a month ago
· Devise (/tag/devise/), Omniauth (/tag/omniauth/), facebook (/tag/facebook/), Ruby on Rails 4 (/tag/ruby-on-rails-4/), Ruby (/tag/ruby/), Rails 4 (/tag/rails-4/), Rails (/tag/rails/), Inês Rodrigues (/tag/ines-rodrigues/), Github (/tag/github/), Oauth (/tag/oauth/), Auth (/tag/auth/), Português (/tag/portugues/), PT-PT (/tag/pt-pt/)
· 0 Comments

Rails 4 - Implementar OmniAuth com o Devise (/rails-4-implementar-omniauth-com-o-devise/)

Existem várias formas de implementar OAuth em aplicações de Ruby on Rails, no entanto grande parte delas implica implementar tanta trapalhada que o resultado não podia ser outro para além de dores de cabeça tanto para o programador como para o utilizador final.

Mas existem soluções simples:
Neste tutorial vou explicar passo-a-passo como instalar o **OmniAuth** (http://intridea.github.io/omniauth/) para registo/login de utilizadores pelo Facebook e integra-lo com a arquitectura da gem **Devise** (https://github.com/plataformatec/devise).

Criar uma plataforma decente para registo de utilizadores é uma tarefa complexa (ainda mais se feita de raiz).
Felizmente existe o **Devise** (https://github.com/plataformatec/devise) - uma gem que trata de criar uma solução flexível em que numa questão de minutos conseguimos ter uma interface para registos de utilizadores.
O que o Devise não oferece à partida é login e registo com redes sociais como o facebook, twitter, google+, etc.
Para colmatar essa lacuna existe o **OmniAuth** (http://intridea.github.io/omniauth/).

Mas como juntar os dois de forma eficiente?
O primeiro passo é decidir qual a "strategy" a implementar. Para este tutorial vamos configurar a autenticação do utilizador através do Facebook.

Podem ver aqui a lista de todas as strategys possíveis com o Omniauth (https://github.com/intridea/omniauth/wiki/List-of-Strategies)

ATENÇÃO: Este guia parte do principio que já tem uma aplicação com o Devise a funcionar a 100%

pst...se quiserem depois posso fazer um tutorial que explique como funciona o Devise, just let me know :)

1 - Adicionar as seguintes gems à `Gemfile`

```
gem 'omniauth'
gem 'omniauth-facebook'
```

Não esquecer de fazer `bundle install` para que as novas gems sejam instaladas na aplicação.

2 - Criar uma nova App no Facebook Developers (https://developers.facebook.com)

É chegada a hora de criar uma ponte entre o nosso servidor e o servidor do provedor OAuth a que pretendemos aceder.

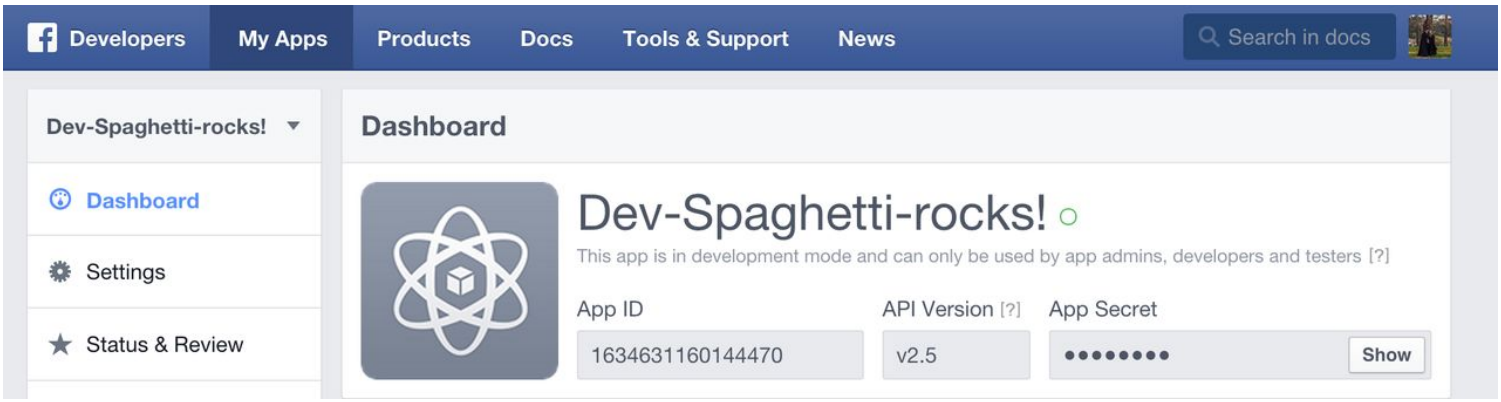
No menu principal do Facebook Developers (https://developers.facebook.com) procurem:
My Apps > **New App**

