MVC and REST

MVC

Model

View

Controller

MVC

Model

Interacts with the DB, contains business logic. Has class and instance methods with descriptive names

View

Describes the user interface. Uses values created by Controller

Controller

Interacts with Models to create the values for Views

M, V or C?

Doctors

Patients

Appointments

M, V or C?

Display Dr Barker’s available times

View

M, V or C?

Create an Appointment in the database

Model

M, V or C?

Get doctors with “Heart” specialism

Model

M, V or C?

Receives request, renders page based on request

Controller

M, V or C?

Shows a form to create a new doctor

View

M, V or C?

Receives request, renders page based on request

Controller

MVC

In your own words

What does a Model do? What doesn’t it do?
What does a View do? What doesn’t it do?
What does a Controller do? What doesn’t it do?

Code - where should it go?

```
upcoming_appointments = doctor.appointments.select do |appointment|
    appointment.date > Date.today
End


next_appointment = upcoming_appointments.first
```

Code - where should it go?

```
doctor = Doctor.find(params[:id])

patient = Patient.find_by(name: params[:patient_name])
```

Code - where should it go?

```
<h2>Doctor: <%= @doctor.name %></h2>

<h2>Patient: <%= @patient.name %></h2>

<form>

When would you like your appointment?

<input type=”date” />

<input type=”submit” />

</form>
```

```
end



def rest
```

REST

Representational state transfer

It’s a design pattern (set of rules) for APIs

REST

Client — the client is the **person** or **software** who uses the API.

Resource — a resource can be any object the API can provide information about.

HTTP methods (or verbs)

GET

POST

PATCH

DELETE

URLs

/doctors

/doctors/:id

/doctors/new

/doctors/:id/edit

RESTful routes

GET /doctors

GET /doctors/new

POST /doctors

GET /doctors/:id/edit

PATCH /doctors/:id

DELETE /doctors/:id

Do the same for patients