


【テックジム】 Railsコース 第1章 「問い合わせフォームを作ろう」


■ 1 - 0 : 講座を受ける前に

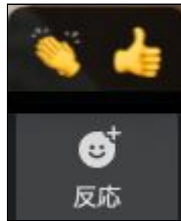
【ZOOMの使い方】

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くださると助かります。
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_course01 (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1章 「問い合わせフォームを作ろう」

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_course01

☆ 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。

■ 1-0-1：実行環境を整えよう(初めての方)

【各種バージョン】

Rails: 5.2.4.2

Ruby: 2.6.3

Linux: Amazon Linux AMI release 2018.03

bundle: 1.17.3

【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
```

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
```

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\_course01.git
```

techgym_rails_course01フォルダに移動する。

```
$ cd techgym_rails_course01
```

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

```
An error occurred while installing pg (1.2.3), and Bundler cannot continue.  
Make sure that `gem install pg -v '1.2.3' --source 'https://rubygems.org/'` succeeds before bundling.
```

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
```

データベースサーバーの起動

```
$ sudo service postgresql start
```

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

【テックジム】Railsコース 第1章 「問い合わせフォームを作ろう」

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

```
FATAL: role "ec2-user" does not exist
Couldn't create 'contact_app_development' database. Please check your configuration.
rake aborted!
```

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
```

作業ブランチを切り替える

```
$ git checkout -b lesson5 remotes/origin/lesson5
```

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、13ページ「Cloud9でブラウザを立ち上げる」をご確認ください。

 お問い合わせ

下記フォームよりお問い合わせください。

件名

サービスについて

氏名

Eメール

内容

送信する

【テックジム】 Railsコース 第1章 「問い合わせフォームを作ろう」

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_course01

☆ 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。

- 1-0-2：実行環境を整えよう(既に環境設定が終了している方)

【手順】

環境設定として下記のコマンド入力してください。

DBサーバーの起動

```
$ sudo service postgresql start
```

techgym_railsフォルダに移動する。

```
$ cd techgym_rails
```

techgym_rails_course01フォルダに移動する。

```
$ cd techgym_rails_course01
```

新しいブランチをダウンロードする。

```
$ git fetch origin
```

前回の修正が残っていた場合の対処(修正が残ってなくても、実行して問題ありません。)

```
$ git add .
```

```
$ git commit -m "Basic認証追加"
```

```
$ git checkout lesson4
```

```
$ git branch -D lesson5
```

```
$ git checkout -b lesson5 remotes/origin/lesson5
```

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
```

Railsのサーバーを起動する。

```
$ bundle exec rails server
```

【テックジム】 Railsコース 第1章 「問い合わせフォームを作ろう」

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

■ 1 - 5 : 入力チェックをしよう : lesson5

【問題】

間違った入力が行われた際に、データが保存されないようにしましょう。

【修正する内容】

ファイル : app/models/contact.rb

追加する機能 : 各カラムのバリデーション

必須 : title, name, email, content, status

適切なフォーマット : email

【実行結果】

間違えた情報を入力し、お問い合わせフォーム「送信する」のボタンをクリック。その後、管理者画面を確認してもデータが作成されていない。

お問い合わせフォーム URL: /

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

管理者画面 URL: /admin/contacts

Basic認証: ユーザー名 techgym, パスワード password

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

【ヒント】

- validates :name, presence: true を入力すると、nameが必須入力の対象になる。
- validates :name, format: { with: /\Atest/} と入力すると、nameの値がtestから始まらなければならなくなる。(/\Atest/ の部分は正規表現と呼ばれる。)
- /\A[w+\.]+@[a-z\d\-.]+\.[a-z]+\z/i はemailの正規表現

【テックジム】 Railsコース 第1章 「問い合わせフォームを作ろう」

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

■ 1 - 6 : エラー表示 : lesson6

【はじめに】

```
$ git add .
```

```
$ git commit -m "バリデーション追加"
```

```
$ git checkout -b lesson6 remotes/origin/lesson6
```

【問題】

間違えたデータが送信された際、エラー文をフォームの上部に表示する。

※ 実際にエラーを表示している部分は、app/views/contacts/new.html.erbの(8~18行目)です。

※ エラーの文章は、config/locales/ja.yml(109~135行目), config/locales/translation_ja.ymlで修正することが出来ます。

【修正する内容】

ファイル : app/controllers/contacts_controller.rb

メソッド : create

追加する機能 : お問い合わせ作成時に、エラーが存在しなかったらサンクスページにリダイレクトされる。エラーが存在したらもう一度同じページを表示した上で、エラーも表示する。

