The following exercises are related to the use of Node/Express.

1. Create a git repository for your answers to this problem sheet. Push the repository to GitHub. Make a commit and push it to GitHub after each exercise.

2. In our last Angular lab we built an Angular application that used Angular routing. The final solution to the lab can be found at `https://github.com/MeanApplication/lab2_Routing`. Clone this application if you did not finish the application last week. To clone an the application
   git clone `https://github.com/MeanApplication/lab2_Routing`

   if the application does not run you must install angular material

   ```
   npm install --save @angular/material @angular/cdk @angular/animations
   ```

3. Make a new folder in your Angular App called Backend and add a file to that folder called server.js. Develop a Server that uses the Express Framework to returns the following JSON data when a GET request is made to /api/posts.

   ```
   {
      "posts":[
         {
            "id":"fadf12421l",
            "title":"First server-side post",
            "content":"This is coming from the server"
         },
         {
            "id":"ksajflaj132",
            "title":"Second server-side post",
            "content":"This is coming from the server!"
         }
      ]
   }
   ```

4. Change the Angular app to read JSON data from the Node/Express server. You will need to add the following lines to your server to avoid a CORS error.

   ```
   app.use(function(req, res, next) {
       res.header("Access-Control-Allow-Origin", "*");
       res.header("Access-Control-Allow-Headers",
       "Origin, X-Requested-With, Content-Type, Accept");
       next();
     });
   ```

5. Change the Angular app so that it now makes a post request to the server (sending a ”post” object to the server). Also adding a post method on the Express Server that will

console log both the title and content of the object passed up by the Angular app. Again the post method should be at /api/posts url on the server.