



【テックジム】 Railsコース 第2章 「ウェブサイトから情報を取得しよう」

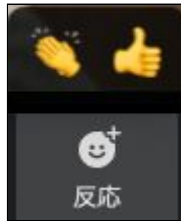
■ 2 - 0 : 講座を受ける前に

【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_course02 (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2章 「ウェブサイトから情報を取得しよう」

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_course02

☆ 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。

■ 2 - 0 : 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\_course02.git
```

techgym_rails_course02フォルダに移動する。

```
$ cd techgym_rails_course02
```

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2章 「ウェブサイトから情報を取得しよう」

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
```

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。

案件管理

ID	タイプ	タイトル	報酬(最小)	報酬(最大)	期限日
----	-----	------	--------	--------	-----

新規作成

【テックジム】 Railsコース 第2章 「ウェブサイトから情報を取得しよう」

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

■ 2 - 1 : タスクを作成しよう : lesson1

【はじめに】

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

【問題】

スクレイピングを実行するためのタスクを作成して、"スクレイピングを開始しました。"と表示させましょう

【実行するコマンド】

タスクファイルの作成

```
$ bundle exec rails generate task scraping fetch_crowdworks
```

【修正する内容】

ファイル : lib/tasks/scraping.rake

実行する内容 : "スクレイピングを開始しました。"の表示

【実行結果】

タスクが正常に設定されているか確認

```
$ bundle exec rake -T scraping
```

```
→ rake scraping:fetch_crowdworks # クラウドワークスの情報取得
```

タスクの実行

```
$ bundle exec rake scraping:fetch_crowdworks
```

```
→ "スクレイピングを開始しました。"
```

【ヒント】

□ task fetch_crowdworks: :environment do ~ endの中に実行したい処理を記述する

□ p関数は文字列を引数として渡すと、渡された引数を表示する。

例) p "a"

```
→ "a"
```

□ descはtaskの直上で文字列を引数にとり実行することで、タスクの説明を設定することができる。(デフォルトは "TODO")

例) desc "タスクに関する説明"

【テックジム】 Railsコース 第2章 「ウェブサイトから情報を取得しよう」

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

■ 2 - 2 : モジュールを使おう : lesson2

【はじめに】

```
$ git add .  
$ git commit -m "スクレイピング用タスク追加"  
$ git checkout -b lesson2 remotes/origin/lesson2
```

【問題】

スクレイピングを行う上で必要な関数をモジュール化していますので、一度モジュールを呼び出してみましょう。
該当モジュールのパスは、lib/scraping_work.rb

【修正する内容】

ファイル : lib/tasks/scraping.rake
修正するタスク : fetch_crowdworks
実行する内容 : モジュールのクラス変数(sample_function)をタスクから呼び出す。

【実行結果】

タスクの実行
\$ bundle exec rake scraping:fetch_crowdworks
→
"スクレイピングを開始しました."
"モジュールが正しく読み込まれています。"

【ヒント】

- ファイルの一番上部に require “ファイルパス” 記述すると、該当モジュールを扱うことができる。(拡張子は省略することが可能)
- Rails.rootには、アプリケーションのルートパスが格納されている。
- 該当モジュールの絶対パスは、下記のように表現することができる。
"#{Rails.root}/lib/scraping_work"
- モジュール内のクラス変数は、モジュール名.関数名で実行することができる。
例) ScrapingWork.function_name

【テックジム】 Railsコース 第2章 「ウェブサイトから情報を取得しよう」

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

■ 2 - 3 : タイトルデータを取得しよう : lesson3

【はじめに】

```
$ git add .  
$ git commit -m "モジュール追加"  
$ git checkout -b lesson3 remotes/origin/lesson3
```

【問題】

実際のクラウドワークスの案件からタイトルデータを取得しましょう。
取得するページの例)

<https://crowdworks.jp/public/jobs/1035766>

【修正する内容】

ファイル : lib/tasks/scraping.rake

修正するタスク : fetch_crowdworks

実行する内容 : URLを指定して該当ページのHTMLを取得し、タイトルデータを表示しましょう。

使用するGem :

HTMLを取得 : OpenURI

タイトルデータ取得 : Nokogiri

【実行結果】

タスクの実行

```
$ bundle exec rake scraping:fetch_crowdworks
```

→

"スクレイピングを開始しました。"

"モジュールが正しく読み込まれています。"

"Ruby on Railsでwebクローラーを作成してください。"

【ヒント】

□ HTML取得はScrapingWorkモジュール内のget_work_docメソッドを利用しましょう。

□ get_work_docメソッドは引数としてURL(文字列)をとり、該当のHTMLをNokogiriのオブジェクトに変換した値を返します。

□ Nokogiriオブジェクトにはatメソッドが定義されており、atメソッドは引数にXPath(文字列)をとり、XPathに応じたオブジェクトを返します。XPathの末尾が"text()"である時に、返されたオブジェクトはtextメソッドを持ち、textメソッドは該当するテキストを返します。

