BEWD - Forms & Validations

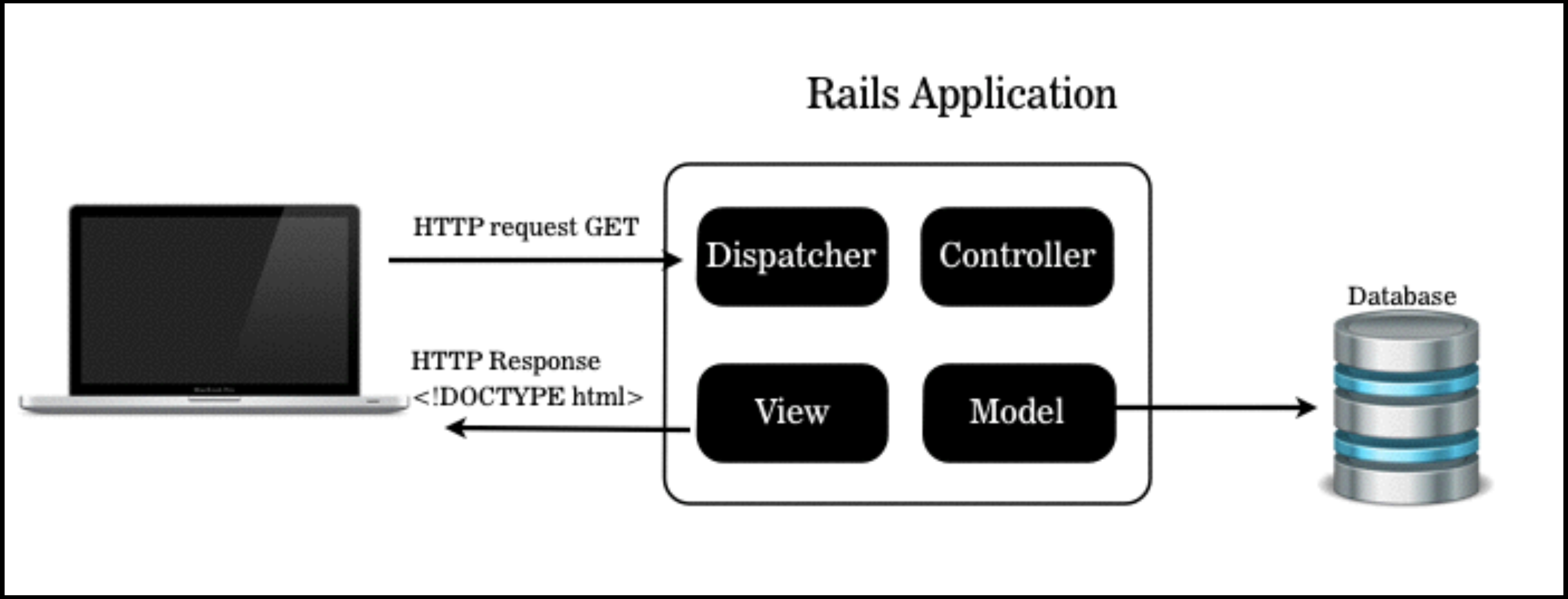
Bill Patrianakos / Week 5 Lesson 2

Agenda

- Review
- Forms
    - Search Forms, Create Forms, Partials
- Validations
- Lab Time

