Myles Pruitt
09.18.2022
CS230

REST APIs

The client-server pattern satisfies software requirements for a few reasons. The client-server patterns essentially separate the server from its clients providing a layer of security, so the internal functions are not vulnerable to unauthorized users. In a web-based application, the client receives both input and outputs provided by the server while authorized users such as developers maintain and update clients' software. This is also a very efficient algorithm since it gives much control for the author to make adjustments as needed without interrupting client processes.

REST APIs is an application programming interface that establishes the same format of communication. The beauty of REST APIs is that they communicate via HTTP requests and perform standards of creating, reading, updating, and deleting records. REST APIs are also stateless meaning each request needs to have all necessary information before processing resulting in they do not require in server-side sessions. There is also an emphasis on client-server decoupling, where the server and client must be completely independent of each other. JSON is also the standard format protocol they used for encrypted transit communication, which prevents or mitigates malicious users.

Developers should make it a priority to manage performance, maintainability, and scalability for their clients. For instance, when adding more users to the database there should be a set of best practices in place such as normalization which would help with the maintainability and performance of the database. More users require more requests and load on the database. If the database is not designed in a way efficiently algorithmically to handle thousands of requests it could cause performance issues on the client side or even cause the server to crash. This would make the company lose money or worst of all you could potentially lose clients.

Features that could be included, would be a ranking system. It's important for some people to feel some level of competition, which would drive people to constantly challenge themselves. Another feature to consider is achievements for completionists and casual users just in case of competition between other players is not their drive. To make the app more profitable we could include some level of monetization where a banner or Interstitial banner would appear after a set amount of plays, this could be used to develop a second installment of the game or build another game.

If the Gaming room asked me to host the application on a fourth or fifth client I would have to consider the player base and the technology stack to see if it's compatible. There is always a casual gaming market for most platforms but some platforms have a larger casual player base than others which would be a huge factor in decision making.