□ 文字列の末尾に余計な文字(改行・空白 等)が入ることがあるので、文字列に定義されているstripメソッドを実行すると、末尾の余計な文字を削除することができます。

【テックジム】 Railsコース 第2章 「ウェブサイトから情報を取得しよう」

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

■ 2 - 4 データを保存しよう : lesson4

【はじめに】

```
$ git add .  
$ git commit -m "HTML取得機能追加"  
$ git checkout -b lesson4 remotes/origin/lesson4
```

【問題】

今回取得したい全ての情報を取得し、データベースに保存しましょう。
取得する情報： タイトル、報酬(最小)、報酬(最大)、詳細、期限日

【修正する内容】

ファイル : lib/tasks/scraping.rake
修正するタスク : fetch_crowdworks
実行する内容 : 該当ページからデータを取得し、データベースに保存する。

ファイル : lib/scraping_work.rb
修正する関数 : detail
実行する内容 : 該当ページから詳細を取得したテキストを返す。

【実行結果】

タスクの実行
\$ bundle exec rake scraping:fetch_crowdworks

案件管理画面にアクセス(URL: /)

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

案件管理

ID	タイプ	タイトル	報酬(最小)	報酬(最大)	期限日	
1	クラウドワークス	Ruby on Railsでweb...	16200		2017-01-27	<button>表示</button> <button>編集</button> <button>削除</button>

新規作成

【ヒント】

- 全ての情報を取得する時には、ScrapingWorkモジュールのfetch_workメソッドを利用しましょう。
fetch_workメソッドは、第一引数に、URL(文字列)、第二引数にdoc(Nokogiriオブジェクト)をとります。
- detail関数をデバッグしたい場合には、タスク内でScrapingWork.detail(doc)を実行しましょう。
- fetch_workメソッドは、Workモデルのカラムに対応したハッシュを返します。
- Workモデルのcreate!メソッドは、引数に与えられたハッシュをデータベースに保存します。
create!メソッドは、第一引数にハッシュをとります。

【テックジム】Railsコース 第2章 「ウェブサイトから情報を取得しよう」

■ XPath取得方法

ブラウザ：Chrome

PC：Mac

対象のページの適当な部分で、右クリックを行うと、下記（左）のようにポップアップが出現します。ポップアップの「検証」をクリックすると、下記（右）のエリアが出現します。

