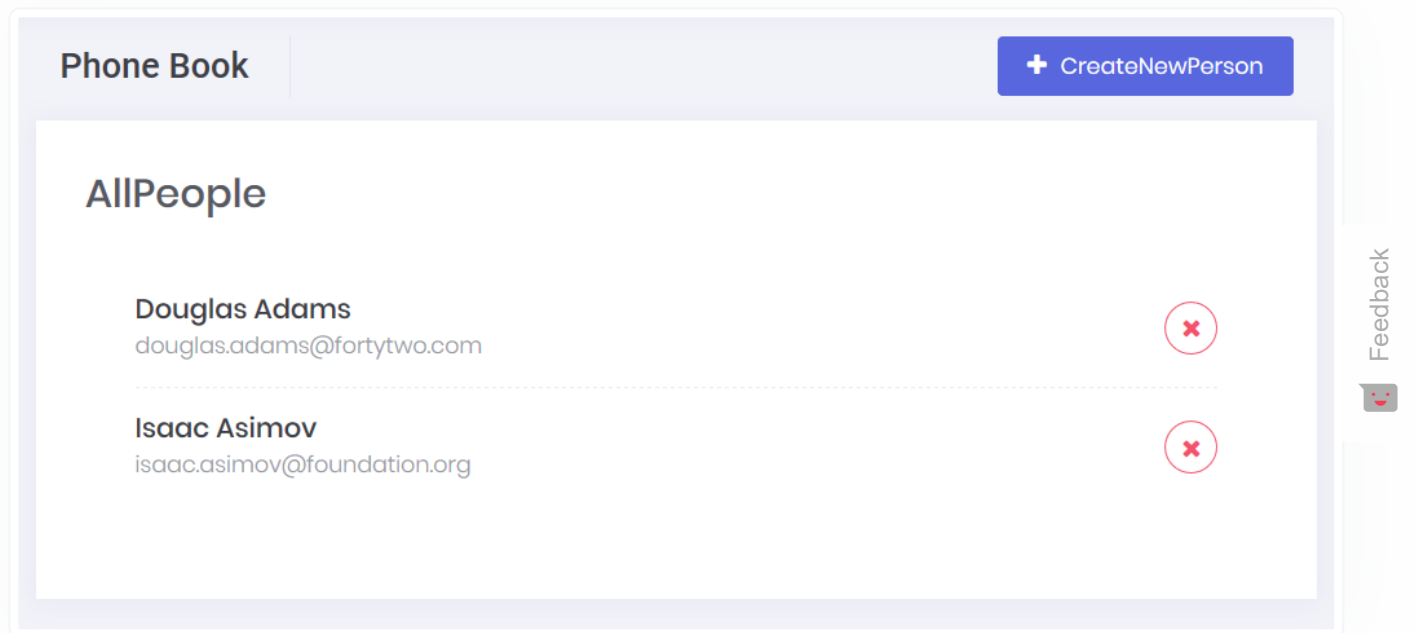




# Deleting a Person

Let's add a delete button in people list as shown below:



We're starting from UI in this case.

## View

We're changing **phonebook.component.html** view to add a delete button (related part is shown here):

```
...
<h3>{{"AllPeople" | localize}}</h3>
<div class="row kt-row--no-padding align-items-center" *ngFor="let person of people"
  <div class="col">
    <h4>{{person.name + ' ' + person.surname}}</h4>
    <span>{{person.emailAddress}}</span>
  </div>
  <div class="col kt-align-right">
```

```
<i class="Ta Ta-times"></i>
</button>
</div>
</div>
...
```

We simply added a button which calls **deletePerson** method (will be defined) when it's clicked. You can define a permission for 'deleting person' as we did for 'creating person' above.

## Style

We're using **LESS** files for styling the components. We created a file named **phonebook.component.less** (in phonebook folder) with an empty content.

```
/* styles */
```

And adding the style to the **phonebook.component.ts** Component declaration:

```
@Component({
  templateUrl: './phonebook.component.html',
  styleUrls: ['./phonebook.component.less'],
  animations: [appModuleAnimation()]
})
```

Now, we can now see the buttons, but they don't work since we haven't defined the deletePerson method yet.

## Application Service

Let's leave the client side and add a DeletePerson method to the server side. We are adding it to the service interface, **IPersonAppService**:

```
Task DeletePerson(EntityDto input);
```





```
[AbpAuthorize(AppPermissions.Pages_Tenant_PhoneBook_DeletePerson)]
public async Task DeletePerson(EntityDto input)
{
    await _personRepository.DeleteAsync(input.Id);
}
```

## Service Proxy Generation

Since we changed server side services, we should re-generate the client side service proxies via NSwag. Make server side running and use refresh.bat as we did before.

## Component Script

Now, we can add **deletePerson** method to **phonebook.component.ts**:

```
deletePerson(person: PersonListDto): void {
    this.message.confirm(
        this.l('AreYouSureToDeleteThePerson', person.name),
        isConfirmed => {
            if (isConfirmed) {
                this._personService.deletePerson(person.id).subscribe(() => {
                    this.notify.info(this.l('SuccessfullyDeleted'));
                    _remove(this.people, person);
                });
            }
        }
    );
}
```

It first shows a confirmation message when we click the delete button:





## ASP.NET CORE ANGULAR

DOCUMENTS



### Are you sure?

AreYouSureToDeleteThePerson

Cancel

Yes

Feedback



If we click Yes, it simply calls **deletePerson** method of **PersonAppService** and shows a **notification** if operation succeed. Also, removes the person from the person array using [lodash-es](#) library. We also added an import statement before the @Component declaration:

```
import { remove as _remove } from 'lodash-es';
```

## Next

- [Filtering People](#)

