MVC and Associations

Dylan Cashman
@dylan_cashman

Dynamic Websites

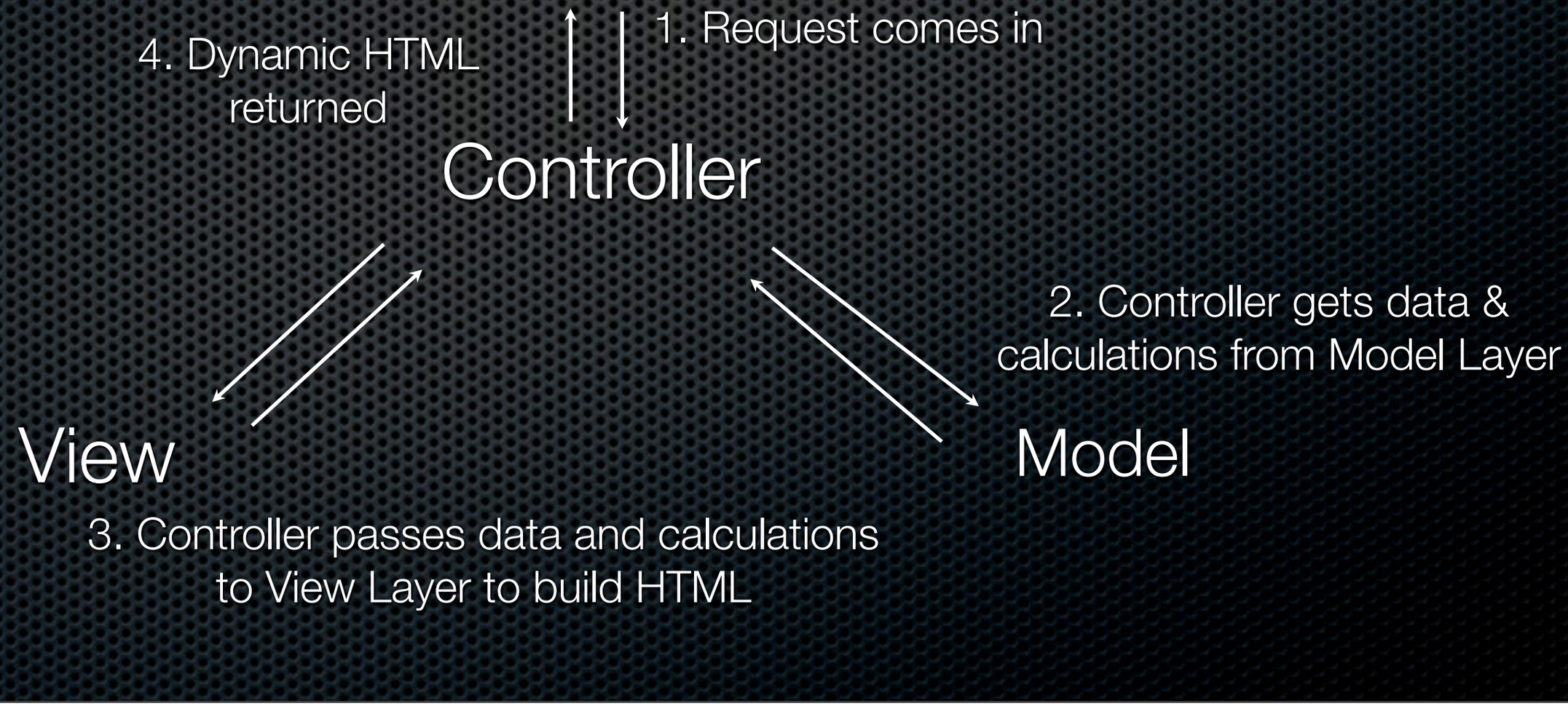
- Receive user input
- Persists information in database
- Respond to user requests by serving up the correct information
- How does our application handle these actions?