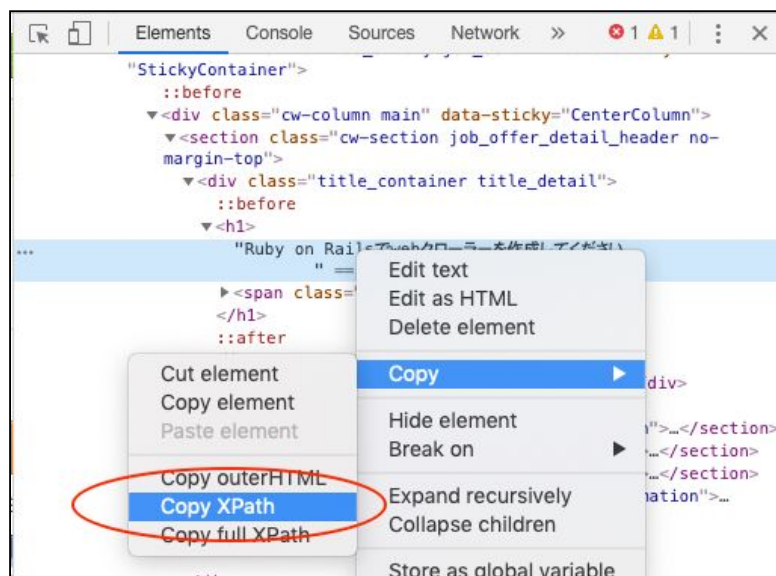


次に、出現したエリアの左上のアイコンをクリックした状態で、XPathを取得したい要素付近をクリックします。



すると、該当部分のHTMLが表示されますので、実際に取得する部分をクリックして、クリックした行の上で右クリックをすると、下記画像のポップアップが出現します。

「Copy」 → [Copy XPath]を選択するとクリップボードに該当要素のXPathがコピーされます。

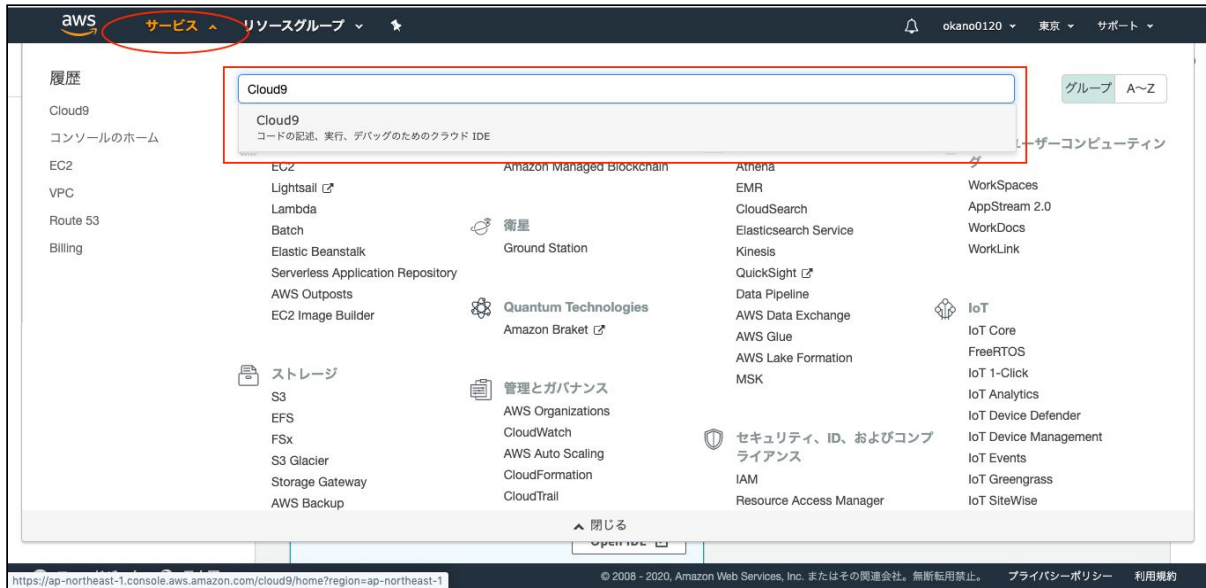


【テックジム】 Railsコース 第2章 「ウェブサイトから情報を取得しよう」

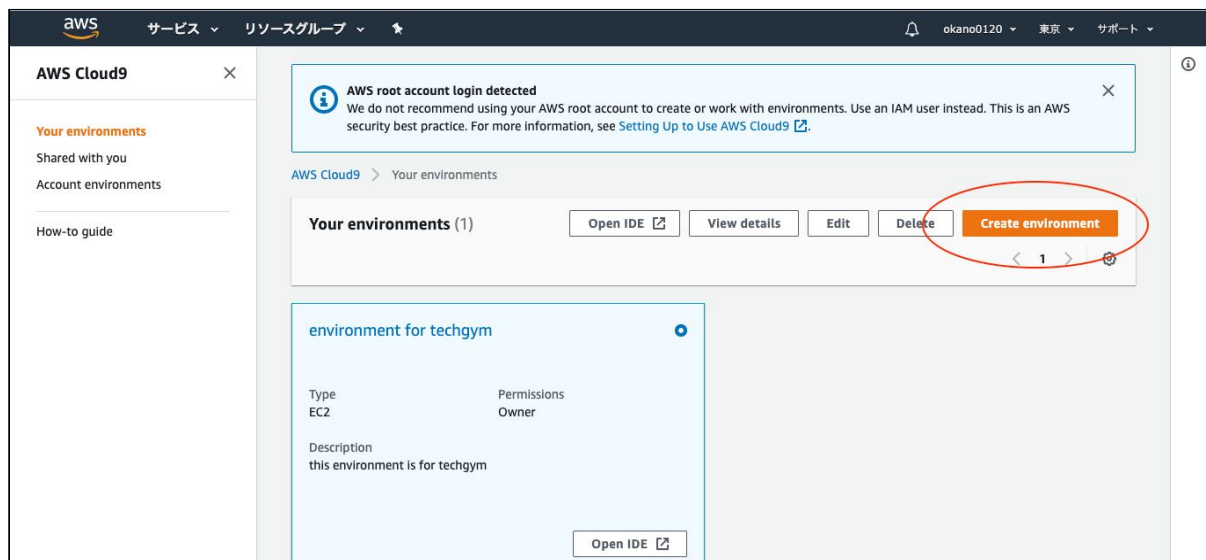
■ Cloud9の立ち上げ方

【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2章 「ウェブサイトから情報を取得しよう」

- ・ 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2章 「ウェブサイトから情報を取得しよう」

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.

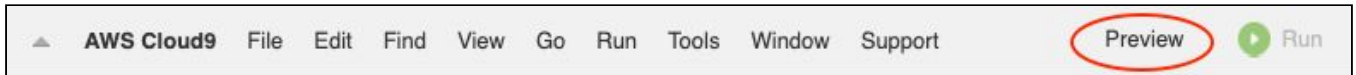
Cancel Previous step **Next step**

- ・ 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2章 「ウェブサイトから情報を取得しよう」

■ 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くと、エラーがなくなると思います。

【テックジム】 Railsコース 第2章 「ウェブサイトから情報を取得しよう」

■ 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
```