

【テックジム】Railsコース 第3章 「APIを使って読書管理アプリを作ろう」



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

■ 3 - 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3章 「APIを使って読書管理アプリを作ろう」

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_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。

■ 3-0-1：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
```

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_course03.git
```

techgym_rails_course03フォルダに移動する。

```
$ cd techgym_rails_course03
```

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
```

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

データベースサーバーの起動

```
$ sudo service postgresql start
```

【テックジム】 Railsコース 第3章 「APIを使って読書管理アプリを作ろう」

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

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
```

ブランチを切り替える

```
$ git checkout -b lesson1 remotes/origin/lesson1
```

Railsのサーバーを起動する。

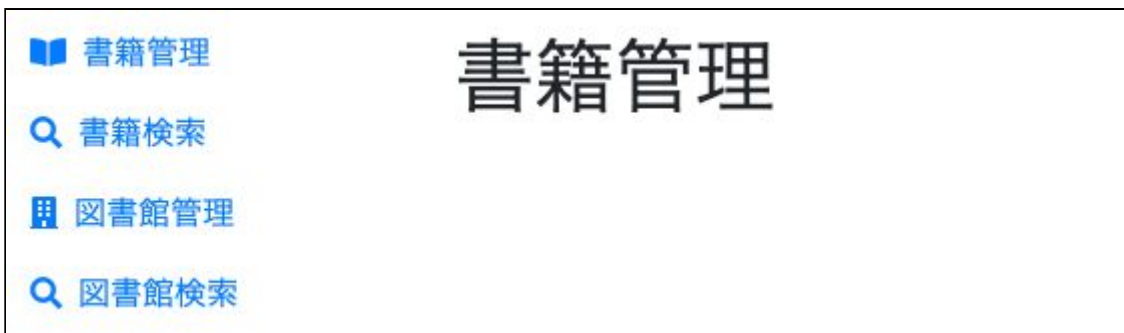
```
$ bundle exec rails server
```

【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3章 「APIを使って読書管理アプリを作ろう」

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

■ 3 - 1 : リクエストを送信しよう : lesson1

【問題】

Googleが提供しているBooks APIにリクエストを送信し、書籍のデータを取得しましょう。

【修正する内容】

ファイル : app/controllers/books/search_controller.rb

修正するアクション : new

修正する内容 :

1. Books APIから取得した情報(文字列)を、JSONオブジェクトに変換し、変数(books_json)に格納する。
 2. books_jsonのキーが"items"である値は配列であり、配列の各要素を一つずつ引数として get_book_from_json関数を実行する。
 3. get_book_from_json関数を実行した時の返り値の配列をインスタンス変数(@books)に格納する。
 4. 確認のため、@booksをrender関数で表示する。
- ※ @booksの要素にnilが存在する可能性があるため、配列からnilを取り除く

【実行結果】

検索ページにアクセス(URL: /books/search/new)

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

検索フォームに「rails」と入力して、「検索」ボタンをクリック

下記のような内容が表示される

```
[{:isbn=>"9784873113869"}, {:isbn=>"9784798162874"}, {:isbn=>"9784873113326"}, {:isbn=>"9784274067853"},  
{:isbn=>"9784798156170"}, {:isbn=>"9784774164786"}, {:isbn=>"9784873113128"}, {:isbn=>"9784798116907"},  
{:isbn=>"9784274068669"}]
```

