

【テックジム】 Railsコース 第4章 「ゼロからアプリケーションを作成しよう」


1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分


■ 4 - 0 : 講座を受ける前に

【ZOOMの使い方】

リアクションについて

ZOOM画面下部に「反応」というボタンがありますので、

- ・ 講師が皆さんの進捗を伺いますので、何も問題ない場合には、右側のいいねマーク 

- ・ 何かわからない点・つまづいた点がありましたら、左側の拍手マーク を押してください。

チャット機能について

わからないことがあれば、基本的にチャットで質問してください。

チャット機能で改行する方法

Windows → Shift + Enter.

Mac → control + Return (Enter)

質問の仕方について

下記のように、質問内容・入力したコマンドの行全体・出力結果をチャットして下さると助かります。

ex)

下記のエラーが出ます。どうすればいいですか？

```
ec2-user:~/environment/contact_app/techgym_rails_course03 (lesson4) $ git checkout lesson4
```

```
Already on 'lesson4'
```

```
Your branch is up-to-date with 'origin/lesson4'.
```

ミュートについて

基本的にはミュートしててください。

チャットでは、質問しづらい内容がありましたら、ミュートを解除し、発言してください。

【テックジム】 Railsコース 第4章 「ゼロからアプリケーションを作成しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

サンプルソースの公開場所：https://github.com/techgymjp/techgym_rails_course03

☆ 実行環境はCloud9(<https://aws.amazon.com/jp/cloud9/>)を使用する。

☆ 対象のgithubリポジトリをクローンする。

☆ 各レッスンの回答は、masterブランチのprocedureフォルダ内のフォルダをご参照ください。

■ 4 - 0 - 1 : 実行環境を整えよう

【手順】

環境設定として下記のコマンド入力してください。

コマンドは、Terminalに下記図のように入力し、EnterキーまたはReturnキーを押してください。

※ \$マークは、すでに入力されているため、\$より後ろを入力してください。

コマンドを実行しても、何も表示されない場合がありますが、問題ありません。

```
ec2-user:~/environment $ mkdir techgym_rails
```

(lesson1)

techgym_railsという名前のフォルダを作成する。

```
$ mkdir techgym_rails
```

techgym_railsフォルダに移動する。

```
$ cd techgym_rails
```

(lesson2以降)

ホームディレクトリからパス指定でtechgym_railsフォルダに移動する。

```
$ cd ~/environment/techgym_rails
```

対象のgithubリポジトリをクローンする。

クローン： github上のプロジェクトをカレントディレクトリに複製する。

```
$ git clone https://github.com/techgymjp/techgym_rails_course04.git
```

techgym_rails_course04フォルダに移動する。

```
$ cd techgym_rails_course04
```

プロジェクトに必要なプログラムをインストールする。

```
$ bundle install --path vendor/bundle
```

※ postgresqlがエラーが発生した場合

必要なパッケージをインストールする。

```
$ sudo yum install postgresql postgresql-server postgresql-devel postgresql-contrib -y
```

データベースの初期化

```
$ sudo service postgresql initdb
```

```
$ bundle install --path vendor/bundle
```

データベースサーバーの起動

```
$ sudo service postgresql start
```

データベースをセットアップする。

```
$ bundle exec rake db:setup
```

※ データベース作成時にpostgresqlのエラーが発生した場合

【テックジム】 Railsコース 第4章 「ゼロからアプリケーションを作成しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

ユーザーの作成

```
$ sudo -u postgres createuser -s ec2-user  
$ bundle exec rake db:setup
```

※ 下記のエラーが発生した場合

Could not find public_suffix-4.0.4 in any of the sources
Run `bundle install` to install missing gems.

```
$ bundle install --path vendor/bundle  
$ bundle exec rake db:setup
```

ブランチを切り替える

```
$ git checkout -b lesson2 remotes/origin/lesson2
```

Railsのサーバーを起動する。

```
$ bundle exec rails server
```

【実行結果】

URL: /

ex) <https://f24e3029423e4xxxxxx38c8888d4.vfs.cloud9.ap-northeast-1.amazonaws.com/>

