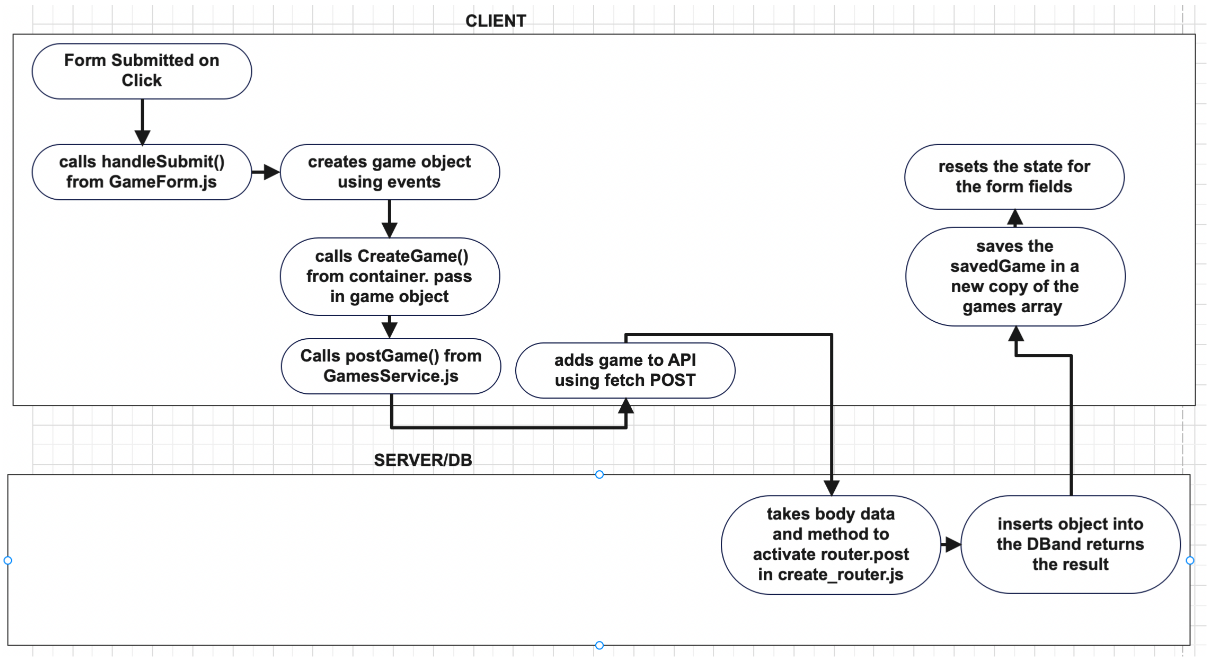
Task

Draw a diagram showing the dataflow through the application starting with a form submission, ending with the re-rendering of the page. This will involve a multi-direction data-flow with the client posting data to the server and the server sending data back to the client with the response. Detail the client, server and database in the diagram and include the names of the files involved in the process.


CLIENT
Form Submitted on Click
calls handleSubmit() from GameForm.js
creates game object using events
calls CreateGame() from container. pass in game object
Calls postGame() from GamesService.js
adds game to API using fetch POST
resets the state for the form fields
saves the savedGame in a new copy of the games array
SERVER/DB
takes body data and method to activate router.post in create_router.js
inserts object into the DBand returns the result

Questions

1. What is responsible for defining the routes of the `games` resource?
   createRouter function within the create_router.js creates the routes by passing in the games array.

2. What do you notice about the folder structure? Whats the client responsible for? Whats the server responsible for?
   The client is responsible for all front end and the Games Service the link to back end. Server sets up and runs the db and webserver.

3. What are the the responsibilities of server.js?
   Server sets up and runs the db and webserver.

4. What are the responsibilities of the `gamesRouter`?
   gamesRouter passes in the games collection to the createRoutes to set up the RESTful routes

5. What process does the the client (front-end) use to communicate with the server?
   GamesService.js consists of 3 fetch requests to communicate with back end routes. Show, Create, delete.

6. What optional second argument does the `fetch` method take? And what is it used for in this application? Hint: See [Using Fetch](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch) on the MDN docs
   The second argument is an init object that allows you to control a number of different settings. For our example, method, body and header. Which are used to create and delete games.

7. Which of the games API routes does the front-end application consume (i.e. make requests to)?

The API route is the base url http://localhost:9000/api/games/ but specifically uses api/games to then make requests.

8. What are we using the [MongoDB Driver](http://mongodb.github.io/node-mongodb-native/) for?

It allows the application to connect to MongoDB and work with the data. The driver features an asynchonous API which allows us to interact woth MongoDB using promises or via callbacks