

 無料電子ブック

学習

phoenix-framework

Free unaffiliated eBook created from
Stack Overflow contributors.

#phoenix-
framework

.....	1
1: phoenix-framework	2
.....	2
.....	2
Examples.....	4
.....	4
.....	6
Phoenix.....	6
/OSX.....	8
.....	8
2: Ecto	10
.....	10
Examples.....	10
.....	10
.....	10
3:	11
Examples.....	11
.....	11
.....	13

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.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

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

1: phoenix-frameworkのい

このセクションではphoenix-frameworkのと、なぜがそれをいたいのかをします。

phoenix-frameworkのきなテーマについてもし、するトピックにリンクするがあります。

phoenix-frameworkのドキュメンテーションはしいので、それらのトピックのバージョンをするがあります。

バージョン

バージョン	
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	2014121010
0.7.1	2014121010
0.7.2	2014-12-11

バージョン	
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
1.2.0	2016623
1.2.2	2017-03-14

バージョン	
1.2.3	2017-03-15
1.2.4	2017-05-16
1.3.0-rc.1	2017-03-15
1.3.0-rc.2	2017-05-16

Examples

インストール

PhoenixフレームワークはElixirでかかれており、ElixirはErlangに基づいており、、、フォールトトレラントシステムであるErlang VMをしています。のはPhoenixフレームワークのにです。Phoenixフレームワークをインストールするのはのとおりです。

1.あなたのマシンにElixirをインストールします。ElixirのインストールとどのようにするElixirガイドをインストールします。

2. Hexパッケージマネージャをインストールします。Hexは、Phoenixアプリケーションをさせ、なをインストールするのになツールです。ターミナルまたはコマンドコントロールウィンドウでのようにします。

```
$ mix local.hex
```

