
Devise- Uni-Sale

Michael A Stokes, Michael Stokes

20th March 2015

1 Introduction

Implementing Devise within the website for User priviledges such as login, register etc.

1.1 Installation

1. gem devise
2. rails generate devise:install
3. rails generate devise:views
4. rails generate devise User
5. rake db:migrate
6. rails server

1.2 Devise Configurations

1.2.1 Modules

Extra Modules will be added:

1. Confirmable : sends emails with confirmation instructions and verifies whether an account is already confirmed during sign in <http://www.rubydoc.info/github/plataformatec/devise/master/Devise/Models/Confirmable>
2. Timeoutable : expires sessions that have not been active in a specified period of time. <http://www.rubydoc.info/github/plataformatec/devise/master/Devise/Models/Timeoutable>
3. Lockable : locks an account after a specified number of failed sign-in attempts. Can unlock via email or after a specified time period. <http://www.rubydoc.info/github/plataformatec/devise/master/Devise/Models/Lockable>

1.2.2 User Attributes

Extra Attributes will be added to the User and Devise Configuration:

1. Username : Users must register a username and will be able to login by that username.
The username is the Users identity.
 - (a) rails generate migration add_username_to_users username:string:uniq
 - (b) before_action :configure_permitted_parameters, if: :devise_controller?
 - (c) attr_accessor :login
<https://github.com/plataformatec/devise/wiki/How-To:-Allow-users-to-sign-in-using>
2. First Name : Users must register a First Name.
 - (a) rails generate migration add_first_name_to_users first_name:string
3. Last Name : Users must register a Last Name.
 - (a) rails generate migration add_last_name_to_users last_name:string
4. Phone Number : Users must register a Phone Number.
 - (a) rails generate migration add_phone_number_to_users phone_number:string
5. rake db:migrate

1.2.3 Override Default Devise

We will override the default Devise Configuration and thus generate a Devise Controller and then override the methods i.e. in registration and session.

1. Controller : Override default Devise and generate Devise Controllers:
 - (a) rails generate devise:controllers users
 - (b) Add Filters i.e. before calling create call private method prior_create to check domain valid and other checks etc and if not redirect with flash message.
<https://github.com/plataformatec/devise>
2. Routes : Alter the Routes to use the defined Devise Controllers:
 - (a) Registrations and Sessions Controller for Sign-up and Login

<https://github.com/plataformatec/devise>

2 Registration

Implementing Devise Custom Registration. The default Devise Configuration will be changed and thus Devise Controller Actions will be override and filters will be used.

2.1 Registration Steps

1. User Enters Registration details
2. Controller called, before filter in place i.e. before create in register controller is called prior_create will be called rather which will perform many checks.
3. Registration prior create checks include:
 - (a) Validation.
 - (b) Check Domain, is it a valid ac.uk address i.e. surrey.ac.uk and does it exist in the Domains table.
 - (c) Uniqueness.
4. Registration Post Steps include:
 - (a) Send Activation Email
 - (b) User Activates and then can login.

References

- [1] Devise - Ruby Gem : *User Login/Register*. <https://github.com/plataformatec/devise>,