

Code critique for Random Engineers

by \mathcal{D}_p (drachma@mit.edu)

Routing

For the most part, RESTful routing is intuitive and functional. There are a few anomalies, though:

- Going to non-existent pages (e.g. `rushees/3000`) right now just displays a Heroku error message. Trying to do something without authorization (e.g. going to `accounts/verify/[:id]` from a non-admin account) just redirects. Better error messages would be a boon.
- Why not make events a resource, instead of having separate routes for `events/new`, `/edit`, `/delete`, and `/ => index`?
- The line

```
get "rushees/edit" => "rushees#edit"
```

is incorrect and extraneous; it should be `"rushees/edit/[:id]"`, and this case is already handled by `resources :rushees` anyway.

Make Proper Use of Rails Automagic

Avoid passing around IDs where possible

```
# accounts_controller.rb, line 13
@rushees = Rushee.findAllByPrimaryContactID
@brother.id
# rushee.rb, lines 18-20
def self.findAllByPrimaryContactID id
  self.where :primary_contact_id => id
end
```

```
# ALL of the above can be replaced by simply
@rushees = Rushee.where(primary_contact:
@brother)
```

By convention, "find_by_foo" returns only one record!

Association names should be consistent and meaningful

```
# rushee.rb, lines 4 and 16
belongs_to :brother, :foreign_key => "primary_contact_id"
attr_accessor :primary_contact_brother
# should be
belongs_to :primary_contact, :class_name => "brother", :
foreign_key => "primary_contact_id"
```

Factor common code through callbacks

```
# accounts_controller.rb, lines 21-24 and 33-36
if ! current_brother.is_admin
  redirect_to accounts_path
  return
end

# These should be factored out into a separate method
# (and the "return" statement can be removed)
before_action :ensure_admin, only: [:verify, :deny]
```

Coding Style

- Comments in your code are very sparse, and some of them are extraneous, for instance “*# For APIs, you may want to use :null_session instead.*” in `application_controller.rb`. Add more comments to make your code more readable.
- Tabs and spaces are mixed; indentation levels are inconsistent. `rushees_controller.rb` is one instance.
- `!` is sometimes followed with a space and sometimes not (example: in `accounts_controller.rb`, `if !current_brother.is_admin` and `if !current_brother.is_admin`)
- You have residual `console.logging` in `rushees.js`.
- In `accounts_controller.rb`, in the `processInvite` method, instance variables that are used only once should instead be local variables (`@firstname @lastname` and `@email`)

Better Self-Documenting Code

```
# rushee.rb, line 38-53
def validActionSelectStatuses
  return [["None", "None"], ["Stay the
Course", "Stay the Course"],
["Push Harder", "Push Harder"],
["Repudiate", "Repudiate"]]
end
# ... two similar methods ...
def validBidStatuses
  return ["None", "Offered",
"Accepted", "Rejected"]
end
```

1. Should be constants or at least class methods, not object methods that return literals.
2. Inconsistent: both flat and nested arrays.
3. Would a hash be a better data structure here?

```
# accounts_controller.rb, lines 2-3
before_filter :authenticate_brother!
before_filter :is_verified!
```

1. `authenticate_brother!` isn't defined anywhere.
2. `!` conventionally denotes methods that modify the object. Something called `is_verified` shouldn't modify the object; I'd expect it to be called `is_verified?`. Your method should be called `ensure_verified`.
3. Why are there (nearly) identical `is_verified` methods in your other controllers? Should these be in `application_controller.rb` instead?

```
# rushee.rb, lines 30-36
def assignedBrother
  if self.brother
    brother.firstname + " " + brother.lastname
  else
    "Unassigned"
  end
end
```

The method name suggests that this should return a `Brother`, not a string.