CS169 Week 4 Section

Administrivia & Agenda

- **HW3 Sinatra** is due this **Saturday** 9/28 @ 11:59pm
- Peer Reviews for HW2 due **Friday** 9/27 @ 11:59pm
- Project List will be released **this week**

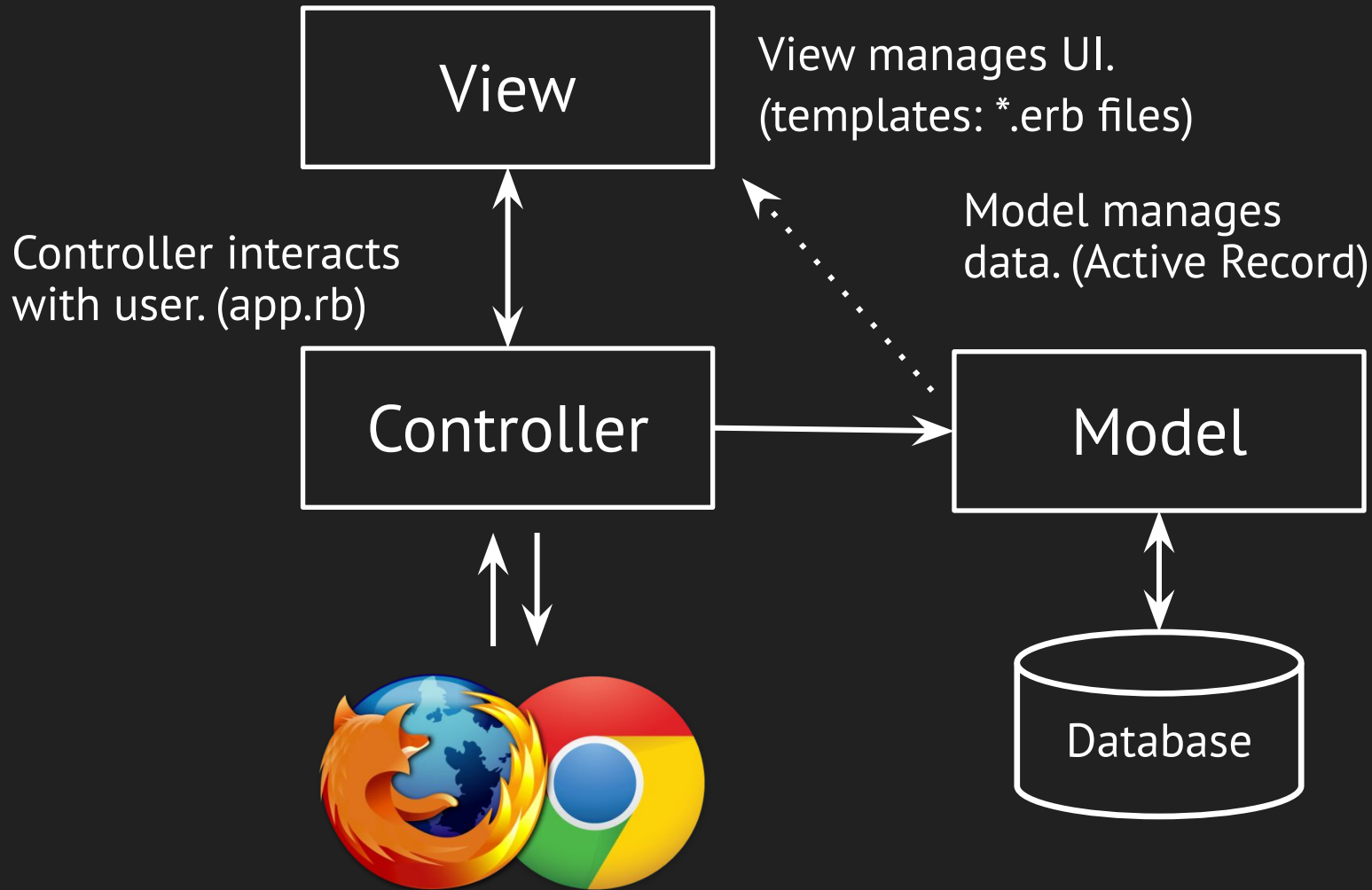Today's Agenda:

- Review of MVC (with Sinatra) and Active Record
- Rails Intro! (with a worksheet)