【ヒント】

- OpenURIモジュールのopen_uri関数の返り値(f)はreadメソッドを持ち、readメソッドは取得した書籍データ(文字列)を返します。
- 文字列をJSONオブジェクトに変換するためには、JSON.load関数を利用します。
JSON.load関数は引数に文字列をとり、JSONオブジェクトを返します。
- JSONオブジェクトから、キーを指定して対応する値を取得したい場合には、下記のように記述します。
json["a"]
(json : JSONオブジェクトが格納されている変数、a : 取得したい値に対応するキー)
- 配列の各要素に対して一つずつ関数を実行し、関数の返り値の配列を取得したい場合には、下記のように記述します。
p ["a", "b", "c"].map { |item| item.upcase }
→ ["A", "B", "C"]
(文字列のメソッド upcaseはローマ字が大文字に変換される)
- 配列からnilである要素を取り除くのは、Arrayオブジェクトのcompactメソッドを利用します。
p ["a", "b", nil, "c"].compact
→ ["a", "b", "c"]
- 値を画面に表示されたい時には、下記のように記述します。
render plain: f.read

【テックジム】 Railsコース 第3章 「APIを使って読書管理アプリを作ろう」

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

■ 3 - 2 : Jsonから必要な情報を取得しよう : lesson2

【はじめに】

\$ git add .

\$ git commit -m "リクエスト送信"

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

【問題】

取得したJSONオブジェクトから必要な値のみを取得し、変数(book)に格納しましょう。

取得する値のキーと、bookに格納する際のキーは下記のようにしてください。

値が存在しない場合には、nilを返してください。

※ 3 - 1で追加した、render関数は削除してください。

取得する値のキー	bookに格納する際のキー
"volumeInfo", "title"	"title"
"volumeInfo", "previewLink"	"url"
"volumeInfo", "description"	"description"
"volumeInfo", "imageLinks", "thumbnail"	"thumbnail_url"

【修正する内容】

ファイル : app/controllers/books/search_controller.rb

修正する関数 : get_book_from_json

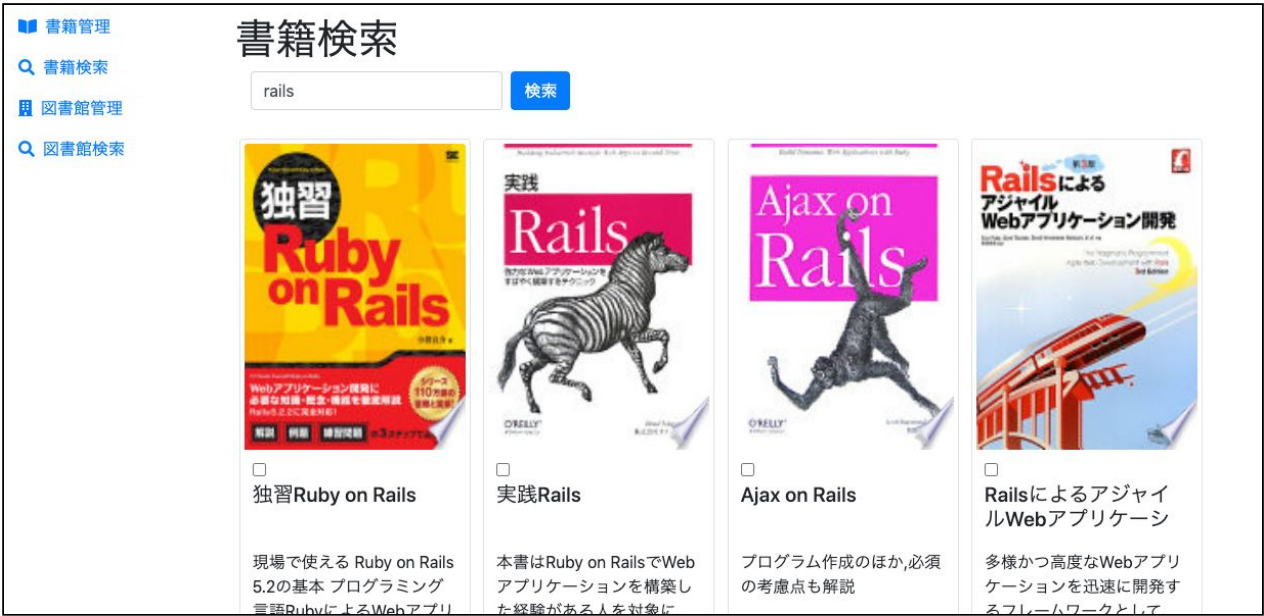
修正する内容 : JSONオブジェクトから必要な値を取得し、bookに適切なキーを指定して値を格納する。