ページの表示方法がわからない方は、11ページ「Cloud9でブラウザを立ち上げる」をご確認ください。



【テックジム】 Railsコース 第4章 「ゼロからアプリケーションを作成しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

■ 4 - 1 : Railsアプリケーションを作成しよう

【問題】

Railsのアプリケーションを作成しましょう。

【作業の流れ】

- ・ Rails5.2の最新バージョンをインストール
- ・ 下記の要件で、Railsの新規プロジェクト作成
 - プロジェクト名 : rails_template_new
 - バージョン : インストールしたRails5.2の最新バージョンを指定
 - オプション ;
 - データベース : postgresql
 - turbolinksのセットアップをスキップ
 - testのセットアップをスキップ
 - bundle installをスキップ
- ・ Gemをvendor/bundleフォルダ内にインストール
- ・ データベースの作成
- ・ Railsサーバーの起動

【実行結果】

URL: /

ex) <https://f24e3029423e4xxxxxx38c8888d4.vfs.cloud9.ap-northeast-1.amazonaws.com/>

ページの表示方法がわからない方は、11ページ「Cloud9でブラウザを立ち上げる」をご確認ください。



【ヒント】

- 下記コマンドを実行することで、新規Railsプロジェクトを作成する際のオプション一覧が表示されます。
`$ bundle exec rails new -h`
- 「Rails new オプション」「Rails new バージョン指定」のように検索しましょう。

【テックジム】 Railsコース 第4章 「ゼロからアプリケーションを作成しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

■ 4 - 2 : 日本語対応しよう : lesson2

【問題】

Railsではデフォルトで扱う言語が英語となっていますが、日本語に変更し、日本語のロケールファイルを作成しましょう。

ロケールファイルの作成にはGem(i18n_generators)を利用しましょう。

また、タイムゾーンをTokyoに変更しましょう。

【作業の流れ】

- ・ デフォルトで扱う言語を日本語に変更し、タイムゾーンをTokyoに変更
- ・ 利用するGemのインストール
- ・ ロケールファイルの作成

【確認方法】

コンソールを起動

```
$ bundle exec rails console
```

現在時間を確認

```
> DateTime.current
```

```
=> Tue, 16 Jun 2020 12:43:27 +0900
```

→ 正しい時間であるかどうかを確認する

現在時間の日本語化

```
> helper.l(DateTime.current, format: :default)
```

```
=> "2020年06月16日(火) 12時44分17秒 +0900
```

→ 日本語で表示されているかどうかを確認する

コンソールを終了させる。

```
> exit
```

※ コンソールを起動したままconfigフォルダ内のファイルを修正しても結果が反映されないため、configフォルダ内のファイルを修正したら、一度コンソールを終了させ、再度コンソールを起動させる必要があります。

【ヒント】

- デフォルトで扱う言語を変更するためには、config/application.rbを修正する。
- 「Rails i18n_generators 導入」「Rails 日本語化対応」のように検索しましょう

【テックジム】 Railsコース 第4章 「ゼロからアプリケーションを作成しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

■ 4 - 3 : コントローラーを作成しよう : lesson3

【始めに】

```
$ git add .  
$ git commit -m "日本語対応"  
$ git checkout -b lesson3 remotes/origin/lesson3
```

【問題】

Railsのジェネレータを利用して、コントローラーを作成しましょう。
また、ジェネレータを実行する際に、自動生成されるファイルを制限しましょう。

【作業の流れ】

- ・ 自動生成されるファイルを制限
自動生成されるファイル
app/controllers/home_controller.rb
app/views/home/index.html.erb
- ・ コントローラー(HomeController)の作成
indexアクション及び、ビュー(index)も同時に生成する
- ・ homeコントローラーのindexアクションがrootになるようにroutesを変更

【実行結果】

URL: /
ex) <https://f24e3029423e4xxxxxx38c8888d4.vfs.cloud9.ap-northeast-1.amazonaws.com/>

