Client Side

1. Go To URL
2. HTTP Request
3. Route Request
4. Process Request
5. Build Response
6. HTTP Response
7. Parse Response
8. Render Page

Server Side
Client Side

- View (Templates)
- User Interactions

Server Side

- Controller (Routes)
- Model (Data)

HTTP Request

HTTP Response

CRUD Manipulations
Client Side

- View (Templates)
- User Interactions
- Two-Way Data Binding

Controller
- ViewModel
- Routes

Server Side

- Model (Data)
- CRUD Manipulations
Client Side

- View (Templates)
  - User Interactions
  - Two-Way Data Binding

Server Side

- Controller
  - View/Model
  - Routes
- Model (Data)
  - CRUD Manipulations
  - CRUDD Manipulations

- User Interactions
How it feels to learn JavaScript in 2016

Cool. I need to create a page that displays the latest activity from the users, so I just need to get the data from the REST endpoint and display it in some sort of filterable table, and update it if anything changes in the server. I was thinking maybe using jQuery to fetch and display the data?

-Oh my god no, no one uses jQuery anymore. You should try learning React, it's 2016.

Oh, OK. What's React?

-It's a super cool library made by some guys at Facebook, it really brings control and performance to your application, by allowing you to handle any view changes very easily.

That sounds neat. Can I use React to display data from the server?

-Yeah, but first you need to add React and React DOM as a library in your webpage.

Wait, why two libraries?

One is the actual library and the second one is for manipulating the DOM.
✓ Small surface area (i.e., fewer new things to learn).
✓ Thriving ecosystem + community (i.e., easy to find help).
✓ Better modularity + separation of concerns via components.