【実行結果】

検索ページにアクセス(URL: /books/search/new)

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

検索フォームに「rails」と入力して、「検索」ボタンをクリック



【テックジム】 Railsコース 第3章 「APIを使って読書管理アプリを作ろう」

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

【ヒント】

- 変数にキーを指定して、値を格納する際には、下記のように記述する。

```
hash = {}  
hash["a"] = 1  
p hash  
→ {"a"=>1}
```

- JSONオブジェクトからキーを指定して値を取得する時、キーが本当に存在するかどうか分からない場合には、digメソッドを使用して値を取得する。digメソッドは指定したキーに対応する値が存在しなかった時にnilを返します。

```
hash = {"a1" => {"a2" => 1}}  
p hash.dig("a1", "a2")  
→ 1  
p hash.dig("a", "a2")  
→ nil
```

```
p hash["a1"]["a2"]  
→ 1  
p hash["a"]["a2"]  
→ error
```

【テックジム】 Railsコース 第3章 「APIを使って読書管理アプリを作ろう」

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

■ 3 - 3 : フォームを正しく修正しよう : lesson3

【はじめに】

```
$ git add .
```

```
$ git commit -m "Jsonから必要な情報を取得"
```

```
$ git checkout -b lesson3 remotes/origin/lesson3
```

【問題】

現状のフォームではformの送信先が指定されておらず、必要な情報が送信されていません。

フォームを修正し正しい情報が送信されるようにしましょう。

【修正する内容】

ファイル : app/views/books/search/new.html.erb

修正する内容 :

formの送信先を指定、必要なフィールドを追加する。

送信先は、books/searchコントローラーのcreateアクションに指定する。

ファイル : app/controllers/books/search_controller.rb

修正するアクション : create

修正する内容 : 送信したパラメータから必要なデータのみ表示する。

【実行結果】

検索ページにアクセス(URL: /books/search/new)

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

検索フォームに「rails」と入力して、「検索」ボタンをクリック

いくつかのチェックボックスにチェックを入れて、「保存」ボタンをクリック

必要な情報が全て表示されているかどうかを確認する

```
{"books"=><ActionController::Parameters {"0"=><ActionController::Parameters {"book"=><ActionController::Parameters {"isbn"=>"97848731113869", "thumbnail_url"=>"http://books.google.com/books/content?id=-7iMOLUQDqQC&printsec=frontcover&img=1&zoom=1&edge=cur&source=gbs_api", "title"=>"実践Rails", "description"=>"本書はRuby on RailsでWebアプリケーションを構築した経験がある人を対象に、Railsの実用テクニックを解説する。Rubyのメタプログラミング機能やRailsの内部構造に始まり、データベースやセキュリティ、パフォーマンス、RESTfulアーキテクチャなどについての実践的なテクニックに至るまでを詳述し、その上で大規模なRailsプロジェクトの管理および他のシステムとのRailsの統合についてを説明する。また、Railsの開発に参加する方法や、Railsを構成するコンポーネントの代替となるソリューション、アプリケーションの国際化などについても扱っている。実践的にRailsを使いたいユーザ必携の一冊である。Rails 2.0対応。Rails 2.1プラグイン環境についての記述を付録として収録。", "url"=>"http://books.google.co.jp/books?id=-7iMOLUQDqQC&pg=PA226&dq=rails&hl=&cd=1&source=gbs_api"} permitted: true>} permitted: true>, "1"=><ActionController::Parameters {"is_create"=>"1", "book"=><ActionController::Parameters {"isbn"=>"97848731113326", "thumbnail_url"=>"http://books.google.com/books/content?id=r4NQ2UjeBt8C&printsec=frontcover&img=1&zoom=1&edge=cur&source=gbs_api", "title"=>"Ajax on Rails", "description"=>"プログラム作成のほか、必須の考慮点も解説", "url"=>"http://books.google.co.jp/books?id=r4NQ2UjeBt8C&pg=PA180&dq=rails&hl=&cd=2&source=gbs_api"} permitted: true>} permitted: true>, "2"=><ActionController::Parameters {"book"=><ActionController::Parameters {"isbn"=>"9784274067853", "thumbnail_url"=>"http://books.google.com/books/content?id=Aof6YVzLEfYC&printsec=frontcover&img=1&zoom=1&edge=cur&source=gbs_api", "title"=>"RailsによるアジャイルWebアプリケーション開
```