【実行結果】

URL: /contacts/new → 「送信ボタン」をクリック

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

2つのエラーがあります。

- 氏名を入力してください
- Eメールは不正な値です

件名

サービスについて

氏名

Eメール

email

内容

テストです。

送信する

【テックジム】 Railsコース 第1章 「問い合わせフォームを作ろう」

【ヒント】

□ create関数内に下記のように記述し、「送信する」をクリックすると、正しい値の時には、「正しい値です。」と表示され、不正な値の時には、「不正な値です。」と表示されます。

```
@contact = Contact.new(contact_params)
if @contact.save
  render plain: "正しい値です。"
else
  render plain: "不正な値です。"
end
```

□ create関数内に下記のように記述し、「送信する」をクリックすると、エラーメッセージの一覧を表示することが出来ます。@contact.errors.full_messagesは不正な値の時にsaveメソッドを実行すると値が格納されます。

```
@contact = Contact.new(contact_params)
if @contact.save
  render plain: "正しい値です。"
else
  render plain: @contact.errors.full_messages
end
```

□ render :new を実行すると再度お問い合わせフォームが表示されます。

【テックジム】 Railsコース 第1章 「問い合わせフォームを作ろう」

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

■ 1 - 7 : メールを送信しよう : lesson7

【はじめに】

```
$ git add .
```

```
$ git commit -m "エラー文表示"
```

```
$ git checkout -b lesson7 remotes/origin/lesson7
```

【問題】

お問い合わせを作成した時に、メールが送信されるようにしましょう。

また、デフォルトの送信元メールアドレスを変更してください。

※ メールの確認は、letter_opener_webというGemを使用しています。

【実行するコマンド】

```
$ bundle exec rails generate mailer contact send_confirm
```

※ 本プログラムではgenerateのプログラムをカスタマイズしており、実際の挙動とは少し異なります。

【修正する内容】

ファイル : app/controllers/contacts_controller.rb

メソッド : create

ファイル : app/mailers/application_mailer.rb

追加する機能 : メール送信

※ メールの文章を修正するには、app/views/contact_mailer/send_confirm.html.erb,

app/views/contact_mailer/send_confirm.html.erbを修正します。

【実行結果】

URL: /letter_opener

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

Letters

Refresh

Clear

ID

Sent at

Action

1587436180_6061475_3d7ebe4

2020-04-21
02:29:40 +0000

From: noreply@mycontact.techgym

Subject: お問い合わせありがとうございます

Date: Apr 21, 2020 02:29:40 AM UTC

To: test@test.techgym

お問い合わせいただきありがとうございます。
2営業日以内に担当者からご連絡いたします。

お問い合わせ内容

件名： サービスについて

氏名： テスト

内容：
テストです。

※ このメールは、送信専用メールアドレスから配信されています。

【テックジム】 Railsコース 第1章 「問い合わせフォームを作ろう」

【ヒント】

- 修正前は、from@example.comがデフォルトの送信元メールアドレスとして設定されています。
- ContactMailer.send_confirm(@contact) は、Mail::Messageオブジェクトを生成し、@contactに応じたメールを作成します。
- render ContactMailer.send_confirm(@contact)でエンコードされたメール情報を表示することができます。
- send_confirmメソッドはこちらで用意した関数で、app/mailers/contact_mailer.rbに記載されています。
- ※ app/mailers/contact_mailer.rbでは、宛先、件名などを指定しています。
- Mail::Messageオブジェクトは、deliverメソッドを持ち、deliverは対象のメールを送信します。

