JavaScript for client-side and server-side programming

CS2003 - Week 9 Exercise

As with all lab exercises, this exercise is not directly assessed. It is intended to complement the material presented in lectures, and material from this session may turn up in the exam. You are encouraged to complete it in your own time if you do not finish it during the lab hour.

The aim this week is to experiment with some more aspects of web application programming. Specifically:

- Client-side state storage (‘cookies’) with localStorage.
- Asynchronous communication
- The use of AJAX.

As always, there is much to be gained in undertaking exercises together with one or two other people from the class, discussing what you observe, as well as attempting the tasks and exercises below. Please ***run****,* ***examine****, and* ***discuss*** *the code in the lab, and work on exercises and questions with others.*

1. Client-side programming

Please refer to the directories:

https://studres.cs.st-andrews.ac.uk/CS2003/Examples/web/node/web4/cookie-tute/

https://studres.cs.st-andrews.ac.uk/CS2003/Examples/web/node/web4/localstorage-tute/

For the code in cookie-tute/ and localstorage-tute/, compare the two mechanisms in terms of API, and visibility of the cookie/state.

Modify the textsize.js and textsize.html files so that the font can be chosen (from a selection of 4 or 5 fonts of your choice), and text colour can also be chosen, and both can be ‘saved’, to be re-instated when the page is revisited after the browser has been closed. Also include a popup/dialogue that confirms the user’s permission to do save these preferences.

2. Asynchronous programming

Look at the examples of asynchronous programming from the lectures and in the directory:

https://studres.cs.st-andrews.ac.uk/CS2003/Examples/web/node/node-events-tute/

Make sure you understand how they work. Try modifying the fetch example shown in the lectures so that it reads some JSON from the server and modifies the contents of a table. You might want to use your Pokémon objects from the last Web tutorial for this purpose.

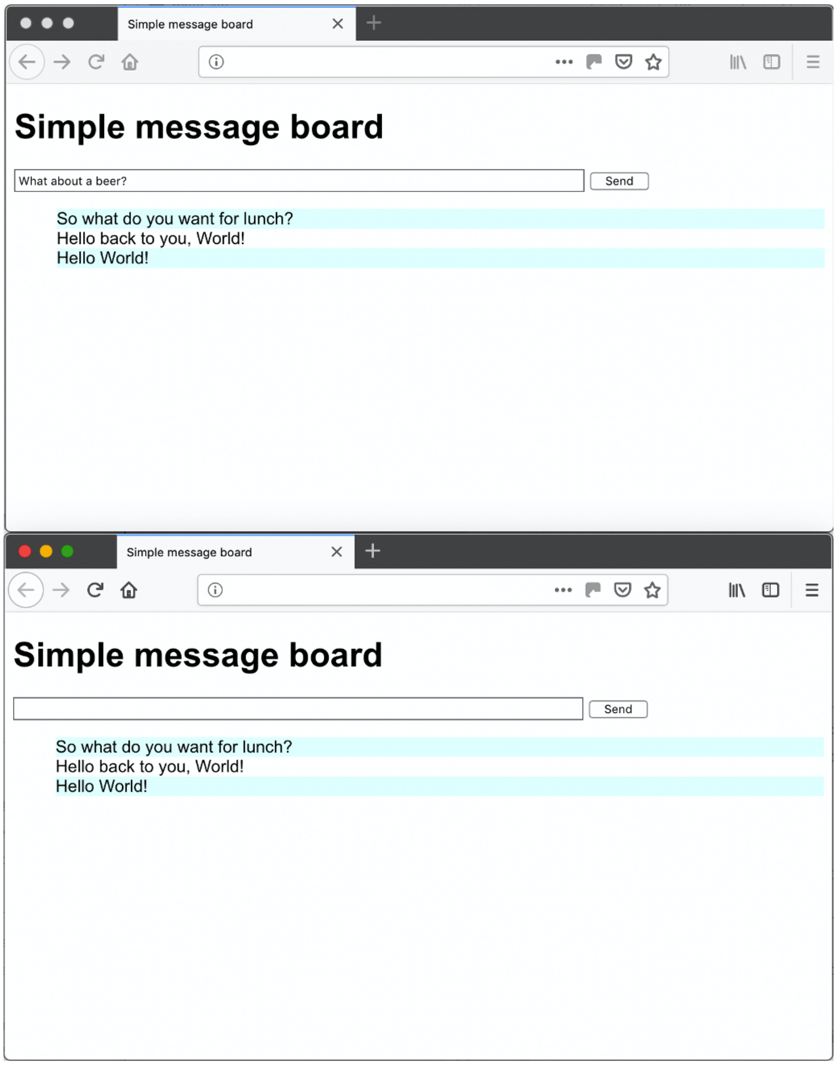
3. AJAX

This question is intended to help with the next Practical.

See directory: express-ajax-demo – this contains the code we developed during our Wednesday lecture (I did some tidying up after the lecture too, so it's not identical.)

Examine this code and make sure you understand how the server is set up, how the AJAX connections are triggered and handled. Copy the code to your host space and make sure you can run it. You will need to set up the nginx proxy as usual.

Use as much or as little of this code as you see fit. Work towards implementing a simple message board application. Ask a friend to access your web application, can you see each other's messages? You can add a simple handling of usernames to distinguish messages from different clients. See screenshots for a simple example design.



Simple message board

What about a beer?

So what do you want for lunch?
Hello back to you, World!
Hello World!

Simple message board

So what do you want for lunch?
Hello back to you, World!
Hello World!

Figure 1: Two Firefox browsers connected to the same server.