Attendance

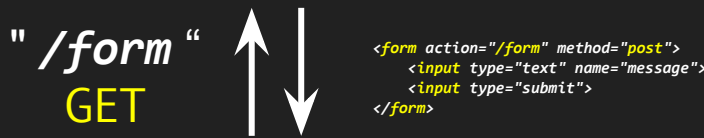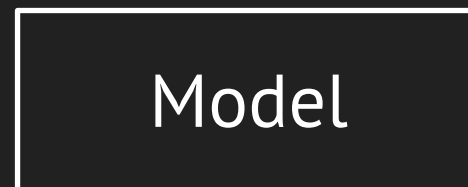https://bit.ly/2mFcZfr

MVC (Review)

View manages UI.
(templates: *.erb files)

Controller interacts
with user. (app.rb)

Model manages
data. (Active Record)

View

Controller

Model

Database

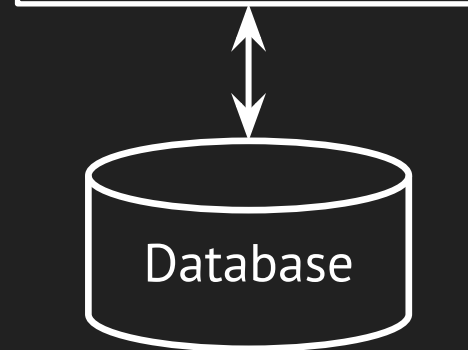# MVC (Review)

```
<form action="/form" method="post">
    <input type="text" name="message">
    <input type="submit">
</form>
```

View
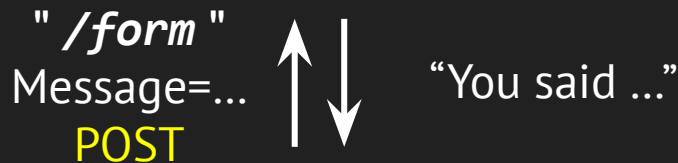
```
get "/form" do
    erb: form
end
```

Controller

"/form"
GET

```
<form action="/form" method="post">
    <input type="text" name="message">
    <input type="submit">
</form>
```

Model

Database

MVC (Review)

View

```
post "/form" do
" You said
#{params[:message]}"
end
```

Controller

Model

" /form "
Message=...
POST

"You said ..."

Database
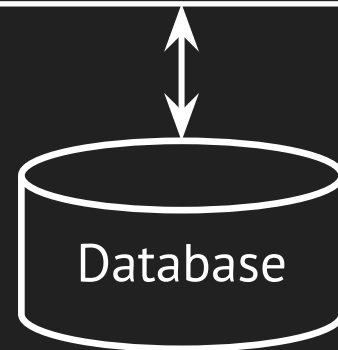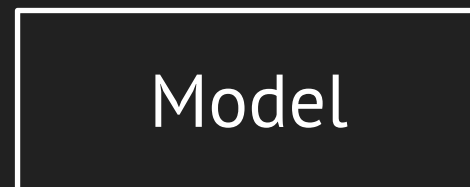
MVC (Review)

View

What about model?

```
post "/form" do
" You said
#{params[:message]}"
end
```

Controller

" /form "
Message=...
POST

"You said ..."

Model

Database

Submit

7
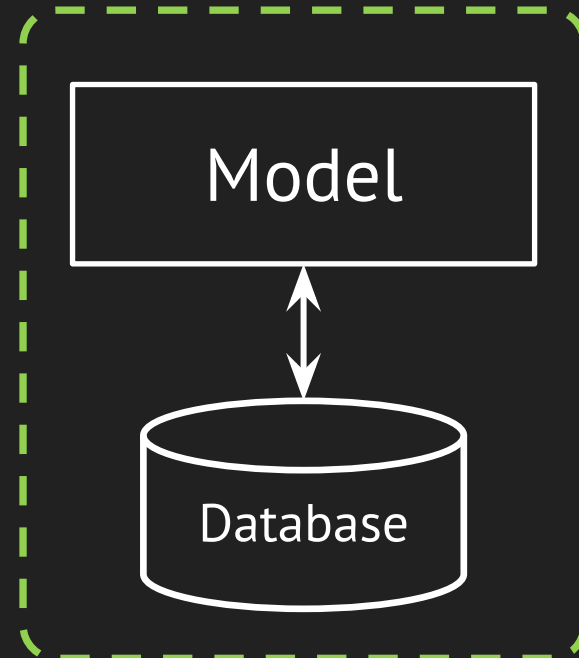
Active Record (Review)

- An Implementation of the object-relational mapping (ORM) pattern.