Review

Routes, Model, View, Controller



Shirt Management App

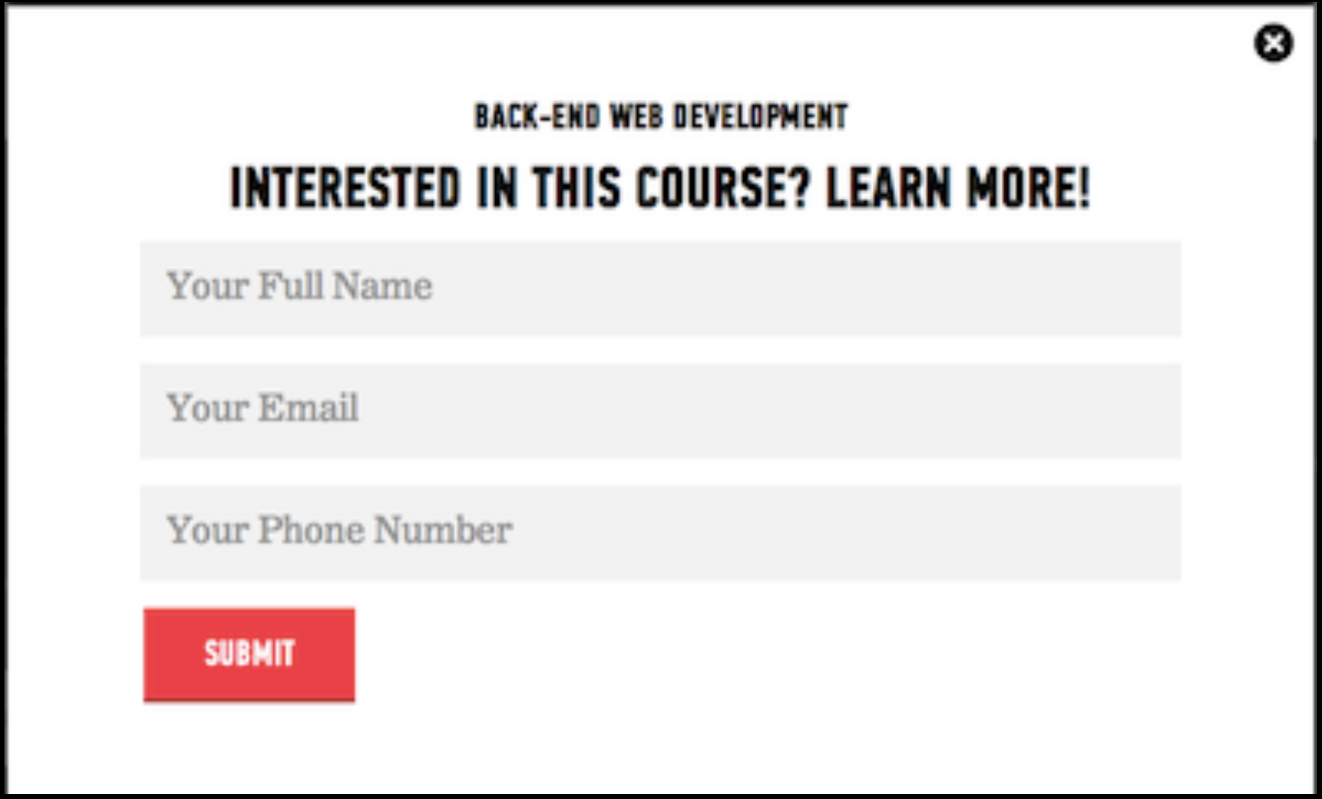
Review

- Add controller and view to Shirts App

Forms

Adding Forms to Views

- Used to 'GET' user input



Shirt Management App

Search Form

- As a user I want to be able to search for a specific T-Shirt in my collection.

</> CODE DEMO

Recap

Search Forms

When a user clicks 'Search'

1. Goes to the Shirts controller index action.
2. Sends the query as a querystring parameter (params[:q])
3. Filters results by the query
4. Searches shirts by name or description fields

Recap

Partials

- File name starts with an underscore
- Inserted into page using the "render" keyword
- Have access to instance variables
- Forms are great candidates for partials
- Rails magically handles the methods for us if we use model objects

Recap

Queries

- Search database with the keywords "where" & "LIKE"

```
where('name LIKE :query OR description LIKE :query', query: “%#{query}%")
```

- Searches the Shirt table's name and description fields.

Recap

Refactor

- Changing the structure of your code without changing behavior
- Move all complex data queries to the model

Movie App

Search

- Create a search box that will search for a term in the movie's title, description and the year
- Create partials for the search box
- Move all complex data queries to the model

Forms

Adding to Database

- We saw how a user can search the database with a form, but how does a user add to the database using forms.

cut it out.

Shirt Management App

Create & Edit Forms

- As a user I want to add shirts to the database.
- As a user I want to edit shirts.

Recap

REST

- REpresentational State Transfer

Recap

REST

- :resources in the routes file gives us LOTS (7) of routes
- Perfect for managing "Resources" (Things in your domain, like Shirts and Movies)

Recap

Form Builders

- Form Builders are awesome
- You MUST create an empty object in the controller for the magic to work

```
<%= form_for @shirt do |f| %>
<% end %>
```

Recap

Strong Parameters

- Rails Controllers require you to whitelist which fields a form can send
- Used to protect from nasty attacks. Github got bitten by one a while back.

Recap

Validation

- Ensures that we are saving accurate information in the database.
    - If our validation fails, don't add record to the database table

* If our validation fails let the user know it failed.

Recap

Private Methods

- Not routable, or callable from the outside world

Recap

Flash

- Used to send messages to the user
- Good for letting them know something succeeded
- Use flash[:notice] and flash[:error]
- Put checks in the layout to always show messages

Homework

Flash

- Movies app due (Lesson 11)

Let's start now.

Movie App

Resources: Forms & Validation

CHEAT SHEET

Reload Code in Console

- Reloads freshest code in the console

```
reload!
```

Validations

Validate Presence

- Validates that title, description and year_released, rating are not empty.

```
validates :name, :description, presence:true
```

Validate Length

- Validates that all descriptions are a minimum of 10 characters.

```
validates :description, length: { minimum: 10 }
```

Numericality

```
validates :year_released, numericality:
      { only_integer: true }
validates :rating, numericality:
      { greater_than_or_equal_to: 1,
      less_than_or_equal_to: 5, only_integer: true }
```

Checks if a record is valid

```
movie.valid?
```

Show current errors in an object

- Shows all of the current errors on an object

```
movie.errors
```

Resources: Forms & Validation

CHEAT SHEET

Strong Parameters

```
def create
  shirt_params = params.require(:shirt).permit(:name, :description, :image)
  shirt = Shirt.create(shirt_params)
  redirect_to shirt
end
```

Private Methods

- Not routable, or callable from the outside world

```
class MyClass
    def call_me
        puts "Called"
    end
    private
    def maybe
        puts "Forever Alone"
    end
end
```

Still Feel Lost?

This course focus was on forms. You should have a general understanding of form builders and what happens when using a search form. Next lesson you will continue to practice Forms, Models, Views, Controller and Routes.

Catch Up With These Resources

- Form Tags about the different tags here and here
- Form Helpers