【テックジム】 Railsコース 第3章 「APIを使って読書管理アプリを作ろう」

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

【ヒント】

□ formの送信先を指定するためには、下記のようにパスを指定します。

```
<%= form_with url: new_books_search_path, local: true do |f| %>...<% end %>
```

□ 送信先のパスを確認するためには、/rails/info/routesにアクセスし、Helperの欄を確認します。

ex)

<https://f24e3029423e4xxxxxx38c8888d4.vfs.cloud9.ap-northeast-1.amazonaws.com/rails/info/routes>

□ HTMLのinputタグでtype=hiddenを指定すると、見た目に表示されないデータをformで送信することができます。

□ Railsではinputタグ(type=hidden)を記述するためのビューヘルパー(hidden_field_tag)が用意されています。

ex) <%= hidden_field_tag "books[1][book][isbn]", book[:isbn] %>

<%= hidden_field_tag "{formで送信するデータの構造}", 送信する値 %>

□ 送信されたパラメータから必要なデータのみを取得するには、ストロングパラメータを利用します。

books/searchコントローラーの場合、books_params関数を実行することでストロングパラメータを利用できます。

【テックジム】 Railsコース 第3章 「APIを使って読書管理アプリを作ろう」

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

■ 3 - 4 : データを保存しよう : lesson4

【はじめに】

```
$ git add .
```

```
$ git commit -m "フォーム修正"
```

```
$ git checkout -b lesson4 remotes/origin/lesson4
```

【問題】

3 - 3で受け取ったパラメータを受け取り、パラメータ内のis_createに対応する値を元に、is_createが含まれる書籍を全て保存しましょう。

【修正する内容】

ファイル : app/controllers/books/search_controller.rb

修正するアクション : create

修正する内容 :