すでにインストールされている、このコマンドはHexをインストールまたはします。

3.あなたのマシンにErlangをインストールします。Erlangがなければ、ElixirはコードのコンパイルにErlangのVMをするため、Elixirコードはコンパイルされません。ElixirをインストールするにはErlangもインストールされているがありますが、そうでないは、ElixirガイドののってErlangをインストールしてください。しかし、Debianベースのシステムをおいのは、Erlangをにインストールするがあります。

```
$ 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.あなたのマシンにPhoenixフレームワークをインストールします。ElixirとErlangをしたら、Phoenix Mixアーカイブをインストールするがいました。ミックスアーカイブは、アプリケーションとコンパイルされたBEAMファイルをむZipファイルです。のバージョンのアプリケーションにけられています。このアーカイブは、々がすることができのしいベースのPhoenixアプリケーションをするためにするものです。Phoenixアーカイブをインストールするコマンドはのとおりです。

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

のコマンドがにしないは、パッケージをダウンロードできます。パッケージをファイルシステムの[Phoenixアーカイブ](#)にダウンロードし、のコマンドをします

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

5 Plug、**Cowboy**、**Ecto**はPhoenixフレームワークのコンポーネントで、Phoenixプロジェクトをにするときには、それらのをインストールするようにすれば、にインストールされます。さらに、これらのコンポーネントをダウンロードするミックスをしないは、でそのをミックスするをえてくれるでしょう。

6.マシンに**Node.js v5.0.0**をインストールします。これはオプションのです。[brunch.io](#)のをインストールするには**Node.js**がです。**Brunch.io**は、デフォルトでアセットjavascript、cssなどをコンパイルするためにPhoenixによってされます。

[ダウンロードページ](#)からnode.jsをできます。ダウンロードするパッケージをするときは、Phoenixにバージョン5.0.0がであることにすることがです。

Mac OS Xユーザーは[homebrew](#)でnode.jsをインストールすることもできます。

Node.jsをにしたnpmプラットフォームであるio.jsは、Phoenixですることにはられていません。

Debian / Ubuntuのユーザにのようなエラーがされることがあります

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

これは、Debianがノードのためにするバイナリをっているためです。

[Ubuntuでノードパッケージマネージャをしてパッケージをインストールできない](#)

このをするには、の2つのがあります。

install nodejs-legacy

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

またはシンボリックリンクをする

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

7マシンにデータベース [PostgreSQL](#) をインストールします。Phoenixはデフォルトでアプリケーションをするようにしますが、しいアプリケーションをするときに`--database mysql`フラグをして[MySQL](#)にりえることができます。[PostgreSQL wiki](#)には、さまざまなシステムの[インストールガイド](#)があります。

PostgrexはのPhoenixであり、モデルのにされます。 Phoenixプロジェクトをしてするときに、 **Postgrex**はりのとともにインストールされます。

8 inotify-tools LinuxユーザのPhoenixがライブコードのリロードにするLinuxのファイルシステムウォッチャです。 Mac OS XまたはWindowsユーザはしてありません。

Linuxユーザーはこのをインストールするがあります。 のインストールについては、 [inotify-tools wiki](#)をしてください。

スケルトンインストール

によっては、のphoenixセットアップのインストールがなもあります。 followコマンドはあなたにそれをえるでしょう。

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

スケルトンインストールのElixir、 Erlang、 Hex、 Mix、 Phoenixアーカイブがインストールされているがあります

Phoenixプロジェクトの

ここでPhoenixフレームワークでのプロジェクトをするには、 Elixir、 Erlang、 Hex、 および Phoenixアーカイブがインストールされているがあります。 また、 PostgreSQLとnode.jsをインストールして、 デフォルトのアプリケーションをするがあります。

ターミナルまたはコマンドプロンプトをき、 アプリケーションをするファイルシステムのにします。 phoenix.newはしいプロジェクトをするmixコマンドです。 アプリケーションのが hello_phoenix_world であるとして、 のようにします。

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

あるいは Phoenixアプリケーションをブートストラップするために、のディレクトリからmix phoenix.newをすることもできます。 Phoenixは、しいプロジェクトのディレクトリのパスまたはパスのいずれかをくれます

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

```
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はプロジェクトのディレクトリをし、アプリケーションになすすべてのファイルをしします。**Mix**はのなをインストールするかどうかねます。そのことに「はい」とってみましょう。

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

がインストールされると、プロジェクトディレクトリにしてアプリケーションをするようにすプロンプトがされます。

```
Move into your new project folder:

$cd hello_phoenix_world
```

すでにpostgresのuseranmeとpostgresのパスワードをしていないり、postgresのユーザーとパスワードをするがあります。config/dev.exs ファイルをし、ユーザーとパスワードをしします

```
# 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.

さて、Phoenixアプリケーションにhello worldをししましょう。web/templates/page/index.html.eex ファイルをし、をのようにしえてファイルをしします。

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

サーバーをしていないは、しいコードがにコンパイルされ、ブラウザに「Hello World」というメッセージがされます。

CRUDリソースをしできるようにになりました。

に、サーバーをするには、ctrl-c ctrl-c ctrl-c control key とcキーをしにします。

エリクサー/フェニックスを**OSX**でかす

エリクシール/フェニックス

[ホームブリュー](#)をまずインストールしてください

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

に、 `brew install elixir` をすると、ElixirとそのErlangがインストールされます。

`mix local.hex` ミックスをインストールします。

にってPhoenixをインストールします。

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

Node.js

NVMをしてNode.jsバージョンをインストールしてすることができます。 [nvm](#)をのようにインストールします。

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

はcurlできない、あなたがそれをインストールすることができ `brew install curl`。 に、

