

# Rails : Implement one-one using has\_one and using nested forms

---

Nested Form: means we can save multiple models data from a single form

## 1. Add required associations between models

```
class User < ApplicationRecord
  has_one :profile, dependent: :destroy
end

class Profile < ApplicationRecord
  belongs_to :user
end
```

## 2. Add nested form

```
class User < ApplicationRecord
  # Include default devise modules. Others available are:
  # :confirmable, :lockable, :timeoutable, :trackable and :omniauthable
  devise :database_authenticatable, :registerable,
         :recoverable, :rememberable, :validatable, :trackable

  ## Associations ##
  has_one :profile, dependent: :destroy

  ## Nested Forms ##
  accepts_nested_attributes_for :profile, allow_destroy: true
end
```

## 3. Controller changes to support nested form and we are overriding devise registrations controller

### a. Create a **users** folder in controllers i.e **users/registrations\_controller.rb**

```
class Users::RegistrationsController < Devise::RegistrationsController
  before_action :sign_up_params, only: [:create]
  before_action :account_update_params, only: [:update]

  def new
    puts "\n\n...Im in ur local...\n\n"
  end
end
```

```

    build_resource({})
    resource.build_profile
    respond_with self.resource
  end

  def create
    super
  end

  def update
    super
  end

  private

  def sign_up_params
    devise_parameter_sanitizer.sanitize(:sign_up)
    params.require(:user).permit(:email, :password, :password_confirmation,
    profile_attributes: [:id, :first_name, :last_name, :phone, :location, :zip])
  end

  def account_update_params
    params.require(:user).permit(:current_password, :email, :password,
    :password_confirmation, profile_attributes: [:id, :first_name, :last_name, :phone,
    :location, :zip])
  end
end

```

b. Update routes to tell the devise to override the controller in **routes.rb** file

```
devise_for :users, controllers: {registrations: 'users/registrations'}
```

#### 4. View page changes to support Nested Form

a. `app/views/users/registrations/new.html.erb`

```

<h2>Sign up</h2>

<%= form_for(resource, as: resource_name, url: registration_path(resource_name)) do |f| %>
  <%= render "users/shared/error_messages", resource: resource %>

  <div class="field">
    <%= f.label :email %><br />
    <%= f.email_field :email, autofocus: true, autocomplete: "email" %>
  </div>

  <div class="field">
    <%= f.label :password %>
    <% if @minimum_password_length %>
    <em><%= @minimum_password_length %> characters minimum</em>

```

```

    <% end %><br />
    <%= f.password_field :password, autocomplete: "new-password" %>
  </div>

  <div class="field">
    <%= f.label :password_confirmation %><br />
    <%= f.password_field :password_confirmation, autocomplete: "new-password" %>
  </div>

  <%= render 'profile_fields', f: f %>

  <div class="actions">
    <%= f.submit "Sign up" %>
  </div>
<% end %>

```

#### b. Partial file app/views/users/\_profile\_fields.html.erb

```

<%= f.fields_for :profile do |p| %>
  <div class="field">
    <%= p.label :first_name %><br />
    <%= p.text_field :first_name %>
  </div>
  <div class="field">
    <%= p.label :last_name %><br />
    <%= p.text_field :last_name %>
  </div>
  <div class="field">
    <%= p.label :phone %><br />
    <%= p.text_field :phone %>
  </div>
  <div class="field">
    <%= p.label :location %><br />
    <%= p.text_field :location %>
  </div>
  <div class="field">
    <%= p.label :zip %><br />
    <%= p.text_field :zip %>
  </div>
<% end %>

```

Ref:

<https://github.com/nathanvda/cocoon>