

Adding Devise and a Before Filter

This example shows the power of Ruby gems for Rails by introducing Devise, the most popular extension for Rails authentication.

Add Authentication to our Class Management app:

```
$ cd class_mngmt/  
$ git status # Should show working directory clean, if not add and  
commit
```

Add the Devise gem

Open up 'GemFile' in an editor, and add the line

```
gem 'devise'
```

Save the File.

```
$ bundle # The output should include Devise and its dependencies  
$ rails generate # You should see the Devise generators  
$ rails generate devise:install
```

```
$ git status
```

You can look at the output of 'git status' to see exactly what Devise did in the install process. This is a good idea when installing any extension, so you know exactly what was added or changed in your project

```
$ git add .  
$ git commit -m "installed devise"
```

Follow the instructions that are printed out by Devise. One of them will be to open up and look at the initializer for Devise. Feel free to change the email address that Devise will use, but it's not necessary since we won't really be sending emails as part of this exercise (but, of course, in real life use you will want to use a different email).

One of the instructions will be to edit the 'root :to ...' line in the routes.rb File. Have this go to "assignments#index".

Make sure the public/index.html file is removed:

```
$ rm public/index.html
```

We're removing this so the route change above will be visible Commit to git.

Add a user model for Devise and add a before filter

In normal usage, you might already have a User model. In this case, we're just going to let Devise generate one. As a result, there will only be Fields for things like email and password, and not for things that you might expect such as Full Name.

```
$ rails generate devise User  
$ rake db:migrate
```

Edit the File assignments_controller.rb. Under 'class AssignmentsController ...', add the line:

```
before_filter :authenticate_user!, :except => [:index, :show]
```

Save the File

Apply what you have learned before, to edit routes.rb so that the front page of the site goes to the assignments list page.

```
$ rails s
```

Go to <http://localhost:3000>

When you try to make a change (e.g. editing or adding a new record), you'll notice that you're asked to sign in or sign up.