Edit

Using GetPeople Method From Angular Component

Now, we can switch to the client side and use GetPeople method to show a list of people on the UI.

Service Proxy Generation

First, run (prefer Ctrl+F5 to be faster) the server side application (.Web.Host project). Then run **nswag/refresh.bat** file on the client side to re-generate service proxy classes (they are used to call server side service methods).

Since we added a new service, we should add it to **src/shared/service-proxies/service-proxy.module.ts**. Just open it and add **ApiServiceProxies.PersonServiceProxy** to the providers array. This step is only required when we add a new service. If we change an existing service, it's not needed.

Angular-Cli Watcher

Sometimes angular-cli can not understand the file changes. In that case, stop it and re-run **npm start** command.

PhoneBookComponent Typescript Class

Change **phonebook.component.ts** as like below:

```
import { Component, Injector, OnInit } from '@angular/core';
import { AppComponentBase } from '@shared/common/app-component-base';
import { appModuleAnimation } from '@shared/animations/routerTransition';
import { PersonServiceProxy, PersonListDto, ListResultDtoOfPersonListDto } from '@s

@Component({
```

Feedback

```
export class PhoneBookComponent extends AppComponentBase implements OnInit {

    people: PersonListDto[] = [];
    filter: string = '';

    constructor(
        injector: Injector,
        private _personService: PersonServiceProxy
    ) {
        super(injector);
    }

    ngOnInit(): void {
        this.getPeople();
    }

    getPeople(): void {
        this._personService.getPeople(this.filter).subscribe((result) => {
            this.people = result.items;
        });
    }
}
```

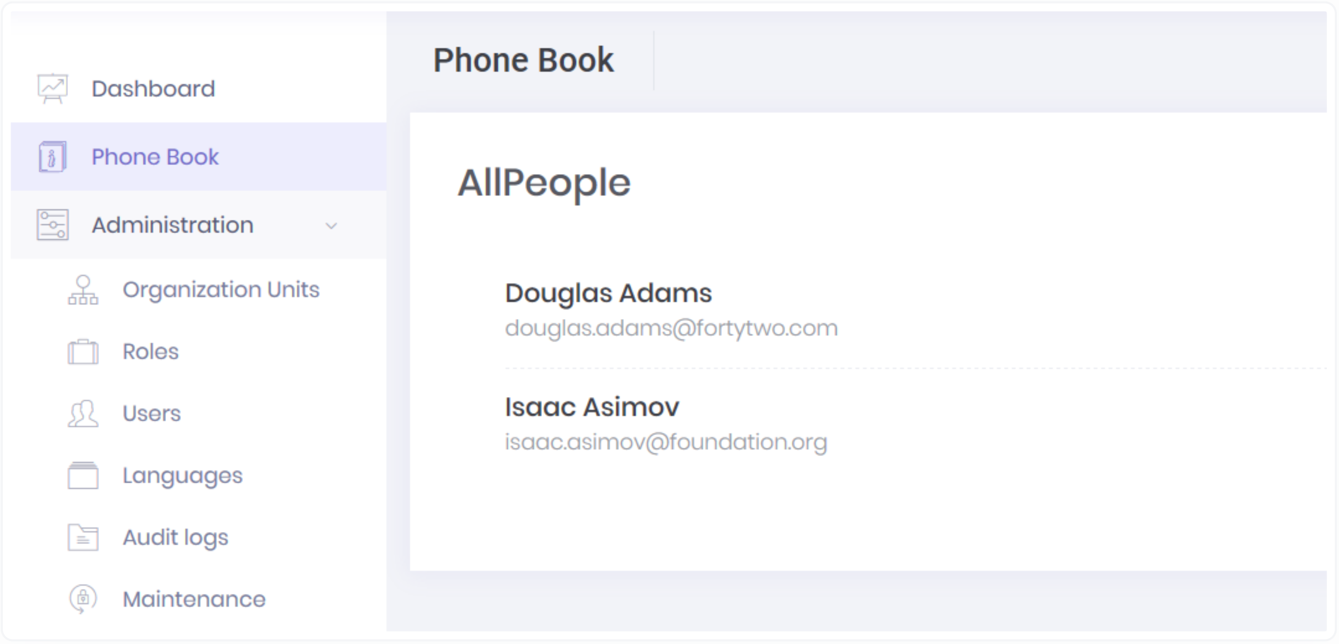
We inject **PersonServiceProxy**, call its **getPeople** method and **subscribe** to get the result. We do this in **ngOnInit** function (defined in Angular's **OnInit** interface). Assigned returned items to the **people** class member.

Rendering People In Angular View

Now, we can use this people member from the view, **phonebook.component.html**:

```
<div [@routerTransition]>
    <div class="kt-content  kt-grid__item kt-grid__item--fluid kt-grid kt-grid--hor
        <div class="kt-subheader kt-grid__item">
            <div class="kt-container ">
                <div class="kt-subheader__main">
                    <h3 class="kt-subheader__title">
                        <span>{{"PhoneBook" | localize}}</span>
                    </h3>
    <div class="kt-container kt-grid__item kt-grid__item--fluid">
      <div class="kt-portlet kt-portlet--mobile">
        <div class="kt-portlet__body  kt-portlet__body--fit">
          <h3>{{"AllPeople" | localize}}</h3>
          <div *ngFor="let person of people">
            <div class="row kt-row--no-padding align-items-center">
              <div class="col">
                <h4>{{person.name + ' ' + person.surname}}</h4>
                <span>{{person.emailAddress}}</span>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

We simply used **ngFor** directive to loop and render people data. See the result:



Phone Book

AllPeople

Douglas Adams
douglas.adams@fortytwo.com

Isaac Asimov
isaac.asimov@foundation.org

We successfully retrieved list of people from database to the page.

We normally use a javascript based rich table/grid library to show tabular data, instead of manually rendering data like that. For example, we used TurboTable library to show users on the Users page of ASP.NET Zero. Always use such components since they make things much more easier and provides a much better user experience.

We did not use a table component here, because we want to show basics of Angular instead of going details of a 3rd party library.

Next

- Creating a New Person