Michael Ross

CS-230

Jan 30, 2022

Journal 4-4

REST and Client – Server Reflections

Section A – Client-Server Pattern:

The client-server pattern will be especially helpful in solving the technical constraints of the Gaming Room application. One of the requests is that it supports web browsers as well as Android clients. To do this, the client-server pattern can be introduced. The server can host a series of APIs which interact with the data in the form of players, game objects, and other game data through the GameController singleton. The web, Android and other mobile clients will simply interact with the APIs which expose the game constructs. Since APIs are supported by most clients, this will allow for future extensibility as well.

Section B – Server Side:

The server side runs a Java Application which interacts with various clients using REST APIs. The REST APIs expose various parts of the game through paths. These paths are implemented using annotations and written code. The DropWizard framework is utilized to make things like Authentication and Authorization much easier. For example, to query a list of players the path “/gameusers” could be utilized. Authentication and authorization can be added to this path through use of the @Auth and @RolesAllowed annotations.

Any functionality of the game can be exposed through these APIs. Since the APIs utilize standard HTTP protocol to communicate, any client can access them and interact with the game. Authorization and Authentication are also provided through these APIs to ensure the application stays secure.

The APIs provided by the server form a contract with the clients accessing the REST APIs. In essence the server is telling the clients that if they provide certain information on a given path, the server will provide the correct data back.

Section C – Client Side:

For the clients to access the data and methods that the server provides, they must authenticate and utilize the REST API endpoints provided by the server. They must abide by the contract in place between server and client through the APIs. To Create, Read, Update, or Replace information on the server they provide the correct information on the correct HTTP path. Because Android application environments and Web Application environments are both able to send HTTP requests, this makes programming for multiple operating systems and clients much easier.

Currently only GET operations are implemented for the clients. In order to perform next steps like adding users to the database and modify information, POST or PATCH operations need to be implemented. There should also be some kind of quality of service or prioritization to ensure that operations which are not critical to serving a running game are de-prioritized to ensure that games run as smoothly as possible. In addition, to replace Basic authentication that is currently in place, it would be great to add some kind of Single Sign-On to Google / Facebook accounts to make it easier for players to sign up / sign in.

If the Gaming Room were to ask for additional clients, it would be much easier to leverage the existing APIs and implement the View for these API calls in the client. The Model and Controller components are already implemented in the APIs and the server side database.