

```
#####
#                                     #
#  sinatra_application.rb  #
#                                     #
#####

require 'sinatra'

set :root, File.dirname(__FILE__)

enable :sessions

def require_logged_in
  redirect('/sessions/new') unless is_authenticated?
end

def is_authenticated?
  return !!session[:user_id]
end

get '/' do
  erb :login
end

get '/sessions/new' do
  erb :login
end

post '/sessions' do
  session[:user_id] = params["user_id"]
  redirect('/secrets')
end

get '/secrets' do
  require_logged_in
  erb :secrets
end
```

```
#####
#                                     #
#  sinatra_modular_application.rb  #
#                                     #
```

```
#####

require 'sinatra/base'

class SimpleApp < Sinatra::Base

  set :root, File.dirname(__FILE__)

  enable :sessions

  def require_logged_in
    redirect('/sessions/new') unless is_authenticated?
  end

  def is_authenticated?
    return !!session[:user_id]
  end

  get '/' do
    erb :login
  end

  get '/sessions/new' do
    erb :login
  end

  post '/sessions' do
    session[:user_id] = params["user_id"]
  end

  get '/secrets' do
    require_logged_in
    erb :secrets
  end

end
```

```
#####
# config.ru #
#####

require File.dirname(__FILE__) + '/app'
```

```
run SimpleApp
```

```
#####
# app.rb #
#####

require 'sinatra/base'

class SimpleApp < Sinatra::Base

  set :root, File.dirname(__FILE__)

  enable :sessions

  def require_logged_in
    redirect('/sessions/new') unless is_authenticated?
  end

  def is_authenticated?
    return !!session[:user_id]
  end

  show_login = lambda do
    erb :login
  end

  receive_login = lambda do
    session[:user_id] = params["user_id"]
    redirect '/secrets'
  end

  show_secrets = lambda do
    require_logged_in
    erb :secrets
  end

  get '/', &show_login
  get '/sessions/new', &show_login
  post '/sessions', &receive_login
  get '/secrets', &show_secrets

end
```

```
#####
# app.rb #
#####

require 'sinatra/base'

require_relative 'helpers'
require_relative 'routes/secrets'
require_relative 'routes/sessions'

class SimpleApp < Sinatra::Base

  set :root, File.dirname(__FILE__)

  enable :sessions

  helpers Sinatra::SampleApp::Helpers

  register Sinatra::SampleApp::Routing::Sessions
  register Sinatra::SampleApp::Routing::Secrets

end
```

```
#####
# helpers.rb #
#####

module Sinatra
  module SampleApp
    module Helpers

      def require_logged_in
        redirect('/sessions/new') unless is_authenticated?
      end

      def is_authenticated?
        return !!session[:user_id]
      end

    end
  end
end
```

```
#####
# routes/secrets.rb #
#####

module Sinatra
  module SampleApp
    module Routing
      module Secrets

        def self.registered(app)
          show_secrets = lambda do
            require_logged_in
            erb :secrets
          end

          app.get '/secrets', &show_secrets
        end

      end
    end
  end
end
```

```
#####
# routes/sessions.rb #
#####

module Sinatra
  module SampleApp
    module Routing
      module Sessions

        def self.registered(app)
          show_login = lambda do
            erb :login
          end

          receive_login = lambda do
            session[:user_id] = params["user_id"]
            redirect '/secrets'
          end

          app.get '/', &show_login
```

```
    app.get  '/sessions/new', &show_login
    app.post '/sessions',      &receive_login
  end

end

end

end
```