

## Questions

What is responsible for defining the routes of the games resource?

The routes of the games resource are defined in the `create_router.js` server/helper. A `gamesRouter` is then created in `server.js`.

What do you notice about the folder structure? What's the client responsible for? What's the server responsible for?

The folder structure is composed by 2 main directories, `client`, responsible for Front-End stuff such as the Vue main app, related subcomponents, including the `GamesService.js` which is the component responsible for communicating to the back end (server).

The server directory contains back-end components, such as the database, any node modules required, the `create_router` file and `server.js`.

What are the responsibilities of `server.js`?

It allows the server to listen for requests on port 3000, and also creates the `gamesRouter` to process "game" requests.

What are the responsibilities of the `gamesRouter`?

`gamesRouter` is created in `server.js` and creates a router using the database collection of games (defined in `seeds.js`, in the `db` directory).

What process does the client (front-end) use to communicate with the server?

The client uses `GamesService.js` to communicate with the server.

What optional second argument does the `fetch` method take? And what is it used for in this application? Hint: See Using Fetch on the MDN docs

The `fetch()` method can optionally accept a second parameter, an init object. This is used in the `postGame` function that runs a `fetch` using the `baseUrl` as a first parameter and the new "game" coming from `GameForm.vue`'s form.

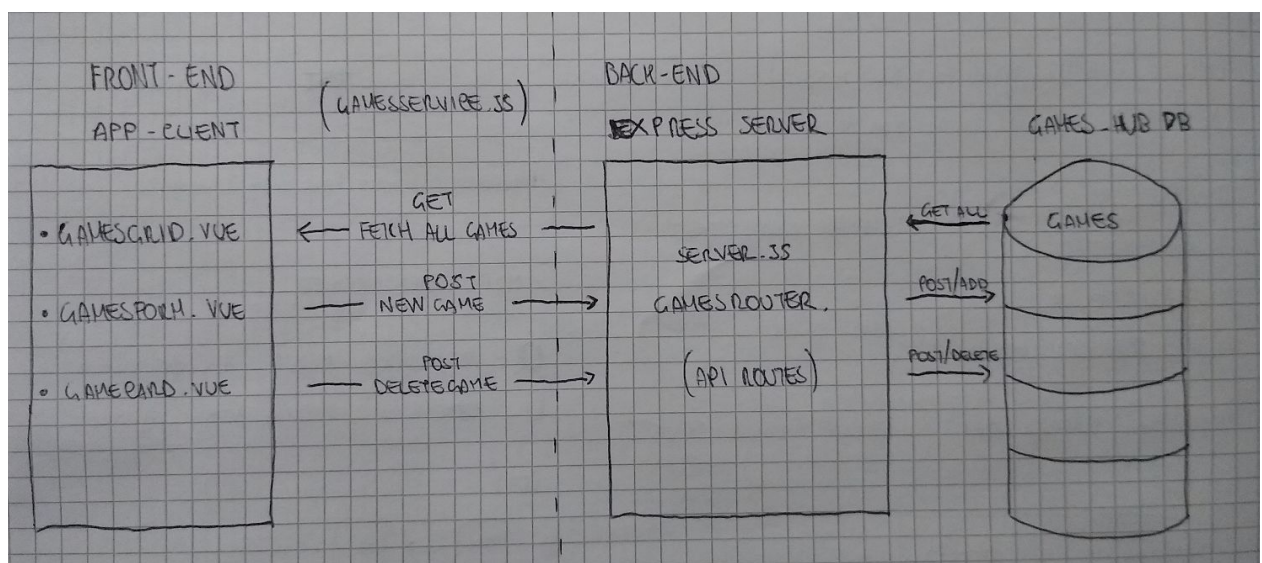
Which of the games API routes does the front-end application consume (i.e. make requests to)?

`router.GET` to display all games available, `router.POST` to add a new game and `router.DELETE` to delete a game.

What are we using the MongoDB Driver for?

The MongoDB driver allows the application to interact with the `games_hub` database and manipulate data.

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.



### Extensions

Why do we need to use ObjectId from the MongoDB driver?

It is used in `create_router.js` to refer to a specific object (game) present in the database collection by their ID. This way we can delete one from the games' page. It could also be used in case we wanted to get details on one game in particular or update a game's details, however these 2 functionalities do not seem to be included in the application.

Add to your diagram the dataflow for removing a game.

(See sketch/diagram above)