



OAuth / OmniAuth Intro

Devpoint Labs - Jake Sorce

History

- OAuth started around November 2006
- Conclusion that there was no open standard for API access delegation
- April 2007, a Google group was created with a small group of implementers to write a proposal for an open protocol.
- July 2007 the team drafted an initial specification and the group was opened to anyone interested in contributing
- October 3rd, 2007 the OAuth Core 1.0 final draft was released

What is it for?

- Valet Car Key vs Regular Car Key Example
- It allows you, the User, to grant access to your private resources on one site (which is called the Service Provider), to another site (called Consumer, not to be confused with you, the User).

Is OAuth a New Concept?

- No
- similar to other protocols currently in use (Google AuthSub, aol OpenAuth, Yahoo BBAuth, Upcoming api, Flickr api, Amazon Web Services api)
- Each protocol provides a proprietary method for exchanging user credentials for an access token.
- OAuth was created by carefully studying each of these protocols and extracting the best practices and commonality that will allow new implementations as well as a smooth transition for existing services to support OAuth.

OmniAuth Rails Follow Along

What is OmniAuth?

- OmniAuth is a Ruby authentication framework aimed to abstract away the difficulties of working with various types of authentication providers. It is meant to be hooked up to just about any system, from social networks to enterprise systems to simple username and password authentication.

Create a new Rails Project

```
rails new omniauth_facebook -T -d postgresql
```

-T => Do not generate the default testing framework (mini test)

-d postgresql => generate rails project with postgresql database (best for heroku integrations)

Add Gems

- `gem 'devise'`
 - devise for user auth
- `gem 'omniauth-facebook'`
 - omniauth facebook integration gem
- `bundle install`

Devise Setup

- rails generate devise:install
 - sets up devise in your rails project
 - make sure you follow all of the devise output
- rails generate devise user
 - generates a devise user model

Controller Setup

- rails g controller home index
 - generate a controller called home with an index action
- open:
app/controllers/application_controller.rb
 - add code: before_action :
authenticate_user!

Omniauth Facebook Setup

- rails g migration AddOmniauthToUsers provider:index uid:index
- bundle exec rake db:drop db:create db:migrate
- register for a facebook developers account
 - <https://developers.facebook.com/docs/apps/register>

Omniauth Facebook Setup Cont...

- create a new 'www' app
- name the app whatever you want
 - (eg.FirstOmniAuthApp)
- Choose a category and create
- save your appld somewhere

Omniauth Facebook Setup Cont...

- Skip the quick start if you see it
- add a new platform via settings on the left hand side
- click 'website'
- add <http://localhost:3000> as your site url

Configure Devise For Facebook OmniAuth

- open: `config/initializers/devise.rb`
 - add this code:
 - `config.omniauth :facebook, "app_id", "app_secret", scope: 'email', info_fields: 'email, name'`
 - `app_id` and `app_secret` can be found in the settings page of your app on the facebook developer account

The screenshot shows the Facebook Developer Settings interface. On the left is a sidebar with navigation links: 'testing' (selected), 'Dashboard', 'Settings' (highlighted with a blue gear icon), 'Status & Review', 'App Details', and 'Roles'. The main content area has three tabs: 'Basic' (active), 'Advanced', and 'Migrations'. Under the 'Basic' tab, there are two columns of settings. The left column contains 'App ID' (771949679576003), 'Display Name' (testing), and 'App Domains' (empty). The right column contains 'App Secret' (a0972a72b5c614de1588cd3bd349aa8d, highlighted in blue), 'Namespace' (empty), and 'Contact Email' (empty). A 'Reset' button is next to the App Secret field. A descriptive text 'Used for important communication about your app' is below the Contact Email field.

testing ▼	Basic	Advanced	Migrations
Dashboard	App ID 771949679576003	App Secret a0972a72b5c614de1588cd3bd349aa8d Reset	
Settings	Display Name testing	Namespace	
Status & Review	App Domains	Contact Email	
App Details		Used for important communication about your app	
Roles			

*Security Notice!!!

```
config.omniauth :facebook, "app_id", "app_secret", scope: 'email',  
  info_fields: 'email, name'
```

app_id and app_secret should NOT be committed to GitHub.

This means that we need to put them in a config file that ignores them and setup an initializer for development purposes

Configure Devise For Facebook OmniAuth

- open: app/models/user.rb
- add this code to the devise list:
 - :omniauthable, :omniauth_providers => [:facebook]

Configure Routes

- Replace `devise_for :users` with
 - `devise_for :users, :controllers => { :omniauth_callbacks => "users/omniauth_callbacks" }`
- add file:
 - `app/controllers/users/omniauth_callbacks_controller.rb`

Configure Controller / Models

- open file: app/controllers/users/omniauth_callbacks_controller.rb
- add code:
 - ```
class Users::OmniauthCallbacksController < Devise::OmniauthCallbacksController
 def facebook
 # You need to implement the method below in your model (e.g. app/models/user.rb)
 @user = User.from_omniauth(request.env["omniauth.auth"])

 if @user.persisted?
 sign_in_and_redirect @user, :event => :authentication #this will throw if @user is
not activated
 set_flash_message(:notice, :success, :kind => "Facebook") if
is_navigational_format?
 else
 session["devise.facebook_data"] = request.env["omniauth.auth"]
 redirect_to new_user_registration_url
 end
 end
end
```



# Controller Description

1. This action has a few aspects worth describing:
2. All information retrieved from Facebook by OmniAuth is available as a hash at `request.env["omniauth.auth"]`. Check the OmniAuth docs and each omniauth-facebook gem's README to know which information is being returned.
3. When a valid user is found, they can be signed in with one of two Devise methods: `sign_in` or `sign_in_and_redirect`. Passing `:event => :authentication` is optional. You should only do so if you wish to use Warden callbacks.
4. A flash message can also be set using one of Devise's default messages, but that is up to you.
5. In case the user is not persisted, we store the OmniAuth data in the session. Notice we store this data using "devise." as key namespace. This is useful because Devise removes all the data starting with "devise." from the session whenever a user signs in, so we get automatic session clean up. At the end, we redirect the user back to our registration form.

# Implement from\_omniauth

- open file: app/models/user.rb
- add code:
  - `def self.from_omniauth(auth)`
  - `where(provider: auth.provider, uid: auth.`  
 `uid).first_or_create do |user|`
  - `user.email = auth.info.email`
  - `user.password = Devise.friendly_token`  
 `[0,20]`
  - `end`
  - `end`



# Model Implementation Description

This method tries to find an existing user by the provider and uid fields. If no user is found, a new one is created with a random password and some extra information. Note that the `first_or_create` method automatically sets the provider and uid fields when creating a new user. The `first_or_create!` method operates similarly, except that it will raise an Exception if the user record fails validation.

# Add Logout Link

- open: `app/views/home/index.html.erb`
- remove all default code
- add code:
  - `<h1>Email: <%= current_user.email %></h1>`
  - `<%= link_to 'Logout', destroy_user_session_path, method: 'delete' %>`



# OmniAuth and Facebook

- <https://github.com/plataformatec/devise/wiki/OmniAuth:-Overview>