Putting the Controller in the MVC

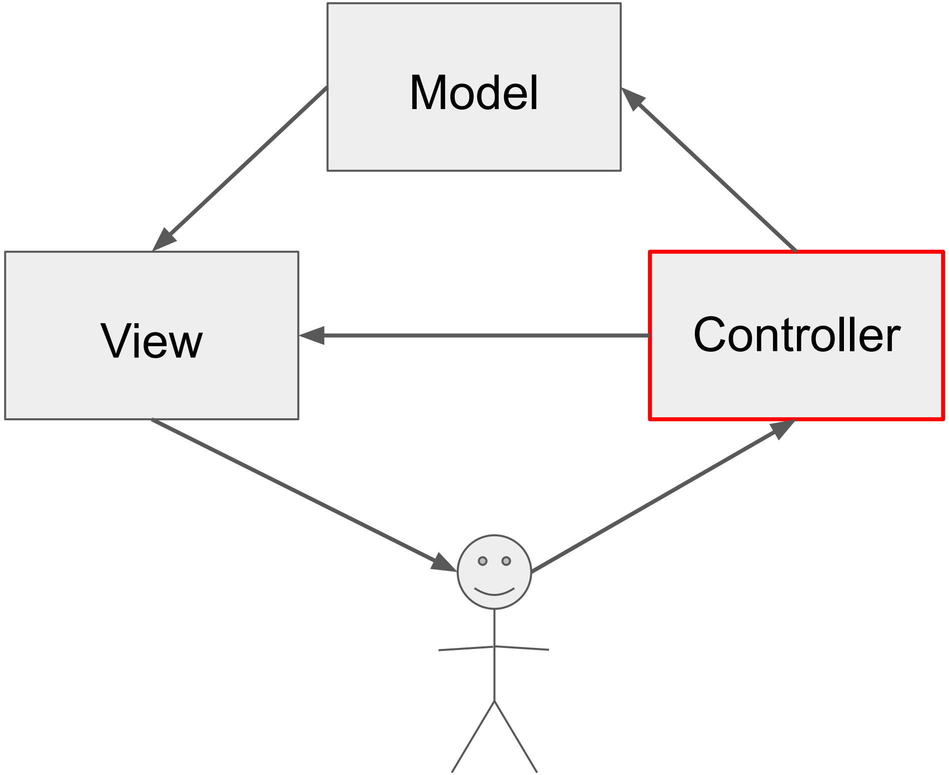
Hugh Cayless
@hcayless

MVC is a “design pattern”

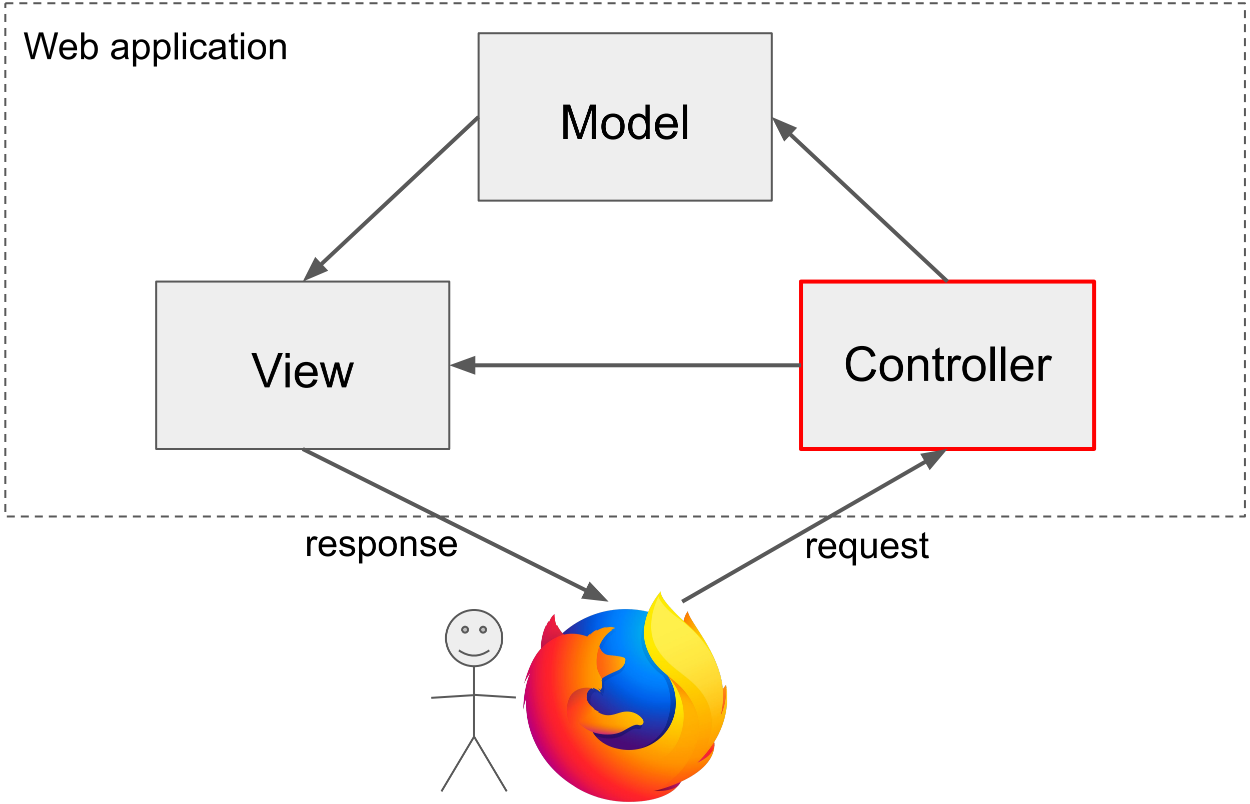
There’s nothing magical about this. It’s just a logical way to organize the code in an application with a data source and a user interface.

- You have code that knows about the data,
- Code that knows about the interface,
- And then code that mediates between them.

MVC Information Flow



MVC Information Flow



What’s a Request?

The web browser sends an HTTP(S) request to the server when you do something that requires new information or that pushes new information to the application. Like clicking a link, or filling in a form and submitting it.

e.g. a GET request:

```
GET /Institute-Materials-2020/ HTTP/2
Host: pittsburgh-neh-institute.github.io
user-agent: curl/7.79.1
accept: */*
```

What’s a Request?

So, the browser sends a request (literally some plain text, formatted according to the HTTP protocol. Besides GET requests, there’s also POST, PUT, DELETE, PATCH, and a few others.

And something on the server is listening, waiting for requests. When it gets one, it has to decide what to do with it.

THAT’S the Controller’s job: dispatching requests. It might send data from a POST to create or update a model, it might just switch to a view without any data behind it.