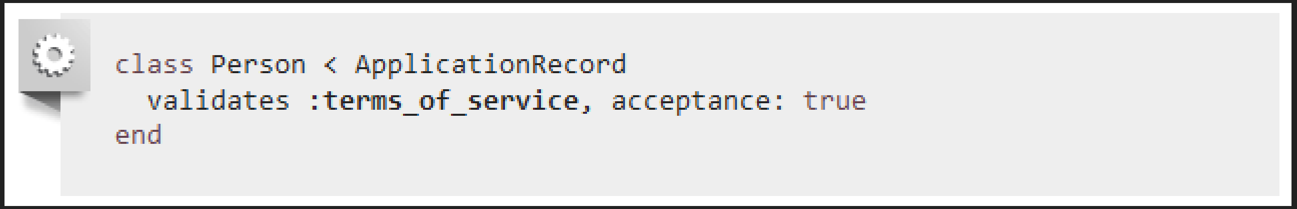
Rails – Parte II

Tópicos

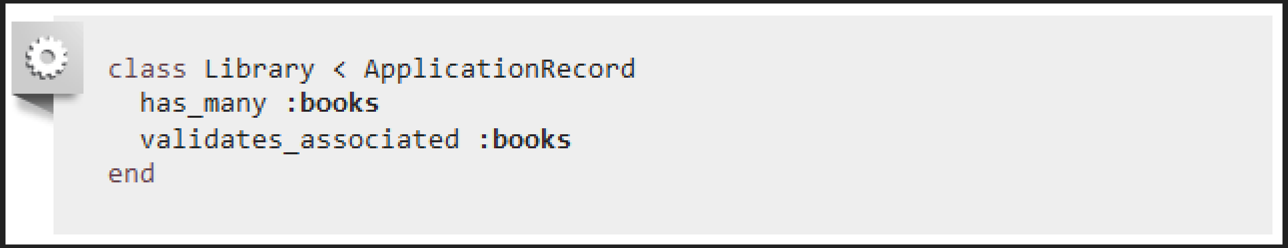
- Validações
- Senha segura
- Sessões
- Filtros

Validações

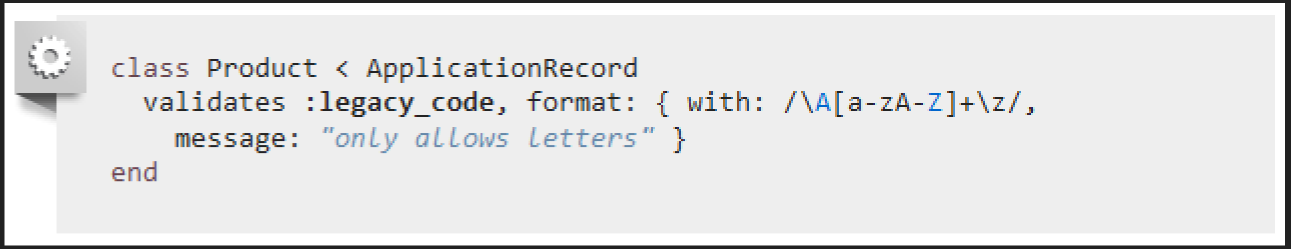
Exemplos



```ruby
class Person < ApplicationRecord
  validates :terms_of_service, acceptance: true
end
```

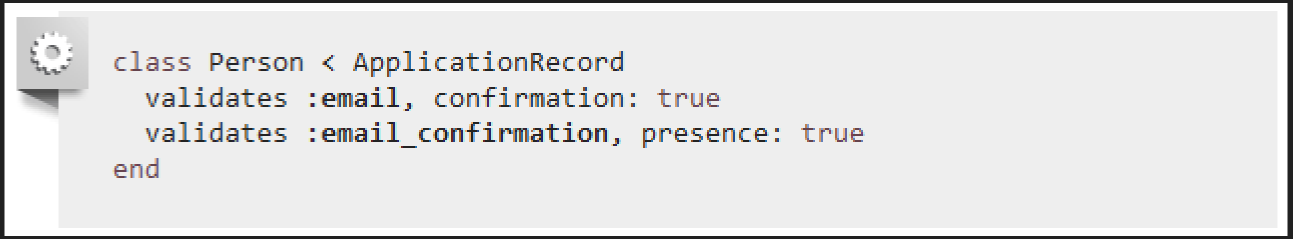


```ruby
class Library < ApplicationRecord
  has_many :books
  validates_associated :books
end
```