Model-View-Controller

- Model-View-Controller (MVC) is the pattern that Rails follows to manage user input
- Models take care of the data
- Views present the data
- Controllers handle interaction with between the user and the rest of the application
- In a Rails application, they are just different files that we add our code to

MVC - Model, View, Controller

- A pattern that describes a user's interaction with a dynamic web application



In Your Rails App

- In your Rails project, you'll see an app/ directory

Courtesy of Dan Pickett's slides of MVC on previous Railsbridge talk

```
|~app/$
|  |+controllers/$
|  |+helpers/$
|  |+models/$
|  `+views/$
```

- This is where our models, views, controllers are kept

Controllers

- Like customer service for your application
- Take user requests, tell the model layer to do things, give back the documents requested to the user

tweets_controller.rb

```ruby
class TweetsController < ApplicationController

  def create
    @tweet = Tweet.new(params[:tweet])
    @tweet.save
    redirect_to('www.twitter.com')
  end

end
```

tweets_controller.rb

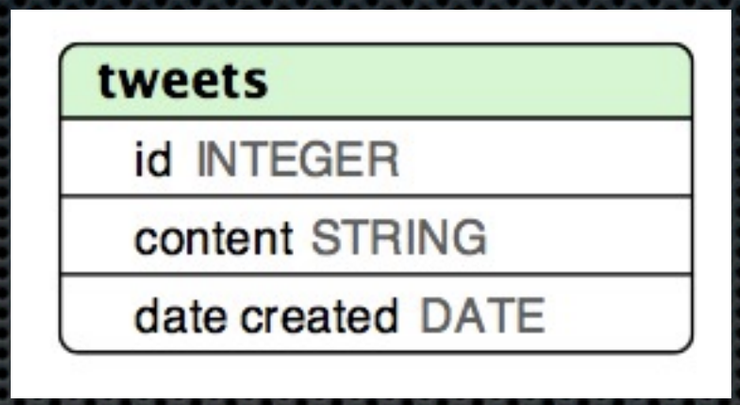
```ruby
class TweetsController < ApplicationController

  def create
    # get the user input
    # post a tweet to the database
    # redirect the user to another page
  end

end
```

Views

- Templates of what a user sees in the browser
- HTML document with spaces to fill in with dynamic data

```
1  <h1>Showing your tweet</h1>
2
3  <p>You just made a tweet.  Here it is!</p>
4  <%= @tweet.content %>
5
6  <p>That's all, folks!</p>
```

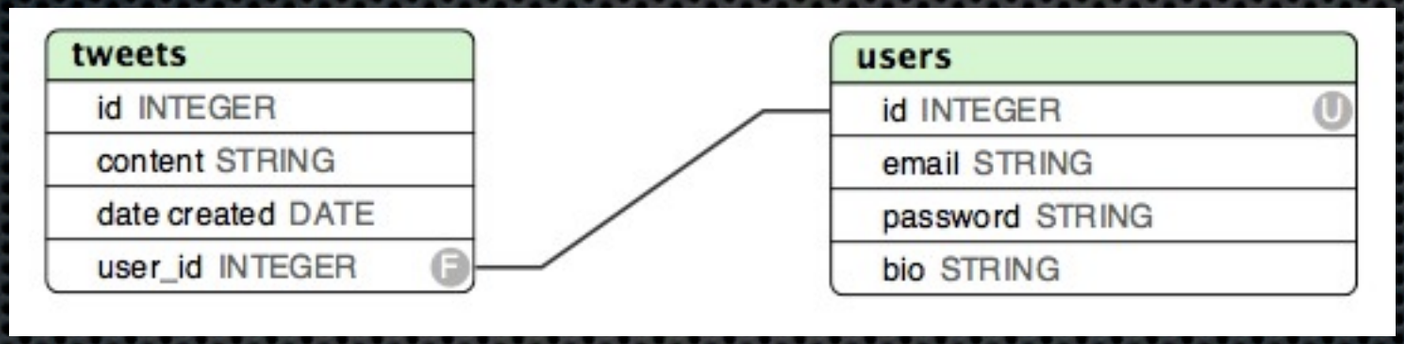
Models



- Represent objects that we keep track of in database
- Twitter models: Tweet, User, FavoriteVote, Message
- CRUD - we can Create a tweet, Read a tweet, Update a user profile, Delete a message
- Each tweet is stored in a database table, and all tweets have the same attributes (text content, date created).

Associations

- How does a single tweet, living in a database, know which user it belongs to?



- Each tweet has to have a reference to the user it belongs to
- In Rails, we tell the model about associations in the database with *belongs_to* and *has_many*

Scaffolding

- Rails generates all these files for you to fill in!
- `rails generate scaffold post`

```
- Creates model (app/models/post.rb)
- Creates controller (app/models/posts_controller.rb)
- Creates views (app/views/posts/)
- Creates migration (db/migrate/...create_posts.rb
```

Conclusion

- MVC is a pattern Rails uses to handle user interaction
- Models represent data that is persisted by our application
- Views are what we see in the browser
- Controllers direct traffic

Thanks!