ActiveRecord

General Terms

- `first` on a collection gets, the first value
- `where` allows you to get a collection of objects
- Rails <5 used `ActiveRecord::Base` for inheritance instead of `ApplicationRecord`
  - `ApplicationRecord` is just an abstract class for `ActiveRecord::Base`

Creating

- Can use `{model}.new` to create a new in-memory object
- Can use `{model}.save` to save something or `{model}.save!` to save and throw exceptions
- Can use `{mode}.create({hash})` to create a new instance and save at the same time

Updating

- Can get the instance with `find` or `where`, update in memory, then call `save`
- Can use the `update` method with a dictionary input specifying the fields and values to update
  - Earlier versions of rails (<v6) uses `update_attributes` method instead
- Transactional: all or none

Deleting

- Use `destory` to delete a record
  - Is an instance method, so you have to get the record first
- Once destroyed, you can access but not modify in-memory object
- Destroy can handle other dependencies
- `delete` does exist but is probably not what you want

Routing

Operation	Method	URI format
Create	POST	/resources
Read	GET	/resources/:id
Update	PUT	/resources/:id
Delete	DELETE	/resources/:id
Index	GET	/resources
(New)	GET	/resources/:id/new
(Edit)	GET	/resources/:id/edit

Figure 1: Screenshot_2023-09-20_at_9.09.16_PM.png

- New and edit routes are forms that allow the user to interact with data

  - Would not need if app were being accessed via an API
- `config/routes.rb` is where routes are defined
  - `{http verb} '{route}' => '{controller}#{controller method}'`
  - Default actions are {create, show, update, destroy, index, new, edit} respectively
  - Can also define all of the above actions with `resources :{controller}`
- `{controller (singular)}_path({id})` gets the path to that record's route
- `new_{controller (singular)}_path({id})` gets the path to that record's new route
- `{controller}_path` gets the path to that record's collection
- Can use `$ rails routes` to get all routes
  - Have to use `$ rake routes` for v5 and before

MVC Responsibilities

- Model: methods to get/manipulate data
- Controller: get data from model, make available to view
- View: display data, allow user interaction

Forms

- Form helpers are used to make variables easy to use in `params` hash
- Strong parameters $\triangleq$ the controller decides which form field parameters are allowed to be passed to the model for update/create
  - For security reasons
  - Use `params.require({model}).permit({params to use})`