Rails API

Seção 1

General information

Create new project

```
rails new nomeDoProjeto --api -d mysql
```

Install dependencies

```
bundle install
```

Criar o banco

```
rails db:create
```

Atualizar o banco

```
rails db:migrate
```

Gerar um scaffold

```
rails g scaffold Contact name:string email:string birthdate:date
```

Ver as tasks

```
rails -T
```

Executa no terminal um código ruby

```
%x(rails db:drop db:create db:migrate)
```

Resources

complete crud

```
resources :contacts
```

custom route in **routes.rb**

```
get '/contacts', to: "contacts#index"
```

only some routes

```
resources :contacts, only: [:new, :create, :destroy]
```

HTTP status codes

Pegar os nomes dos status code no link:

- https://httpstatuses.com/

Exemplo de mudança do code status no rails:

```
render json: @contacts, status: :multi_status
```

Render JSON

only some informations

```
render json: @contacts, only: [:name, :email]
```

except some informations

```
render json: @contacts, except: [:id, :email]
```

Associações

Cria a tabela de referência

```
rails g scaffold Kind description:string
```

Adiciona a coluna que referencia na tabela

```
rails g migration add_kind_to_contact kind:references
```

Adiciona o belongs_to no arquivo que tem a coluna (contacts)

```
class Contact < ApplicationRecord
  belongs_to :kind
end
```

Reinicia o banco

```
rails db:drop
rails db:create
rails db:migrate
rails dev:setup
```

Para listar a tabela associada, reescreve o as_json no model

```
def as_json(options={})
  super(
    include: { kind: { only: :description } }
  )
end
```

```
def as_json(options={})
  super(
    include: :kind
  )
end
```

```
def kind_description
  self.kind.description
end

def as_json(params={})
  super(
    methods: [:kind_description]
  )
end
```

Ou no controller

```
render json: @contacts.as_json(include: :kind)
```

- Por padrão as associações com belongs_to são obrigatórias. Para tornar opcional, é possível adicionar no model:

  ```
  belongs_to :kind, optional: true
  ```

Tradução

Adicionar no gemfile o i18n

```
gem 'rails-i18n', '~> 5.1'
```

Adicionar a tradução no arquivo de configuração:

/config/application.rb

```
# Traduções e tals em português
config.i18n.default_locale = "pt-BR"
```

Associação has many

Cria o model

```
rails g model Phone number:string contact:references
```

Na tabela principal (contacts)

```
has_many :phones
```

Para listar no controller

```
render json: @contacts, include: [:kind, :phones], root: true
```

Nested Atributes

- Gera um model igual o exemplo acima
- Usado para o has_many, has one...

Exemplo

```
class Member < ActiveRecord::Base
  has_many :posts
  accepts_nested_attributes_for :posts
end
```

Usar esse code no model para aceitar nested attributes

```
accepts_nested_attributes_for :phones
```

- CREATE

Adicionar suporte no controller, note que foi adicionado o phone attributes como array

```
# Only allow a trusted parameter "white list" through.
def contact_params
  params.require(:contact).permit(
    :name, :email, :birthdate, :kind_id,
    phones_attributes: [:number]
  )
end
```

Para fazer a requisição create no posts:

```
{
    "contact": {
        "name": "Leonardo Rocha",
        "email": "lerooks.nick@gmail.com",
        "birthdate": "1999-09-24",
        "kind_id": 2,
        "phones_attributes": [
            {
                "number": "(41) 99778-0579"
            },
            {
                "number": "(41) 3049-4455"
            }
        ]
    }
}
```

- READ

Usar o include

```
render json: @contact, include: [:kind, :phones]
```

- UPDATE

Adicionar suporte ao id nos parametros de phones_attributes

```
# Only allow a trusted parameter "white list" through.
def contact_params
  params.require(:contact).permit(
    :name, :email, :birthdate, :kind_id,
    phones_attributes: [:number, :id]
  )
end
```

Fazer um patch com um exemplo de json abaixo:

```
{
    "contact": {
        "name": "Leonardo Roberto Rocha",
        "email": "lerooks.nick@gmail.com",
        "birthdate": "1999-09-24",
        "phones_attributes": [
            {
                "id": 214,
                "number": "(41) 99987-2850"
            }
        ]
    }
}
```

- DELETE

Adicionar o allow_destroy no model

```
accepts_nested_attributes_for :phones, allow_destroy: true
```

Adiciona suporte ao parâmetro no controller

```
# Only allow a trusted parameter "white list" through.
def contact_params
  params.require(:contact).permit(
    :name, :email, :birthdate, :kind_id,
    phones_attributes: [:number, :id, :_destroy]
  )
end
```

Faz um próprio patch (update)

```
{
    "contact": {
        "name": "Leonardo Roberto Rocha",
        "email": "lerooks.nick@gmail.com",
        "birthdate": "1999-09-24",
        "phones_attributes": [
            {
                "id": 214,
                "_destroy": 1
            }
        ]
    }
}
```

has_one

Gera o model

```
rails g model Address street:string city:string contact:references
```

No contact model

```
has_one :address
accepts_nested_attributes_for :address
```

- CREATE

Liberar o parametro:

```
# Only allow a trusted parameter "white list" through.
def contact_params
  params.require(:contact).permit(
    :name, :email, :birthdate, :kind_id,
    phones_attributes: [:number, :id, :_destroy]
    address_attributes: [:id,  :street, :city]
  )
end
```

Faz o post

```
{
    "contact": {
        "name": "Leonardo Rocha",
        "email": "lerooks.nick@gmail.com",
        "birthdate": "1999-09-24",
        "kind_id": 2,
        "phones_attributes": [
            {

                "number": "(41) 99778-0579"
            },
            {
                "number": "(41) 3049-4455"
            }
        ],
       "address_attributes": {
            "city": "Curitiba",
            "street": "Santa Maria"
       }
    }
}
```

- READ

Usa o include no render json

```
render json: @contact, include: [:kind, :phones, :address]
```

- UPDATE

Usa o json com patch

```
{
    "contact": {
        "name": "Leonardo Rocha",
        "email": "lerooks.nick@gmail.com",
        "birthdate": "1999-09-24",
        "kind_id": 2,
       "address_attributes": {
            "city": "Curitiba",
            "street": "Santa Maria 164"
       }
    }
}
```

- DELETE

CORS

Usar o CORS para liberar acesso externo da api. (Aula 28, Seção 1)