chase.smith

Authorization Gems

CanCan

Key Points

- Seems fairly simple
- `load_and_authorize_resource` function can be used to verify authentication of user for every resource
- Authorization can be enforced before every action on a resource (this is different from the above point)
- Role based authorization
- Seems lightweight

CanCan is based on posing the (boolean) question "can the user do this?". Example: `<% if can? :update, @article %>`
`  <%= link_to "Edit", edit_article_path(@article) %>`

`<% end %>`

https://github.com/ryanb/cancan

Devise

Key Points

- Looks relatively complex.
- Composed of 10 modules. However, not all of these modules are required, Devise seems to be flexible in that regard.
- The modules include
  - "Database Authenticatable" (not even sure if that's a real word)
  - "Confirmable" (sends email with confirmation instructions)

  - "Recoverable" (resetting the user password)
  - "Timeoutable" (expires sessions that have not been active in a specified period of time)
  - "Lockable" (locks an account after a specified number of failed sign-in attempts)

Devise looks like it would require a pretty decent amount of work to get set up properly, and, though it would provide a lot more functionality than we would likely need, it could be a good solution.
https://github.com/plataformatec/devise

Rolling our own

Key Points

- We learned how to do this, basically, in Chapter 9
- We have the tutorial as a basis
  - Tutorial has a good security model, as well as good tests
- Could be more 'risky'
- We would have a lightweight security system, as we wouldn't have unneeded bloat like if we were to use a generic security gem

Rolling our own would not be a bad idea for this project, as we do not (seem to) need any of the more complex things like password resets, emails, etc. While it would be more 'risky' to roll our own, the tutorial seems to have done a good job in terms of creating proper security, and testing the security model. One thing we would certainly have to do, if we roll our own instead of going with a pre-existing gem, is create authentication tests (like in the tutorial), to make sure that we don't have any glaring vulnerabilities.