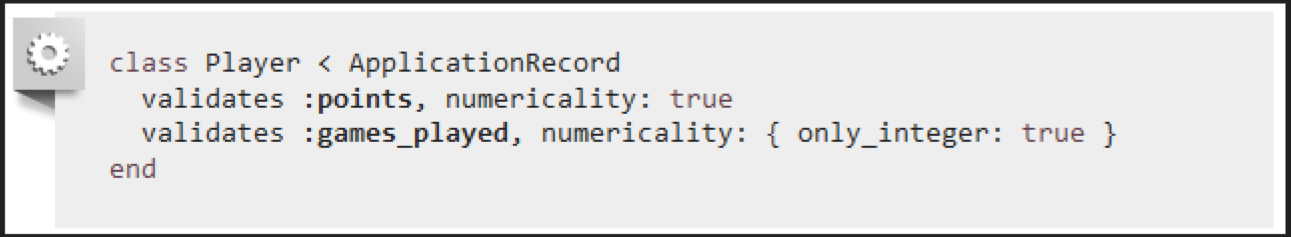


```ruby
class Product < ApplicationRecord
  validates :legacy_code, format: { with: /\A[a-zA-Z]+\z/,
    message: "only allows letters" }
end
```

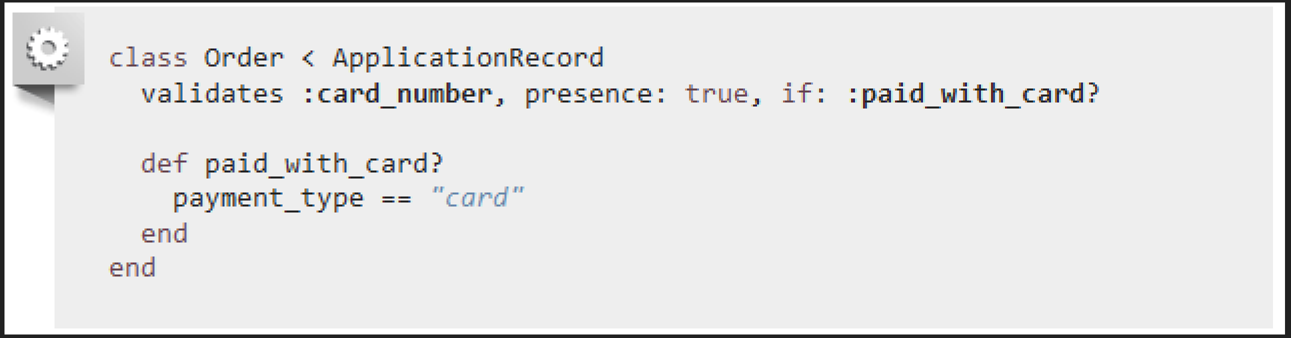
Exemplos



```ruby
class Person < ApplicationRecord
  validates :email, confirmation: true
  validates :email_confirmation, presence: true
end
```

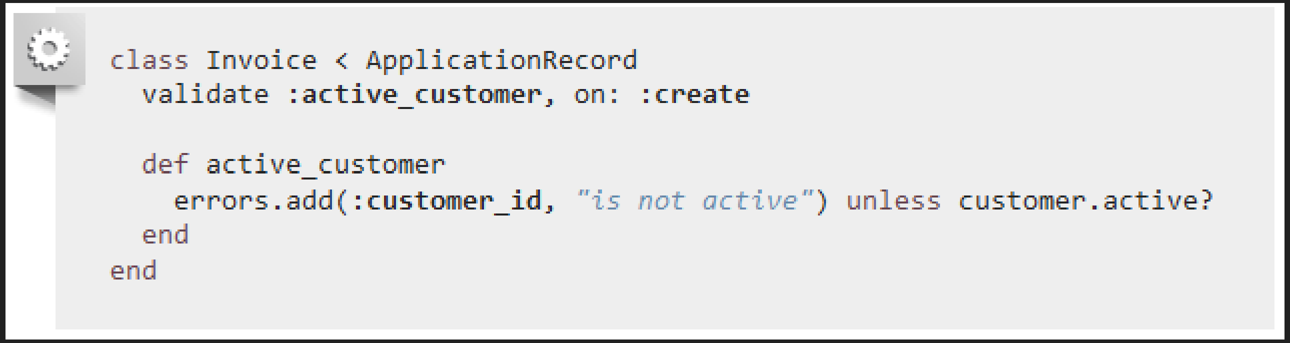


```ruby
class Player < ApplicationRecord
  validates :points, numericality: true
  validates :games_played, numericality: { only_integer: true }
end
```



```ruby
class Order < ApplicationRecord
  validates :card_number, presence: true, if: :paid_with_card?

  def paid_with_card?
    payment_type == "card"
  end
end
```