パラメータの表示を削除

受け取ったパラメータから、書籍データの配列を取得し、is_createの値に応じてデータベースに保存する。

書籍一覧ページに通知の文言「書籍を追加しました」を指定しリダイレクトを行う。

※ 初期一覧ページは、booksコントローラーのindexアクションにて表示される。

【実行結果】

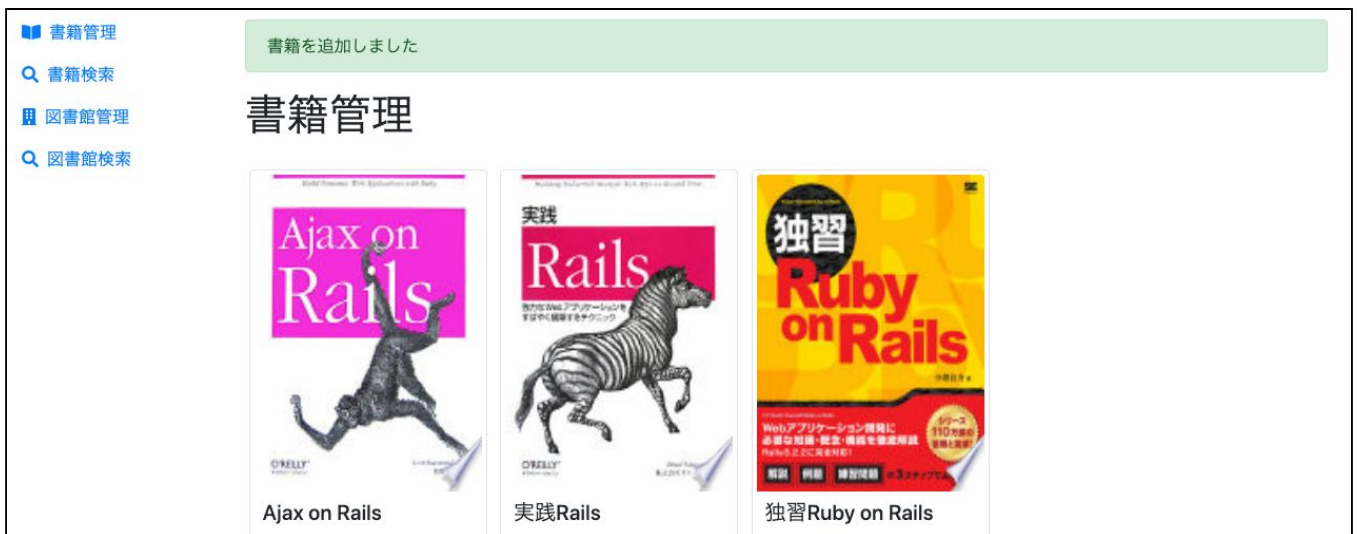
検索ページにアクセス(URL: /books/search/new)

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

検索フォームに「rails」と入力して、「検索」ボタンをクリック

いくつかのチェックボックスにチェックを入れて、「保存」ボタンをクリック

チェックを入れた書籍が追加されているかを確認する



【テックジム】 Railsコース 第3章 「APIを使って読書管理アプリを作ろう」

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

【ヒント】

□ 下記のように記述することで、書籍データの配列を取得することができます。

```
books_params[:books].values
```

□ 配列の各要素を元に繰り返し処理を行いたい場合には、eachメソッドを利用します。

ex)

```
[1, 2, 3].each do |num|
```

```
  p num + 1
```

```
end
```

```
→
```

```
2
```

```
3
```

```
4
```

□ 下記のように記述することで、保存する書籍の場合は1, 保存しない場合にはnilを取得することができる。

```
books_params[:books].values[0][:is_create]
```

□ 下記のように記述することで、書籍モデルのカラムに沿ったハッシュを取得することができる。

```
books_params[:books].values[0][:book]
```

□ リダイレクト先のパスを指定してリダイレクトさせる場合redirect_to関数を利用する。

また、リダイレクト先にデータ(notice)を渡したい場合には下記のように記述する。

```
redirect_to books_path, notice: "#{Book.model_name.human}を追加しました"
```

