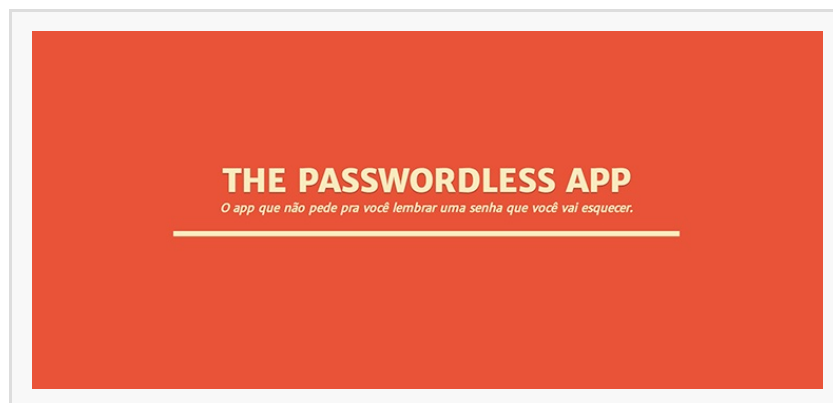


Passwordless Login

Postado por **Matheus Bras** em 11/04/2013



Neste post, vou mostrar como fazer uma aplicação permitir que o usuário faça login sem precisar digitar/lembrar de sua senha. Não será nada complexo demais e nem uma solução perfeita. No final, postarei o link para uma aplicação exemplo que preparei exemplificando o método.

Por que fazer login sem senha?

Fazer login está se tornando uma tarefa complicada. Agora que temos muitos usuários acessando aplicações pelo smartphone, digitar a senha na telinha do telefone é complicado. Mesmo nas telas gigantes dos celulares da Samsung, é chato e ruim. Precisamos de uma maneira mais simples de fazê-lo.

Os usuários têm que lembrar seu username e senha para logarem. Mas muitas vezes não os lembram. E é aí que entra o “esqueci minha senha” e o usuário precisa abrir seu email e clicar no link para poder recuperar sua senha. Ou então para que a senha não seja esquecida eles usam: password, senha123 ou 123456.

Uma solução proposta por Ben Brown neste [post](#) é gerar uma senha totalmente aleatória quando o usuário se cadastrar no site e enviar por email um link para fazer o login. A maioria dos sites permitem que o login seja mantido pra sempre, porém se o usuário precisar fazer o login novamente, a aplicação gera outra senha para o usuário e envia

outro link para ele por email.

Como eu resolvi esse problema

Então vamos ao código:

Primeiro, criei um model User com os campos *email* e *access_token*.

```
1 rails g model User email:string access_token:string
```

Depois, criei um método *generate_access_token* que se encarrega de gerar uma nova senha para o usuário. E também o método *access_token_exists?* (token) para checar se, por acaso, a senha já existe para algum usuário.

```
1 def self.access_token_exists?(token)
2   where(access_token: token).any?
3 end
4
5 private
6 def generate_access_token
7   loop do
8     token = SecureRandom.hex(30)
9     return self.access_token = token unless User.access_token_exists?(token)
10  end
11 end
```

E então implementei o método *generate_access_token_and_save* para gerar a senha para o usuário e salvá-la. Isso fecha por enquanto o model User. Voltaremos nele mais tarde.

```
1 def generate_access_token_and_save
2   generate_access_token and save
3 end
```

Agora vamos ao *UsersController* . Criei um controller simples com duas actions: New e Create.

```
1 # encoding: UTF-8
2 class UsersController < ApplicationController
```

```

3   def new
4     @user = User.new
5   end
6
7   def create
8     @user = User.new(params[:user])
9     if @user.generate_access_token_and_save
10      redirect_to new_user_path, notice: "Agora olha teu email lá! :)"
11    else
12      render :new
13    end
14  end
15 end

```

Em seguida, criei o SessionsController para lidar com o login. O controller vai encontrar o usuário que possua a access_token fornecida e colocar seu id em uma session.

```

1  # encoding: UTF-8
2  class SessionsController < ApplicationController
3    def create
4      user = User.find_by_access_token!(params[:token])
5      session[:user_id] = user.id
6      redirect_to(secret_page_path, notice: "Você está logado! :)")
7    rescue ActiveRecord::RecordNotFound
8      redirect_to(root_url, notice: "Acesso inválido... recupere sua senha.")
9    end
10 end

```

Agora que já tenho o controller para lidar com o link de login, posso criar o Mailer para enviar o link para o email do usuário.

```

1  # encoding: UTF-8
2  class Notification < ActionMailer::Base
3    default from: "estagiario@passwordlessapp.com"
4    layout "mailer"
5
6    def auth_link(user)
7      @user = user
8

```

```
9     mail to: @user.email, subject: "[Passwordless App] Aqui está seu link de ac
10   end
11 end
```

Coloquei, então, a chamada para o envio do email dentro do método `generate_access_token_and_save` no model `User`.

```
1 def generate_access_token_and_save
2   Notification.auth_link(self).deliver if generate_access_token and save
3 end
```

Usei o `SecretPageController` para ter uma action que requer autenticação.

```
1 class SecretController < ApplicationController
2   before_filter :authenticate!
3
4   def index
5   end
6 end
```

Aqui estão os helpers de autenticação criados para o `SecretPageController`.

```
1 # encoding: UTF-8
2 class ApplicationController < ActionController::Base
3   protect_from_forgery
4   ensure_security_headers
5   helper_method :current_user, :user_signed_in?
6
7   private
8
9   def current_user
10    @current_user ||= User.find(session[:user_id]) if session[:user_id]
11    rescue ActiveRecord::RecordNotFound
12      session.delete(:user_id)
13      nil
14    end
15
16    def user_signed_in?
```

```
17     !current_user.nil?  
18   end  
19  
20   def authenticate!  
21     user_signed_in? || redirect_to(root_url, notice: "Você precisa estar autenticado")  
22   end  
23 end
```

E isso já faz o login sem senha funcionar. O usuário se cadastra, recebe um link por email, clica no link, se loga e é redirecionado para a action que requer autenticação.

Eu fiz uma aplicação exemplo que pode ser acessada [clikando aqui](#) e o código também está no [GitHub](#) com todos os testes. Sintam-se à vontade para mandar pull requests, issues e perguntas.

Obrigado pela sua atenção e abraços!

Tags: [matheus bras](#) [ruby](#) [passwordless](#) [rails](#)

Compartilhe