Custom Validation

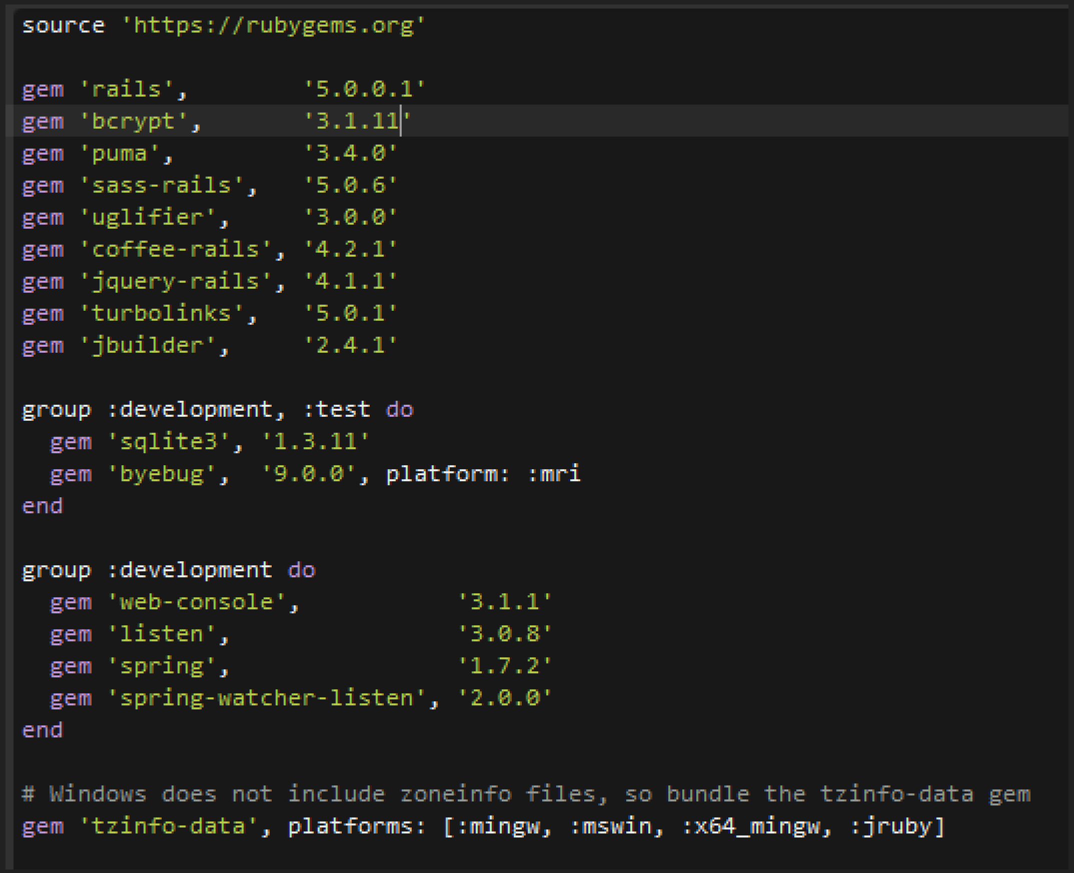


```ruby
class Invoice < ApplicationRecord
  validate :active_customer, on: :create

  def active_customer
    errors.add(:customer_id, "is not active") unless customer.active?
  end
end
```