Home#index

Find me in app/views/home/index.html.erb

【ヒント】

- config/application.rb内に下記の記述を追記するとjavascriptの自動生成が制限されます。
config.generators do |g|
 g.javascripts false
end
- 「Rails config.generators」のように検索しましょう。
- routesを変更するには、config/routes.rbを修正する。
- 「Rails routes root」のように検索しましょう。

【参考】

- Railsのジェネレータを利用して、作成したコントローラー(home)は下記コマンドで削除することができます。
\$ bundle exec rails destroy controller home

【テックジム】 Railsコース 第4章 「ゼロからアプリケーションを作成しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

■ 4 - 4 : デザインを追加しよう : lesson4

【始めに】

```
$ git add .  
$ git commit -m "Homeコントローラー作成"  
$ git checkout -b lesson4 remotes/origin/lesson4
```

【問題】

CSSテンプレートとWEBアイコンをプロジェクトに追加し、トップページを装飾してください。

CSSテンプレート : Bootstrap(<https://getbootstrap.jp/>)

WEBアイコン : Font Awesome(<https://fontawesome.com/>)

※ BootstrapとFont Awesomeにはそれぞれ導入するためのGemが存在するので、それらのGemを利用しましょう。

【作業の流れ】

- ・ 利用するGemのインストール
- ・ cssファイル(app/assets/stylesheets/application.css)をscssファイルに変更し、利用するcssをインポート
- ・ トップページのビュー(app/views/home/index.html.erb)を修正する

教材ではホームアイコンを追加し、タイトルを中央寄せにしています。

余裕があれば、BootstrapとFont Awesomeを利用して、他のデザインも追加してみましょう。

【実行結果】

URL: /

ex) <https://f24e3029423e4xxxxxx38c8888d4.vfs.cloud9.ap-northeast-1.amazonaws.com/>



【ヒント】

□ 追加するGemは、'bootstrap'と'font-awesome-sass'の2つ

□ 「rails bootstrap 導入」「rails fontawesome 導入」のように検索しましょう。

□ 下記のように記述すると、該当のcssをインポートすることができる。

```
@import "bootstrap"
```

□ インポートするCSSは下記の3つ

```
"bootstrap"
```

```
"font-awesome-sprockets"
```

```
"font-awesome"
```

□ Bootstrapでどのようなデザインができるのか調べる場合、「bootstrap リファレンス」と検索しましょう。

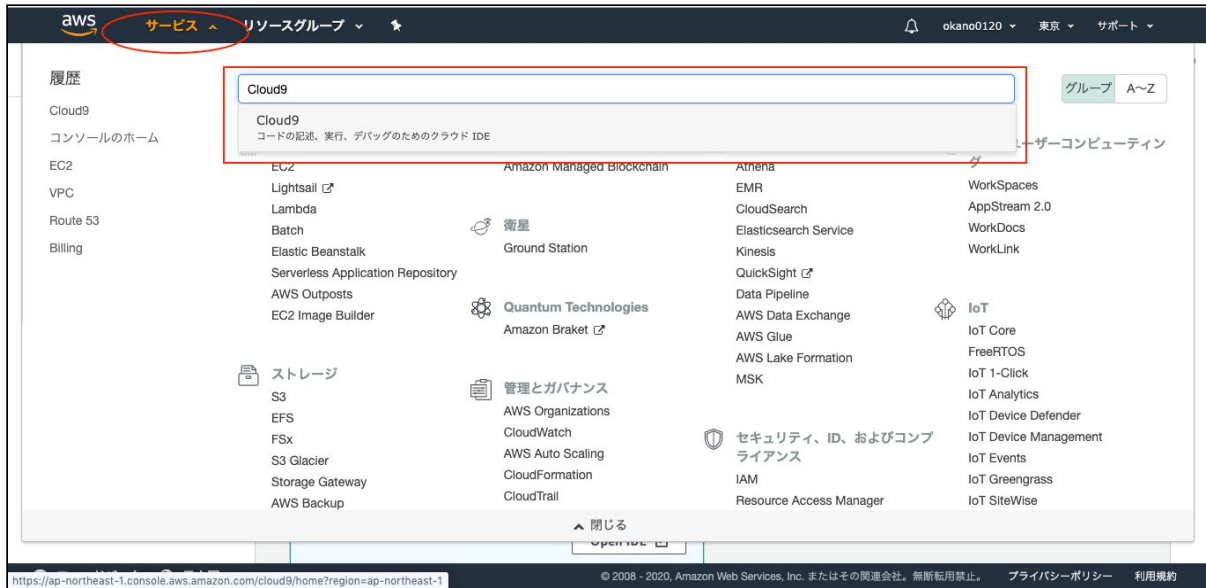
【テックジム】Railsコース 第4章 「ゼロからアプリケーションを作成しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