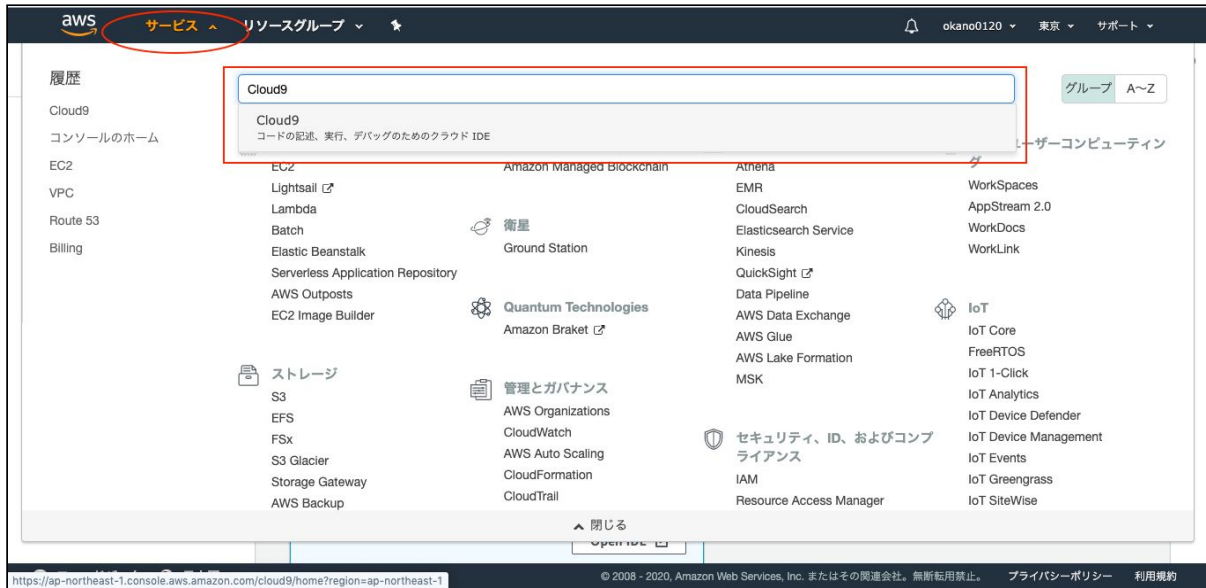
【テックジム】Railsコース 第3章 「APIを使って読書管理アプリを作ろう」

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

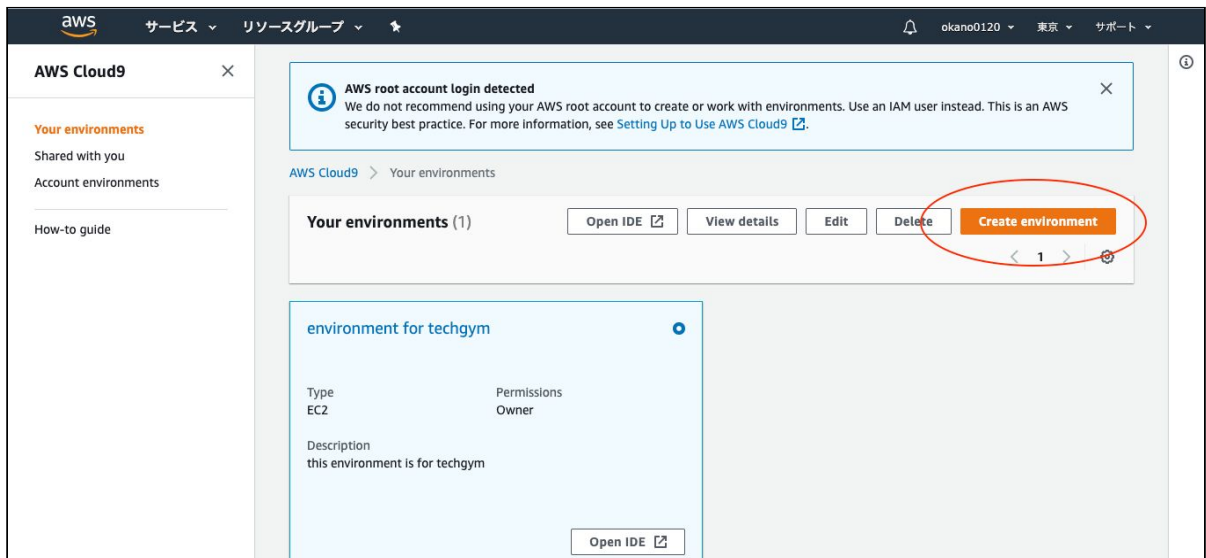
■ Cloud9の立ち上げ方

【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3章 「APIを使って読書管理アプリを作ろう」

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3章 「APIを使って読書管理アプリを作ろう」

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3章 「APIを使って読書管理アプリを作ろう」

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3章 「APIを使って読書管理アプリを作ろう」

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
```