

Task 3. Keeping notes in session

1. Execute node instead of lite-server

To run node, edit package.json in root application folder by changing

```
"start": "tsc && concurrently \"tsc -w\" \"lite-server\" ",
```

to

```
"start": "tsc && concurrently \"tsc -w\" \"node server/server.js\" ",
```

This way we will start node server instead of lite-server.

Otherwise you can install nodemon which will restart node server every time you when you are changing server.js script:

```
npm install nodemon
```

And then starting script will be looking this way:

```
"start": "tsc && concurrently \"tsc -w\" \"nodemon server/server.js\" ",
```

Now we can restart server with npm start. Since now Node is responsible for both client and server side. However, page autoreloading is not working anymore, so you will need to do it manually.

Then add this line to server.js:

```
app.use(express.static(path.join(__dirname, '..')));
```

This will make the root folder as static folder, i.e. index.html will be static resource served by Node server.

Change **notesUrl** in notes.component.ts to 'notes' because now we are working on the same server, so we don't need

2. Load notes from session

Add session middleware to server.js:

```
app.use(session({
  secret: 'angular_tutorial',
  resave: true,
  saveUninitialized: true
}));
```

Also we need to install express-session to Node server. Type this in server folder:

```
npm install express-session --save
```

And add session variable to the top of server.js:

```
var session = require('express-session');
```

It will install express-session and add it to package.json.

Change reading of notes to this:

```
app.get("/notes", function(req,res) {
  console.log("reading notes", req.session.notes);
  if (!req.session.notes) {
    req.session.notes = notes_init;
  }
  res.send(req.session.notes);
});
```

Now notes are loaded from the session or initialized by notes_init variable.

Define notes_init like this:

```
var notes_init = [  
  {text: "First note"},  
  {text: "Second note"},  
  {text: "Third note"}  
];
```

2. Add possibility to add notes to the session

Add this route to server.js:

```
app.post("/notes", function(req,res) {  
  var note = req.body;  
  console.log("adding note", req.session.notes);  
  req.session.notes.push(note);  
  res.end();  
});
```

To retrieve values from HTTP body, you need body-parser middleware. To install it, use

npm install body-parser --save

in server folder. Also add these lines to the beginning of server.js:

```
var bodyParser = require('body-parser')  
app.use(bodyParser.urlencoded({extended: true}));  
app.use(bodyParser.json());
```

Add method which sends the added note to the server to notes.component.ts:

```
addNote(note:Note) {  
  this.http.post(this.notesUrl, note).toPromise()  
    .then(response => console.log("note sent, response", response) );  
}
```

And add call to addNote() to add() method of NotesComponent.

Additional tasks

- 1) Implement remove note from session
- 2) Implement «Send to top» button in session