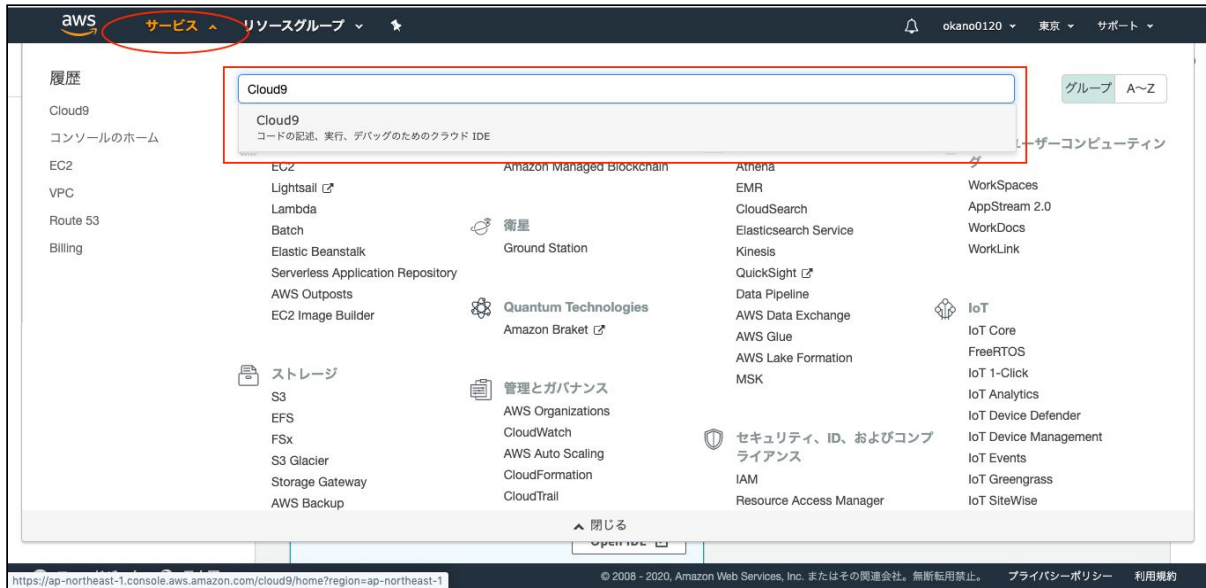
回答はlesson8

【テックジム】Railsコース 第1章 「問い合わせフォームを作ろう」

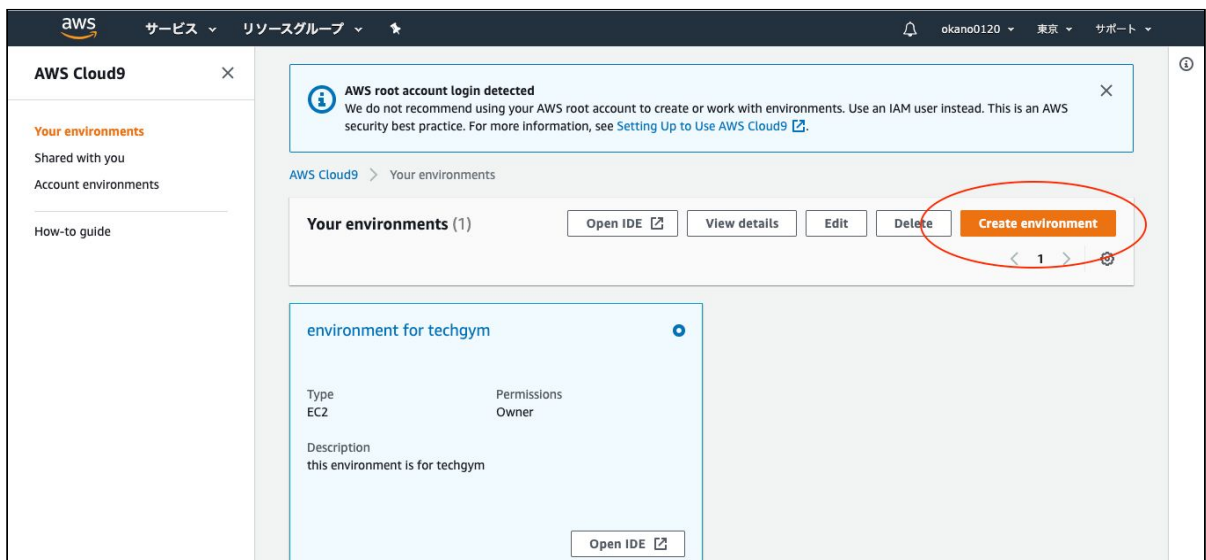
■ Cloud9の立ち上げ方

【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。

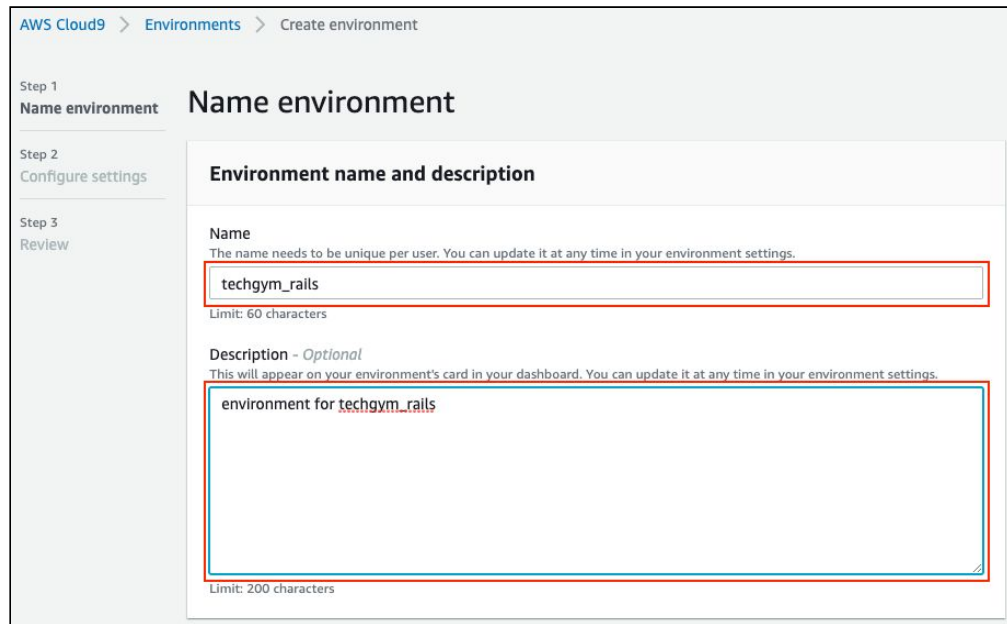


- ・ Cloud9のダッシュボードに移動するので、「Create environment」をクリック



【テックジム】Railsコース 第1章 「問い合わせフォームを作ろう」

- ・ 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

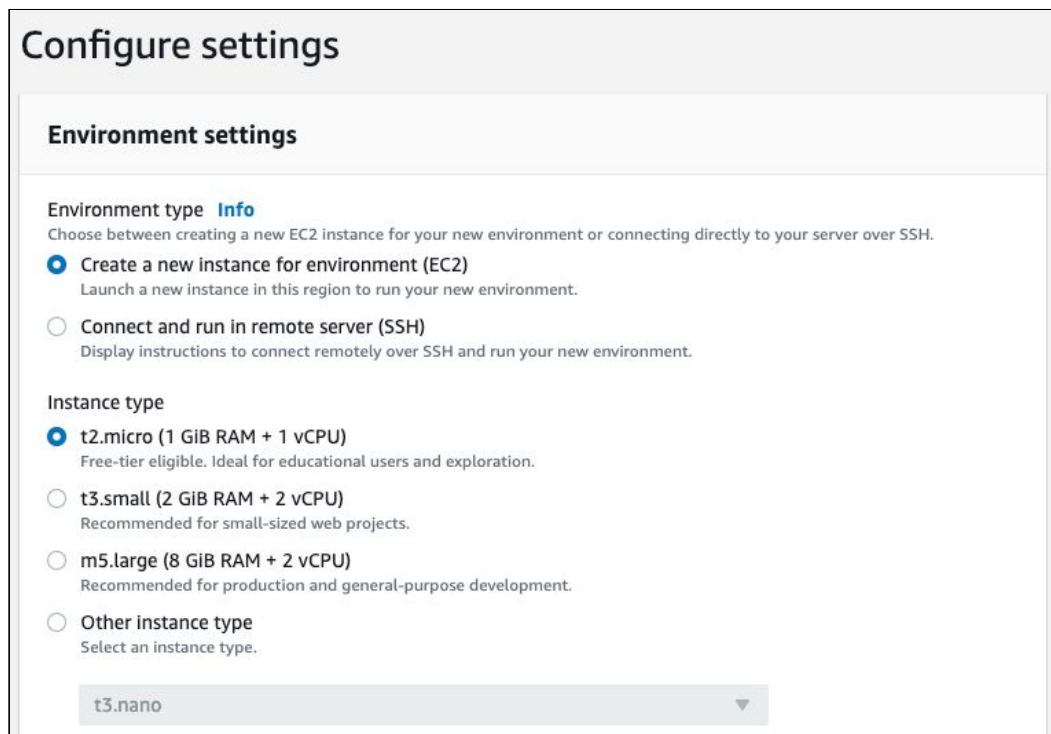
Limit: 60 characters

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1章 「問い合わせフォームを作ろう」

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.

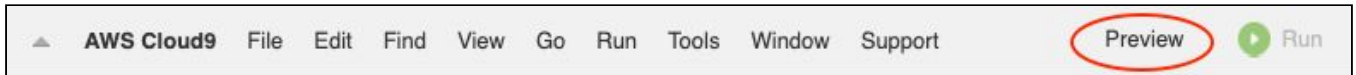
Cancel Previous step **Next step**

- ・ 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1章 「問い合わせフォームを作ろう」

■ 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1章 「問い合わせフォームを作ろう」

■ 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
```