No menu de "New App" seleccionem "Website" (ou o que melhor se identificar com a vossa aplicação).
Coloquem o nome e carreguem no botão "Create App ID"

No **SiteUrl** é importante que coloquem o endereço onde tem a vossa aplicação. No meu caso, está no localhost.

Para mais detalhes sobre a configuração de facebook Apps podem consultar a documentação aqui (https://developers.facebook.com/docs)

Depois das configurações estarem concluídas irá servos atribuída uma `API KEY` (App ID) e um `API SECRET` (App Secret).



3 - Configurações

No ficheiro `config/initializers/devise.rb` vamos começar a integrar o Omniauth com o Devise.

Basta adicionar a seguinte linha de código:
`config.omniauth :facebook, "API KEY", "API SECRET"`
Devem substituir onde diz API KEY e API SECRET pelas vossas respectivas chaves.

Não esquecer que este código tem de ser colocado dentro do ciclo do Devise.setup.
No final deverá ter este aspecto:

```
Devise.setup do |config|
...
   config.omniauth :facebook, "API KEY", "API SECRET"
...
end
```

4 - Base de dados

É do nosso interesse manter registo dos provedores de OAuth pelos quais o utilizador se está a conectar. Também vamos guardar qual é o UID do Utilizador em cada provedor.

Desta forma, sem grandes complicações, conseguimos permitir que o utilizador faça login através de várias plataformas diferentes sem criar qualquer tipo de conflito.

Basta executar os seguintes comandos no terminal para criar as colunas "provider" e "uid" como sendo strings

```
rails g migration AddOmniauthToUsers provider:index uid:index
rake db:migrate
```

---

5 - Criar novos métodos no user.rb

Está na altura de tornar o nosso User Model "omniauthable"!
Vamos a: `app/models/user.rb`

É necessário acrescentar `:Omniauthable, :omniauth_providers => [:facebook]` aos Devise Modules que estiverem definidos neste Model.
No final deverá ficar similar ao seguinte código:

```
ackable, :validatable,:omniauthable, :omniauth_providers => [:facebook]
```

Para que as mudanças surtam efeito é necessário re-iniciar o servidor.

---

6 - Alterar a View de Registo

Após este último passo o Devise deverá ter criado com sucesso routes para usarmos o Omniauth.
Para ver as routes existentes corram o seguinte comando:

```
rake routes
```

Deverão aparecer os seguintes prefixos novos:

```
user_omniauth_authorize
user_omniauth_callback
```

Para acrescentar um link para autenticação do user pelo facebook basta adicionar à view:

```
<%= link_to "Entrar pelo Facebook", user_omniauth_authorize_path(:facebook) %>
```

Ao carregar nesse link o utilizador deverá ser redireccionado com sucesso para o facebook onde deverá aceitar que a nossa aplicação aceda às informações da sua conta.
A pagina de facebook irá depois reencaminhar o utilizador de volta para a nossa página. No entanto ao voltar vamos encontrar uma página de erro. O problema é que ainda não definimos o método do callback.

