перейти в наш каталог проекта и запустить приложение.

```
Move into your new project folder:

    $cd hello_phoenix_world
```

Теперь вам нужно настроить имя пользователя и пароль postgres, если он уже не настроен с использованием по умолчанию postgres useranme и пароля postgres. Измените файл `config/dev.exs` и `config/dev.exs` имя пользователя и пароль:

```
# config/dev.exs
config :hello_phoenix_world, HelloPhoenixWorld.Repo,
  adapter: Ecto.Adapters.Postgres,
  username: "postgres",
  password: "postgres",
  database: "hello_phoenix_world_dev",
  hostname: "localhost",
  pool_size: 10

Now, create the database with the ecto mix task:

    $ mix ecto.create

We have a working application! Run your Phoenix application:

    $ mix phoenix.server

You can also run your app inside IEx (Interactive Elixir) as:
```

```
    $ iex -S mix phoenix.server

Load `http://localhost:4000` into your browser and you will see the default landing page of
your application.
```

Теперь давайте добавим мир привет в приложение Phoenix. Откройте файл `web/templates/page/index.html.eex` и замените содержимое следующим и сохраните файл:

```
<h2>Hello World</h2>
```

Если вы не покинули сервер, новый код будет автоматически скомпилирован, и ваш браузер должен теперь отобразить ваше сообщение «Hello World».

Теперь вы можете создать ресурс CRUD .

Наконец, чтобы выйти из сервера, введите `ctrl-c crtl-c` (нажмите клавишу `control key` и `c` ключ вместе) два раза подряд.

Запуск Elixir / Phoenix на OSX

Эликсир / **Феникс**

Сначала установите Homebrew :

```
/usr/bin/ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Затем, запустив `brew install elixir` установит как Elixir, так и его зависимость - Erlang.

Установите смесь со `mix local.hex` .

Установите Phoenix в соответствии с инструкциями:

```
mix archive.install https://github.com/phoenixframework/archives/raw/master/phoenix_new.ez
```

Node.js

Вы можете устанавливать и управлять версиями Node.js с помощью NVM. Установите nvm с помощью:

```
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.31.4/install.sh | bash
```

Если `curl` недоступен, вы можете установить его с помощью `brew install curl` . Затем выполните:

```
nvm install node
```

для загрузки и компиляции и последней версии Node.js.

База данных

Скачайте Postgres.app и запустите его. Когда вы создаете проект Phoenix, в вашем файле `config/dev.exs` вам просто нужно указать имя для своей базы данных - адаптер будет использовать значения по умолчанию для остальных:

```
config :myphoenixapp, MyPhoenixApp.Repo,
  adapter: Ecto.Adapters.Postgres,
  database: "myphoenixapp_dev",
  hostname: "localhost",
  pool_size: 10
```

Создание ресурсов для модели

Для создания схемы, представления, контроллера, файла миграции для репозитория, шаблонов CRUD по умолчанию и тестовых файлов для модели (например, леса в Rails) можно использовать задачу `phoenix.gen.html` mix:

```
mix phoenix.gen.html Book books title note:text pages:integer author_id:references:authors
```

Где `Book` - это имя модуля, `books` - это множественная форма, используемая для схемы, за которой следуют поля ресурсов: `title` (строка по умолчанию), `note` (текстовое поле), `pages` (целое), `author_id` которое создает ассоциацию `belongs_to` с моделью Author.

Прочитайте Начало работы с феникс-каркасом онлайн: https://riptutorial.com/ru/phoenix-framework/topic/4996/начало-работы-с-феникс-каркасом

глава 2: Использование Ecto-моделей в phoenix

Вступление

Как создавать, редактировать и использовать ecto-модели в феникс-фреймворках.

Examples

Создание модели пользователя из командной строки

Чтобы создать пользовательскую модель `json` с `username` , `password_hash` , `email_id` , `created_at` , `updated_at` , введите

```
mix phoenix.gen.json User users username:string email_id:string password_hash:string
timestamps()
```

Миграция ecto-модели

Когда вы запускаете `mix phoenix.gen.html` или `mix phoenix.gen.json` из командной строки, миграции создаются в `priv -> repo -> migrations` в папке проекта.

Для запуска миграции типа `mix ecto.migrate` .

Чтобы создать миграции для вашего проекта, `mix ecto.gen migrations <model_name>`

Чтобы создать миграцию для другого хранилища, чем по умолчанию, выполните один запуск `mix ecto.gen migrations <model_name> -r <repo_name>`

Прочитайте Использование Ecto-моделей в phoenix онлайн: https://riptutorial.com/ru/phoenix-framework/topic/10890/использование-ecto-моделей-в-phoenix

глава 3: Создание проектной документации

Examples

обоснование

Правильный вызов вспомогательных модулей и функций может быть пугающим, потому что

- они генерируются динамически (например, при создании нового проекта или добавлении нового `resource`)
- они не документируются явно (например, `MyApp.ErrorHelpers.error_tag`)
- документация не охватывает все примеры (например, `MyApp.Router.Helpers.*_path` в `Phoenix.Router`).

Хотя созданные помощники разбросаны по всему вашему проекту, но их расположение следует за твердой логикой. Вы можете привыкнуть к ним довольно быстро и, к счастью, когда вы создаете проект с Phoenix, код поставляется с документацией через `@moduledoc` модуля `@doc` и `@moduledoc` .

Эти документы не ограничиваются только помощниками, но вы также можете

- см. ваш проект, разбитый подмодулями / функциями / макросами
- добавить свою собственную документацию
- найдите любые функции, созданные в пространстве имен вашего проекта (например, `MyApp.Repo` содержит функции функции обратного вызова из `Ecto.Repo`)

Создание документов

Чтобы создать документацию из исходного кода, добавьте `ex_doc` в зависимости от вашего файла `mix.exs` :

```
# config/mix.exs

def deps do
  [{:ex_doc, "~> 0.11", only: :dev}]
end
```

> Вы можете использовать Markdown в `@doc` Elixir `@doc` и `@moduledoc` .

Затем запустите `mix deps.get` чтобы извлечь и скомпилировать новые модули и сгенерировать документацию по проекту с помощью документации по `mix docs` . Пример вывода - официальные документы Elixir .

Чтобы немедленно их обслуживать, используйте `mix docs --output priv/static/doc` и перейдите к `my_app_url_or_ip/doc/index.html` .

Дополнительное чтение:

- `ex_doc`
- Операторы требований к версии (`Elixir.Version`)

Основная часть этого руководства упоминается в рецептах эликсира .

Прочитайте Создание проектной документации онлайн: https://riptutorial.com/ru/phoenix-framework/topic/5868/создание-проектной-документации

кредиты

S. No	Главы	Contributors
1	Начало работы с феникс-каркасом	Community, helcim, penguin, Steve Pallen, SURAJ KUMAR, Svilen
2	Использование Ecto-моделей в phoenix	Faizan Ali
3	Создание проектной документации	toraritte