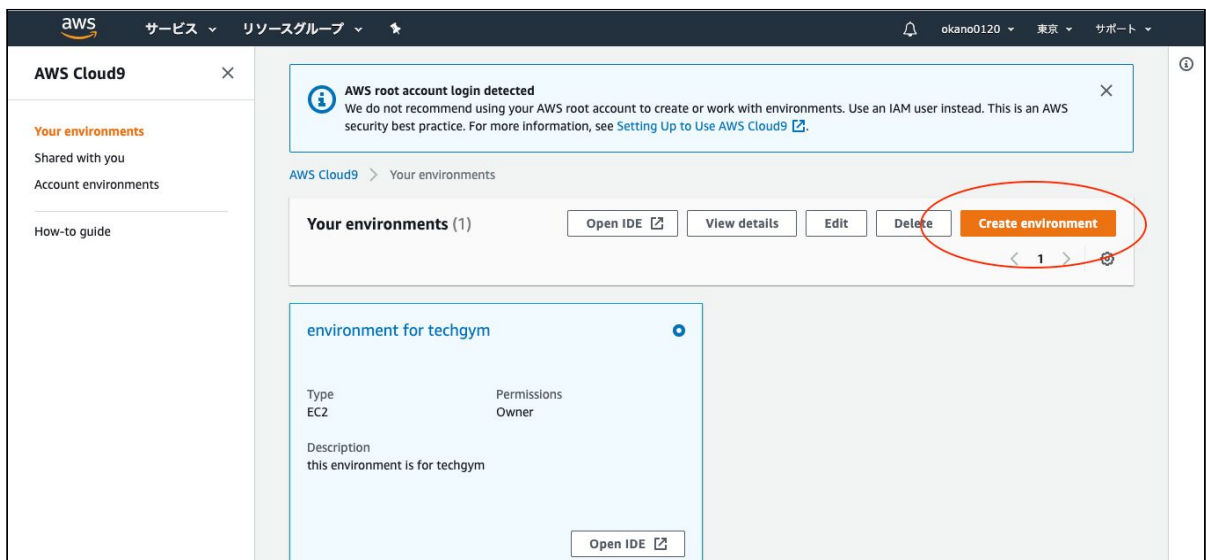
■ Cloud9の立ち上げ方

【手順】

- ・AWS(<https://aws.amazon.com/jp/>)にログインして、フッターの「サービス」をクリックし、検索フォームにCloud9と入力してます。すると、「Cloud9」の項目が出てくるので、クリックしてください。



- ・Cloud9のダッシュボードに移動するので、「Create environment」をクリック



【テックジム】 Railsコース 第4章 「ゼロからアプリケーションを作成しよう」

1回目 : / 分 2回目 : / 分 3回目 : / 分 4回目 : / 分 5回目 : / 分

- ・ Step 1 「Name environment」 では、好きな名前を入力し、任意で説明を入力してます。

AWS Cloud9 > Environments > Create environment

Step 1
Name environment

Step 2
Configure settings

Step 3
Review

Name environment

Environment name and description

Name
The name needs to be unique per user. You can update it at any time in your environment settings.

techgym_rails

Limit: 60 characters

Description - Optional
This will appear on your environment's card in your dashboard. You can update it at any time in your environment settings.

environment for techgym_rails

Limit: 200 characters

- ・ Step 2 「Configure settings」 では、下記の内容を選択し、「Next step」をクリックして下さい。
Environment type : Create a new instance for environment(EC2)
Instance type : t2.micro(1 GiB RAM + 1 vCPU)
Platform : Amazon Linux
Cost-saving setting : After 30 minutes (default)

Configure settings

Environment settings

Environment type [Info](#)
Choose between creating a new EC2 instance for your new environment or connecting directly to your server over SSH.

