import { Injectable } from '@angular/core';import **{ Headers, Http, Response } from '@angular/http';**

import 'rxjs/Rx';import { Observable } **from 'rxjs/Observable';**

**@Injectable()**

export class ServerService {

constructor(private http: Http) {}

**storeServers**(servers: any[]) {

const headers = **new Headers**({'Content-Type': 'application/json'});

return **this.http.put**('https://udemy-ng-http.firebaseio.com/data.json',

servers,

{headers: headers});

}

getServers() {

return this.http.get('https://udemy-ng-http.firebaseio.com/data')

**.map(**

**(response: Response) => {**

const data = **response.json()**;

for (const server of data) {

server.name = 'FETCHED\_' + server.name;

}

**return data;**

}

)

**.catch(**

**(error: Response) => {**

**return Observable.throw(**'Something went wrong');

}

);

}

getAppName() {

return this.http.get('https://udemy-ng-http.firebaseio.com/appName.json')

.map(

(response: Response) => {

**return response.json();**

}

);

}

}

import { Component } from '@angular/core';import { Response } from '@angular/http';import { ServerService } from './server.service';

@Component({selector: 'app-root',templateUrl: './app.component.html',styleUrls: ['./app.component.css']})

**export class AppComponent {**

**appName = this.serverService.getAppName();**

**servers = [{**name: 'Testserver',

capacity: 10,

id: this.generateId()},

{ name: 'Liveserver',

capacity: 100,

id: this.generateId() **}];**

**constructor(private serverService: ServerService) {}**

**onAddServer**(name: string) {

**this.servers.push({**

name: name,

capacity: 50,

id: this.generateId()

});}

**onSave**() {

this.serverService.storeServers(this.servers)

**.subscribe(**

(response) => console.log(response),

(error) => console.log(error)

); }

**onGet**() {

this.serverService.getServers()

**.subscribe(**

**(servers: any[]) =>** this.servers = servers,

(error) => console.log(error)

);}

private **generateId**() {

return Math.round(Math.random() \* 10000);

}}

-------------

<**h1>{{ appName | async }}</**h1>