7 - Callback

Dentro da pasta `app/controllers/users/` vamos gerar um novo controlador para manipular os callbacks que recebermos do Omniauth.

Execute o seguinte comando no terminal:

```
rails generate controller omniauthCallbacks
```

O comando acima deverá ter gerado um ficheiro com o nome

```
omniauth_callbacks_controller.rb
```

Substitua o código que lá estiver pelo seguinte:

```
class Users::OmniauthCallbacksController < Devise::OmniauthCallbacksCon
end
```

Desta forma o nosso controlador faz extend a `Devise::OmniauthCallbacksController`, permitindo-nos ter acesso a métodos do Devise, como por exemplo os de validação.

Como implementar o callback?

O Callback deverá ser implementado como um método com o mesmo nome do provedor.
No nosso caso o **provedor** (provider) é o **facebook**

```
class Users::OmniauthCallbacksController < Devise::OmniauthCallbacksCon

 def facebook # ponto 1
    @utilizador = User.assuntos_do_omniauth(request.env["omniauth.auth"
    if @utilizador.persisted? # ponto 3
      sign_in_and_redirect @utilizador
    else
      session["devise.facebook_data"] = request.env["omniauth.auth"] #
      redirect_to new_user_registration_url
    end
  end

end
```

Existem vários pontos para os quais vale a pena chamar à atenção neste código. Os pontos estão marcados em comentário no código e irei explica-los aqui:

1. Não esquecer que o nome do método tem que ser o nome do provider - aqui é o "facebook"
2. `request.env["omniauth.auth"]` é a hash que contém toda a informação que o facebook nos devolve. A informação varia consoante o "scope" usado.
   Para verem a Hash completa, bem como os diferentes scopes, podem consultar o README do omniauth-facebook (https://github.com/mkdynamic/omniauth-facebook#auth-hash)
3. O código `if utilizador.persisted?` devolve True se o registo da entidade na base de dados não for considerado um novo registo e não tiver sido destruído. Caso tenha ocorrido algum problema com o registo, devolve False.
4. No caso de ocorrer algum problema guardamos toda a informação devolvida pelo facebook na sessão: `session["devise.facebook_data"] = request.env["omniauth.auth"]`

É de salientar que usamos como key namespace o "Devise" para guardar esta informação.
Porque? O Devise irá eliminar toda a informação da Session após o login do Utilizador, fazendo assim um clean up à session por nós.

Para não complicar este guia estes dados guardados na session (em caso de erro) não serão usados, mas fica a dica de como os podem guardar e usar para, por exemplo, display de erros :) Sejam criativos!

---

8 - Para finalizar, terminar o Model (está quase, go go go)

Como passo final vamos adicionar ao nosso Model em `app/models/user.rb` o método `assuntos_ do_omniauth`

```
def self.assuntos_do_omniauth(auth)
  where(provider: auth.provider, uid: auth.uid).first_or_create do |use
    user.email = auth.info.email
    user.password = Devise.friendly_token[6,15]
  end
end
```

Caso não encontre nenhum utilizador com o provider e o uid fornecidos irá criar um novo user com o e-mail devolvido na Hash. Irá também gerar uma password aleatória para este novo utilizador.
Neste método podem adicionar todos os campos que quiserem desde que venham na resposta do auth e a BD esteja pronta para os receber.

Temos agora uma forma de autenticação com o facebook 100% válida e integrada com o Devise na nossa Aplicação de Ruby on Rails 4.

Bom trabalho

Para os mais curiosos:

DEMO | **GITHUB**
(irei tentar colocar os links para um demo no Heroku e para o código no Github o mais brevemente possível ^_^)

WRITTEN BY
Inês Rodrigues ()
Web, Web never changes...
Coimbra

SPREAD THE WORD

0

Published on
December 07, 2015