Fabian Rodriguez

Module 4 Journal

My Home

CS-230-H6818 Operating Platforms 21EW6

07/25/2021

* **Client-Server Pattern: Discuss how the client-server pattern can be used to satisfy software requirements and efficiently solve a problem**. Specifically, the web-based game application must be able to be run on multiple operating platforms.

Client-Server can satisfy software requirements and efficiently solve problems related to project that need multiple Clients. Client-server pattern is majorly strong when the software is intended to be accused from different platforms. It is highly beneficial for the developer since once the server is fully implement, it can be accessed from multiple different platforms without updated/changing how the server works. The reason for this is because although the server and the client are connected, we only need to change the frontend(client-side). Since the server is centralized location and can be accessed from multiple different clients it’s more manageable to scale and maintain, therefore, being more efficient that other approach. Using this approach enables the developers to focus on the client side without limitations since they can use any program to access the resources from the server.

* **Server Side**: You have developed the application from the server side. **Discuss how the server side provides communication to the client side with REST API style**.

By using a REST API, we connect the Client to the Server in the form of response and request. The URL are usually called the request and the data we get back is the response. If we wanted to get the user of our API our request would look similar to [**www.example.com/users/**](http://www.example.com/users/)**,** and our response would look similar to **{id: 1, name: ”John Doe”}**, as we can see in the previous example. We made a request the “example” URL and got a response with only one user and its id. Note that our response was return a JSON. API responses is usually return in JSON or XML, although more formats are available. The previous description is the most basic for of how an API works. In summary our API functions just like the search of Google. Think of your search term as the URL (request) and the search results as your response. On the more advance end you can also perform POST/PUT/DELETE methods which can be used for adding, updating, or deleting data to the server respectively.

* **Client Side**: You wrote an application for multiple clients where the multiple environments can interact with the server. **Discuss what is required of the developers so that the application on all three clients can be used on the website**. Consider what next steps would entail to develop for the client side of the game application. For instance:
  + How would you add more users to the database?

When using an API to manage the connection between server and client we also get the benefit to create new data through the API. Using the POST method to create/add new users to the database should be the most feasible approach.

* + What other features might you include in the game app?

There is a plethora of features we can add to the game, like the ability to see how many players are currently playing., leaderboards and statistics of how well teams are performing. All of this can be achieved with all the data we currently have, or in any case just adding some other stuff like how long each game takes and keeping track of team wins/loses.

* + What if The Gaming Room asked you to host the application on a fourth and fifth client? For example, on Xbox and PS4.

Since we already have the API in place what is left is to understand how this platform Client-side works. Once we figured that out, we can connect the client to our already in place API and perform our usual request as necessary.