☒ **Create a new instance for environment (EC2)**
Launch a new instance in this region to run your new environment.

☐ **Connect and run in remote server (SSH)**
Display instructions to connect remotely over SSH and run your new environment.

Instance type

☒ **t2.micro (1 GiB RAM + 1 vCPU)**
Free-tier eligible. Ideal for educational users and exploration.

☐ **t3.small (2 GiB RAM + 2 vCPU)**
Recommended for small-sized web projects.

☐ **m5.large (8 GiB RAM + 2 vCPU)**
Recommended for production and general-purpose development.

☐ **Other instance type**
Select an instance type.

t3.nano ▼

【テックジム】Railsコース 第4章 「ゼロからアプリケーションを作成しよう」

1回目 : / 分 2回目 : / 分 3回目 : / 分 4回目 : / 分 5回目 : / 分

Platform

☒ Amazon Linux

☐ Ubuntu Server 18.04 LTS

Cost-saving setting

Choose a predetermined amount of time to auto-hibernate your environment and prevent unnecessary charges. We recommend a hibernation settings of half an hour of no activity to maximize savings.

After 30 minutes (default) ▼

IAM role

AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

AWSServiceRoleForAWSCloud9

► Network settings (advanced)

No tags associated with the resource.

Add new tag

You can add 50 more tags.

Cancel Previous step **Next step**

- ・ Step 3 「Review」 では、内容を確認し「Create environment」をクリックして下さい。

【テックジム】 Railsコース 第4章 「ゼロからアプリケーションを作成しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

■ Cloud9でブラウザを立ち上げる

- ・ ページ上部の「Preview」をクリックし、「Preview Running Application」をクリック。



- ・ Cloud9の画面上で、仮想的なブラウザが表示されますので、ブラウザ上部のBrowserの右隣にあるボタンをクリックしてください。すると、新規ブラウザが表示され、bundle exec rails serverで立ち上げたページを表示することができます。



■ Oops VFS connection does not exist と表示された場合

ブラウザが問題を起こしている可能性が高いので、ブラウザを変えていただく(講師はChromeを使用しています)か、シークレットモードで再度AWS・Cloud9にログインしていただけますと、エラーがなくなると思います。

【テックジム】 Railsコース 第4章 「ゼロからアプリケーションを作成しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

■ gitについて

【前提知識】

- ・ 修正： gitではファイルを修正すると、自動で修正部分・新規追加ファイルを認識します。
- ・ コミット： いくつかの修正をひとまとまりにしたものです。
- ・ ブランチ： コミットを順番にまとめたものです。

【コマンド】

ブランチの一覧を表示する。

```
$ git branch
```

特定のブランチ(lesson1)に切り替える

```
$ git checkout lesson1
```

修正・新規ファイルの一覧を表示する。

```
$ git status
```

特定のファイル(app/controllers/contacts_controller.rb)をコミットできる状態にする。

```
$ git add app/controllers/contacts_controller.rb
```

カレントディレクトリ内の全てのファイルをコミットできる状態にする。

```
$ git add .
```

コミットできる状態にした修正・新規ファイルを名前(フォーム送信機能 追加)をつけてコミットする

```
$ git commit -m "フォーム送信機能 追加"
```

コミットを順番に表示する。

```
$ git log
```

特定のファイル(app/controllers/contacts_controller.rb)を修正する前の状態に戻す

```
$ git checkout app/controllers/contacts_controller.rb
```