Senha Segura

Instalar gem bcrypt

- Adicionar ao gemfile: gem ‘bcrypt’, ‘3.1.11’
- bundle install

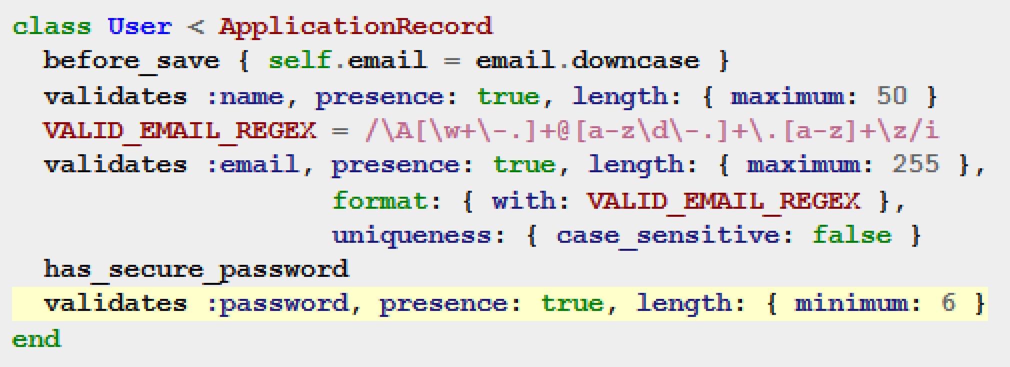


```ruby
source 'https://rubygems.org'

gem 'rails',        '5.0.0.1'
gem 'bcrypt',       '3.1.11'
gem 'puma',         '3.4.0'
gem 'sass-rails',   '5.0.6'
gem 'uglifier',     '3.0.0'
gem 'coffee-rails', '4.2.1'
gem 'jquery-rails', '4.1.1'
gem 'turbolinks',   '5.0.1'
gem 'jbuilder',     '2.4.1'

group :development, :test do
  gem 'sqlite3', '1.3.11'
  gem 'byebug',  '9.0.0', platform: :mri
end

group :development do
  gem 'web-console',           '3.1.1'
  gem 'listen',                '3.0.8'
  gem 'spring',                '1.7.2'
  gem 'spring-watcher-listen', '2.0.0'
end

# Windows does not include zoneinfo files, so bundle the tzinfo-data gem
gem 'tzinfo-data', platforms: [:mingw, :mswin, :x64_mingw, :jruby]
```

Adicionar digest a Users

- rails g migration AddPasswordDigestToUsers password_digest:string
- rake db:migrate
- Adicionar no modelo de User: has_secure_password

User com validações



```
class User < ApplicationRecord
  before_save { self.email = email.downcase }
  validates :name, presence: true, length: { maximum: 50 }
  VALID_EMAIL_REGEX = /\A[\w+\-.]+@[a-z\d\-.]+\.[a-z]+\z/i
  validates :email, presence: true, length: { maximum: 255 },
                    format: { with: VALID_EMAIL_REGEX },
                    uniqueness: { case_sensitive: false }
  has_secure_password
  validates :password, presence: true, length: { minimum: 6 }
end
```

Sessões

Criando um controller

- rails g controller Sessions new
- Não é RESTful, então adicionamos em routes.rb:

```
get     '/login',   to: 'sessions#new‘
post    ‘/login’,   to: ‘sessions#create’
delete ‘/logout’, to: ‘sessions#destroy’
```

Formulário de login

```
<%= form_for(:session, url: login_path) do |f| %>

  <%= f.label :email %>
  <%= f.email_field :email %>

  <%= f.label :password %>
  <%= f.password_field :password %>

  <%= f.submit "Log in" %>
<% end %>
```

Sessions Helper

```
module SessionsHelper

  def log_in(user)
    session[:user_id] = user.id
  end

  def log_out
    session.delete(:user_id)
  end

end
```

Sessions Controller

```
def new
 end


 def create
   user = User.find_by(email: params[:session][:email].downcase)
   if user && user.authenticate(params[:session][:password])
     log_in user
     redirect_to user
   else
      flash.now[:danger] = 'Invalid email/password combination‘
     render 'new'
   end
 end
```

```
def destroy
  log_out
  redirect_to root_url
end
```

Sessions Helper

```
def current_user
  @current_user | |= User.find_by(id: session[:user_id])
 end

 def logged_in?
  !current_user.nil?
 end

 def log_out
  session.delete(:user_id)
   @current_user = nil
 end
```

Links inteligentes

```
<% if logged_in? %>
    # link para index de usuários
    # link para ver sua conta
    # link de logout
<% else %>
    # link de login
<% end %>
```

Filtros

Forçar login

```
before_action :logged_in_user
[...]
private
  def logged_in_user
   unless logged_in?
    flash[:danger] = "Please log in."
    redirect_to login_url
   end
  end
end
```

Restringir ao usuário correto

```
before_action :correct_user,   only: [:edit, :update]
[…]

def correct_user
  @user = User.find(params[:id])
  redirect_to(root_url) unless @user == current_user
end
```