Client - Server Exercise

1. Quotes web server in node.js

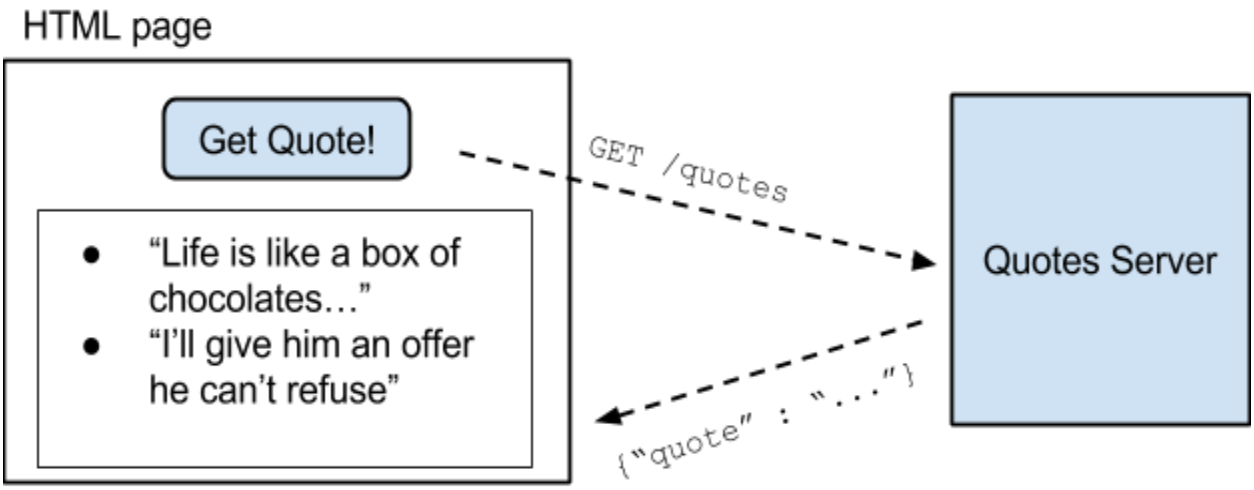
Create a simple Node.js HTTP server that receives requests to the route "/quote" and returns a random quote in JSON format. For example, if the server is running on http://localhost:3000, then sending an HTTP Get request to http://localhost:3000/quote will return a random quote i.e.

```
{
       "quote" : "Live as if you were to die tomorrow.
                    Learn as if you were to live forever. -Mahatma Gandi".
}
```

This task will require you to install node.js, familiarize yourself with javascript and node.js in particular, code the server and test it. You can use plain Node.js libraries for the server or use web server frameworks such as express.js. Focus on writing concise, clean code that's easy to understand and maintain.

2. Quotes client - HTML page

Create an HTML page that fetches quotes from the server. The page should include a button that when clicked, requests a quote from the server and appends the response to the page. The requests should be asynchronous, i.e. use AJAX requests so that the page doesn't reload after each request - use jQuery for this.



3. Deployment

Deploy the quotes server to a service which will enable you to expose it online. You can use hosting platforms such as Heroku, Modulus, Nodejitsu etc. (use the free signup tiers). The focus here is on speed rather than quality. Get it up and running as fast as you can so that it's open publicly and can be accessed by anyone.

4. Android App

Create an Android app that does the same as the Quotes HTML page. it should have a “Get Quote!” button at the top of the screen. Everytime the button is pressed a request should be sent and a spinner should be shown in the app. When the response with the quote returns, append it to a list below the button (just like in the HTML example app). This project will require you to install the Android SDK and Android Studio. When you’re done, pack the Android app into an .apk file and send it

General guideluines

- The code should be written in a high standard, documented and easily understandable.
- Everything needs to be “shipped” - the server needs to be live and public, and the rest of the code should be in a public repo on Github, or alternatively sent by email with running instructions.
- Investing in a nice look’n’feel in the HTML page and the Android App will give extra points, but only given that everything else works properly.