- Automated mapping between classes and tables, attributes and columns
  - Basic operations on object: CRUD
  - (Create, Read, Update, Delete)

- Associations between objects defined by simple class methods (will be covered later)

8

ActiveRecord Methods (Review)

```
class Article < ActiveRecord::Base {
    :id      => :integer,
    :title   => :string,
    :content => :text
}
```

ActiveRecord will dynamically "generate" methods based on the attributes in the table:

- `Article.find_by_title("Week 5")`
- `Article.find_by_title_and_content(`
  `"First Record", "Hello world")`

Metaprogramming in ActiveRecord

```ruby
def method_missing(meth, *args)
    if meth.to_s =~ /^find_by_(.+)$/
        attr = $1.split('_and_')
        find_by_attribute(attr, *args)
    else
        super
    end
End
```

- `Article.find_by_title("Week 5")`
- `Article.find_by_title_and_content(`
  `"First Record", "Hello world")`

Use ActiveRecord in Sinatra

```ruby
require 'sinatra'
# db configuration
db_options = {adapter: 'sqlite3', database: 'todos_db' }
# establish connection
ActiveRecord::Base.establish_connection(db_options);
# write migration
class CreateTodos < ActiveRecord::Migration
    def change; create_table :todos ... end
end
# enforce migration
createTodos.new.change
# define the active record
class Todo < ActiveRecord::Base
    ...
end
```

Use ActiveRecord in Sinatra

```ruby
require 'sinatra'
# db configuration
db_options = {adapter: 'sqlite3', database: 'todos_db' }
# establish connection
ActiveRecord::Base.establish_connection(db_options);
# write migration
class CreateTodos < ActiveRecord::Migration
    def change; create_table :todos ... end
end
# enforce migration
createTodos.new.change
# define the active record
class Todo < ActiveRecord::Base
    ...
end
```

No need to memorize this.
Rails generates this for you.

Rails Command

- Automatically generate/modify/extend/remove framework & components

- Avoid manually typing common code (DRY)
  - Save you a lot of time and effort

- Enforce consistent code structure
  - All components of the system follows the convention
  - Improve readability, easier to maintain and extend

rails new <app name>

- Generate MVC framework code files.
- run **bundle install**
  - Install ruby packages listed in the **Gemfile**

Workspace
Navigate
Commands

▼ 📁 rails-bootstrap
  ▼ 📁 toy-app
    ▶ 📁 app
    ▶ 📁 bin
    ▶ 📁 config
    ▶ 📁 db
    ▶ 📁 lib
    ▶ 📁 log
    ▶ 📁 public
    ▶ 📁 test
    ▶ 📁 tmp
    ▶ 📁 vendor
      📄 config.ru
      📄 Gemfile
      📄 Gemfile.lock
      📄 Rakefile
      📄 README.rdoc

[M] /README.md ×   Gemfile ×   +

```
1   source 'https://rubygems.org'
2
3
4   # Bundle edge Rails instead: gem 'rails', github: 'rails/rails'
5   gem 'rails', '4.2.5'
6   # Use sqlite3 as the database for Active...
7   gem 'sqlite3'
8   # Use SCSS for stylesheets
9   gem 'sass-rails', '~> 5.0'
10  # Use Uglifier as compressor for JavaScript assets
11  gem 'uglifier', '>= 1.3.0'
12  # Use CoffeeScript for .coffee assets and views
13  gem 'coffee-rails', '~> 4.1.0'
14  # See https://github.com/rails/execjs#readme for more supported runtimes
15  # gem 'therubyracer', platforms: :ruby
16
17  # Use jquery as the JavaScript library
18  gem 'jquery-rails'
19  # Turbolinks makes following links in your web application faster. Read more: https
20  gem 'turbolinks'
21  # Build JSON APIs with ease. Read more: https://github.  13:31   Ruby   Spaces: 2   ⚙
22  gem 'jbuilder', '~> 2.0'
23
```

rails new <app name>

- Generate MVC framework code files.
- run **bundle install**
  - Install ruby packages listed in the **Gemfile**

bash - "jacksongl-rail ×   Immediate ×   +

```
jacksongl:~/workspace $ rails new toy-app
      create
      create   README.rdoc
      create   Rakefile
      create   config.ru
      create   .gitignore
      create   Gemfile
      create   app
      create   app/assets/javascripts/application.js
      create   app/assets/stylesheets/application.css
      create   app/controllers/application_controller.rb
      create   app/helpers/application_helper.rb
      create   app/views/layouts/application.html.erb
      create   app/assets/images/.keep
      create   app/mailers/.keep
```

File Structure

- **app**: core directory of app-specific code
- **bin**: app-specific wrapper scripts
- **config**: configuration files, e.g., db config
- **db**: database files
- **tmp**, **lib**, **log**, **test**: self explanatory
- **public**: static (HTML) files of site
- **Gemfile**: app's gem dependencies
- **Gemfile.lock**: locks dep gem versions

...

Pair Programming!

Pick up a worksheet if you haven't already!