1. What is responsible for defining the routes of the games resource?
   gameRouter

2. What do you notice about the folder structure? Whats the client responsible for? Whats the server responsible for?
   Client fetches data from server to display on UI. Accepts user input to create new game and to delete a game.
   Server responds to client requests to GET data or accepts data from client by POST or DELETE. Queries database to update based on client requests, and responds to client with updated data.
   (i.e. client - front-end, server - back-end)

3. What are the the responsibilities of server.js?
   Connects to MongoDB to access game data,
   Uses express to accept client requests,
   Uses createRouter to process requests from client,
   Uses cors to allow transfer of data between ports,
   Listener on port 9000

4. What are the responsibilities of the gamesRouter?
   Sets the routes for the client-side requests (eg "/" to GET or POST game data),
   Queries database to pull data, or add or remove games

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

6. 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
   Second parameter is an init object with additional options, in this case the request method, request headers, and body of the request.

7. Which of the games API routes does the front-end application consume (i.e. make requests to)?
   "/api/games" & "/api/games/:id"

8. What are we using the MongoDB Driver for?
   Connection and CRUD operations