```
nvm install node
```

Node.jsのバージョンをダウンロードしてコンパイルします。

データベース

[Postgres.app](#)をダウンロードしてします。 `config/dev.exs` ファイルでPhoenixプロジェクトをするときには、データベースのをするだけで `config/dev.exs` ます。アダプタはりのデフォルトをします

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

モデルのリソースの

リポジトリのスキーマ、ビュー、コントローラ、マイグレーションファイル、デフォルトのCRUDテンプレート、モデルのテストファイルRailsのスカフォールディングのようなをするには、 `phoenix.gen.html` ようなミックスタスクをできます

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

どこで`Book`、モジュールで`books`スキーマをである、リソースフィールドが`title`デフォルトでは、`note`テキストフィールド、`pages`、`author_id`する`belongs_to`モデルとのけを。

オンラインで`phoenix-framework`のいをむ <https://riptutorial.com/ja/phoenix-framework/topic/4996/phoenix-framework>のい

2: Ectoはフェニックスでのをモデルしています

き

phoenixフレームワークでectoモデルを、、する

Examples

コマンドラインからユーザーモデルをする

するには、`json`をつユーザーモデル `username`、`password_hash`、`email_id`、`created_at`、`updated_at`、`タイプ`を

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

エクトモデルの

`mix phoenix.gen.html` をするか、コマンドラインから `mix phoenix.gen.json` を `mix phoenix.gen.json` と、プロジェクトフォルダの `priv -> repo -> migrations` にかされます。

マイグレーションをするには、`mix ecto.migrate` ます。

プロジェクトのにマイグレーションをするには `mix ecto.gen migrations <model_name>`

デフォルトのものとはなるリポジトリのマイグレーションをするには、`mix ecto.gen migrations <model_name> -r <repo_name>`

オンラインでEctoはフェニックスでのをモデルしていますをむ <https://riptutorial.com/ja/phoenix-framework/topic/10890/ectoはフェニックスでのをモデルしています>

3: プロジェクトのドキュメントをする

Examples

ヘルパーモジュールとのしいびしは、

- これらにはにされます例えば、しいプロジェクトをしたり、しい `resource` するなど
- にされていない `MyApp.ErrorHelpers.error_tag`
- ドキュメントには、すべてのをカバーしていないえば、 `MyApp.Router.Helpers.*_path` で `Phoenix.Router`。

されたヘルパーはプロジェクトにらばっていますが、そのはいロジックにいます。あなたはそれらにかなりく `@doc` ことができます。なことに、Phoenixでプロジェクトをするとき、コードはElixirの `@doc` と `@moduledoc` モジュールをもってドキュメントとともにされます。

これらのドキュメントはヘルパーにされるものではありませんが、

- サブモジュール//マクロによってされたプロジェクトをる
- のドキュメントをする
- あなたのプロジェクトののでされたをべます例えば、 `MyApp.Repo` は `Ecto.Repo` コールバックのがまれています

ドキュメントの

、ソースコードからドキュメントをするには `ex_doc` あなたにとして `mix.exs` ファイル

```
# config/mix.exs

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

Elixir `@doc` と `@moduledoc` でMarkdownをうことができます。

に、 `mix deps.get` をして、しいモジュールをフェッチしてコンパイルし、 `mix docs` プロジェクトドキュメントをします。は、 [なElixir Docs](#) です。

すぐにそれらをするには `mix docs --output priv/static/doc` を `mix docs --output priv/static/doc` し、 `my_app_url_or_ip/doc/index.html` ナビゲートして `my_app_url_or_ip/doc/index.html`。

の

- [ex_doc](#)
- [バージョン](#) `Elixir.Version`

このガイドのは [Elixir Recipes](#) からされています。

オンラインでプロジェクトのドキュメントをするをむ <https://riptutorial.com/ja/phoenix-framework/topic/5868/プロジェクトのドキュメントをする>

クレジット

S. No		Contributors
1	phoenix-framework のい	Community , helcim , penguin , Steve Pallen , SURAJ KUMAR , Svilen
2	Ectoはフェニックス でのモデルしてい ます	Faizan Ali
3	プロジェクトのドク ュメントをする	toraritte