Task 2. Notes application with Angular2

Take initial application, create folder app.

1. Create and use NotesComponent

1) Create component AppComponent in app/app.component.ts. Define this template:

```
template: `
   <h1>Notes Angular 2 App</h1>
  <notes></notes>`
```

2) Create file app/notes.component.ts with this contents:

```
import {Component} from '@angular/core';
@Component({
   selector: 'notes',
   template: `Notes list:`
})
export class NotesComponent { }
```

3) Import NotesComponent: add to the import block in app.module.ts:

```
import { NotesComponent } from './notes.component';
```

4) Define required directive by adding this to declarations of @NgModule in app.module.ts:

```
declarations: [ …, NotesComponent ]
```

5) Execute and check that «Notes list» is shown in the application

2. Show notes in NotesComponent

1) Define interface for Note in notes.component.ts:

```
interface Note {
   text: string;
}
```

2) Add this code to the template:

```
<ul>
   <li *ngFor="let note of notes ">
      {{note.text}}
   </li>
</ul>
```

3) Define initial notes list in NotesComponent class:

```
export class NotesComponent {
   notes: Note[] = [
      {text:"Note one"},
      {text:"Note two"}
   ]
}
```

4) Execute and check that notes list is shown

3. Add possibility to add the note to the list

1) Add these lines to the template in notes.component.ts:

```
<textarea [(ngModel)]="text" ></textarea>
<button (click)="add()">Add</button>
```

2) Define text and add() method in NotesComponent class:

```
text: string

add() {
   let note = { text: this.text }
   this.notes.push(note);
   this.text = "";
}
```

3) Execute and look how it's working

4. Add possibility to remove the note from the list

1) Change the template to show notes this way:

```
<ul>
   <li *ngFor="let note of notes; let i=index">
      {{note.text}} <button (click)="remove(i)">remove</button>
   </li>
</ul>
```

2) Define method remove:

```
remove(idx) {
   this.notes.splice(idx,1);
}
```

3) Check the possibility to remove notes

5. Retrievieng data from the server

1) Change app.module.ts:

Add import

```
        import { HttpModule }   from '@angular/http';
```

Add HttpModule to imports in @NgModule:

```
        imports:     [ BrowserModule, HttpModule ]
```

2) Create server folder and put server.js to it.
  In server/server.js allow cross-origin requests (because lite-server and Node are running on different servers):

```
        app.get("/notes", function(req,res) {
          res.header("Access-Control-Allow-Origin", "*");
          res.header("Access-Control-Allow-Headers", "X-Requested-With");
          var notes = [
              {text: "First note"},
              {text: "Second note"},
              {text: "Third note"}
          ]
          res.send(notes);
        });
```

Install Express by typing in server folder:

```
        npm install express
```

Otherwise you can create package.json by typing **npm init**, and then execute

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

It will add express to package.json.

Execute server on port 8080:

```
        node server.js
```

3) Import Http to notes.component.ts:

```
        import { Http } from '@angular/http';
```

4) Define notesUrl:

```
        private notesUrl = 'http://localhost:8080/notes';  // URL to web api
```

5) Define   getNotes() method:

```
                      getNotes(): Promise<Note[]> {
           return this.http.get(this.notesUrl)
                  .toPromise()
                  .then(response => response.json() as Note[]);
        }
```

Also add this import to use toPromise operator (it's not added automatically):

```
                import 'rxjs/add/operator/toPromise';
```

This is needed for every rxjs operator you want to use

6) Add constructor to inject http and retrieve from server:

```
constructor(private http: Http) {
   this.getNotes().then(notes=>{
      this.notes=notes
      console.log(notes);
   });
}
```

7) Start server.js and see the notes loaded from the server

Additional tasks

1) Add «Send to top» button
2) Implement sending data to server to add a note