Web Architecture Overview

The Basics

We can generally divide website 'stuff' into two parts. The client and the server. In a slightly simplified world the client is your browser and the server is some computer off in the far far away that has data that your client wants. It is more complex than that because there can be clients that request stuff from web servers that are not browsers and sometimes servers can request stuff from other servers. Sort of turning a server into a client temporarily. But the major concept in the same, one computer is asking for stuff and the other computer is serving that stuff.

The Client-Server Relationship

The Client

For the purposes of this class we can think of the client as the web browser. When we say that things happen client side we are referring to things that happen within the browser without the need to contact a server. So we can have pages that are interactive and we can have that all be done client side. An example of a client side interaction can be seen with the jQuery UI Accordion. When you click on a tab it will expand that tab and contract the rest. All of that logic and interaction is happening on the client with no need to communicate to the server. So we say it is a client side interaction or that it is happening client side.

The Server

The server is a bit harder to pin down. To excessively simplify the server is the thing the client is requesting data from. In general this is probably going to be some computer elsewhere like when you are accessing http://news.google.com. But it can also be a server application you are accessing locally. In that case the address would look more like http://localhost:3000. localhost refers to accessing a resource on the same computer. Finally you can use your own computers file system as a server. It can't do anything special like change content on a page, it can only serve static files without changing them. It would look like file:///D:/Documents/hello-world.html. This is just directly accessing a file on your computer. The important thing to know is that other people (like the instructor and graders) can only access your page if your address looks like the first address. If it is being accessed as localhost or with file:// then only you can access it.

When stuff is server side then it has access to all the data on the server. So a weather site might have the most recent weather data on the server or a shopping website might have the most up to date inventory. When a page is generated on the server it can use this to generate the page that will be sent back to the client. But once it gets sent to the client there is nothing more the server can do to change the page unless the client makes another request.

More on the Client:

In the context of this class, it will generally be the browser, but it may also include programs and other techniques involving servers to create a larger client.

Clients are everything that happens in the computer. The client will ask the server for new information and then the server delivers the requested data. And all of that information is stored on the server or is gathered by the server and disseminated by the client. There may be multiple servers available for data and it talks with information and that gathers data and sends it off to the client.

The lifecycle of the website is pretty simple but has an incredible amount of depth. There is banners, other information, what’s happening beneath the hood, and when we request the page, is when time starts to happen.

If you use the developer tools, and it will show up what is happening over the network. A request is sent, a server responds, and then it tells us a bunch of stuff that is important and that we had a response (304) – it happened and it’s in the cache. Status 200 means its loading. 400/500 errors are the things that are not okay and then we might get the language its in and when it was previously modified and when things came in.

Most of the information is just jargon. There are not too many steps, and somewhere there might be an error, and if you run into a problem, and when a page is loaded and displayed, you can find out where things went bad.