# Week 13: Coursework

## Angular Framework

1. This coursework starts from the solution we created for coursework #12. We need to modify it to place all the HTTP calls in an Angular Service. Having a Service is the best approach because we put all the HTTP calls in one place so we can inject the service to any component that requires data from the backend. The steps presented are based on the solution given for coursework #12. You must perform equivalent steps if you use your own coursework #12 solution.

<https://github.com/busyQA-java-developer/busyqa-12-angular-3-coursework-complete.git>

1. In Visual Studio Code, right-click the project’s name and open an integrated terminal. In the terminal run the following command to create an Angular Service.

*ng g service TaskService*

1. In the *task.service.ts*  file, you have to do the following:
   1. Import the HttpClient, map, environment and Task modules, then add a constructor and inject the HttpClient. Please add the lines in red to the *task.service.ts* file as described below.

*import { Injectable } from '@angular/core';*

*import { HttpClient } from '@angular/common/http';*

*import { map } from 'rxjs/operators';*

*import { environment } from '../environments/environment';*

*import { Task } from './task.model';*

*@Injectable({*

*providedIn: 'root'*

*})*

*export class TaskService {*

*constructor(private http: HttpClient) { }*

*}*

* 1. Modify the *environment.ts* file in the environments folder to add the *host* property that contains the backend’s URL:

export const environment = {

production: false,

host: 'https://test1-9621d-default-rtdb.firebaseio.com'

};

* 1. Copy the createTask(), fetchTasks(), and deleteTask() methods from the app component’s TS file to the task.service.ts file. Then modify the methods to remove the subscribe() method and return just the observable. The subscribe() method will be added later to the app component that uses the taskService. Also, replace the hard-coded backend URL with the host property from the environment object. The methods should look like this:

fetchTasks() {

return this.http

.get<Task[]>(environment.host + '/tasks.json')

.pipe(

map(responseData => {

const tasks = [];

for (const key in responseData) {

if (responseData.hasOwnProperty(key)) {

tasks.push({ ...responseData[key], idTask: key});

}

}

return tasks;

})

)

}

createTask(task: Task) {

return this.http

.post<{ id: string }>(

environment.host + '/tasks.json',

task

);

}

deleteTask(idTask: string){

return this.http

.delete(environment.host + `/tasks/${idTask}.json`);

}

1. In the *app’s* component TS file, you have to do the following:
   1. First, import the TaskService from the ‘*./task.service’* package, then remove the *HttpClient* and *map* Modules as they are no longer needed in the app component. The import statements in the app component should like the code below:

import { Component, OnInit } from '@angular/core';

import { Task } from './task.model';

import { TaskService } from './task.service';

* 1. Modify the constructor to inject the TaskService.

constructor(private taskService: TaskService) {}

* 1. Modify the createTask(), fetchTasks(), and deleteTask() methods to replace the HTTP call with the corresponding *taskService* method call. The methods should look like this:

createTask(task: Task): void {

this.taskService.createTask(task)

.subscribe(() => {

this.fetchTasks();

});

}

fetchTasks(): void {

this.taskService.fetchTasks()

.subscribe(tasks => {

console.log(tasks);

this.tasks = tasks;

});

}

deleteTask(idTask: string): void {

this.taskService.deleteTask(idTask)

.subscribe(()=> {

this.fetchTasks();

});

}

1. After completing the project, test it, and create a new repository on GitHub, finally commit and push your application to GitHub.
2. You can find the solution for the project in the Github repo:

<https://github.com/busyQA-java-developer/busyqa-13-angular-3-coursework-complete.git>

// End