



Edit Mode For People

Now we want to edit name, surname and e-mail of people:

EditPerson

Name

Douglas

Surname

Adams

Email address

douglas.adams@fortytwo.com

Cancel Save

Feedback

First of all, we create the necessary DTOs to transfer people's id, name, surname and e-mail. We can optionally configure auto-mapper, but this is not necessary because all properties match automatically. Then we create the functions in PersonAppService for editing people:

```
[AbpAuthorize(AppPermissions.Pages_Tenant_PhoneBook_EditPerson)]
public async Task<GetPersonForEditOutput> GetPersonForEdit(GetPersonForEditInput input)
{
    var person = await _personRepository.GetAsync(input.Id);
    return ObjectMapper.Map<GetPersonForEditOutput>(person);
}

[AbpAuthorize(AppPermissions.Pages_Tenant_PhoneBook_EditPerson)]
```



ASP.NET CORE ANGULAR

DOCUMENTS

```
person.name = input.name;
person.Surname = input.Surname;
person.EmailAddress = input.EmailAddress;
await _personRepository.UpdateAsync(person);
}
```

Then we add configuration for AutoMapper into CustomDtoMapper.cs like below:

```
configuration.CreateMap<Person, GetPersonForEditOutput>();
```

View

Create edit-person-modal.component.html:

```
<div bsModal #modal="bs-modal" (onShown)="onShown()" class="modal fade" tabindex="-1">
  <div class="modal-dialog modal-lg">
    <div class="modal-content">
      <form *ngIf="active" #personForm="ngForm" novalidate (ngSubmit)="save()">
        <div class="modal-header">
          <h4 class="modal-title">
            <span>{{"EditPerson" | localize}}</span>
          </h4>
          <button type="button" class="close" (click)="close()" aria-label="Close">
            <span aria-hidden="true">&times;</span>
          </button>
        </div>
        <div class="modal-body">

          <div class="form-group">
            <label>{{"Name" | localize}}</label>
            <input #nameInput class="form-control" type="text" name="name" [(ngModel)]="person.name">
          </div>

          <div class="form-group">
            <label>{{"Surname" | localize}}</label>
            <input class="form-control" type="email" name="surname" [(ngModel)]="person.surname">
          </div>

        </div>
      </form>
    </div>
  </div>
</div>
```





ASP.NET CORE ANGULAR

DOCUMENTS

```
<input class="form-control" type="email" name="emailAddress" [(ngModel)]
</div>

</div>
<div class="modal-footer">
  <button [disabled]="saving" type="button" class="btn btn-secondary" (click)
  <button type="submit" class="btn btn-primary" [disabled]="!personForm.form
</div>
</form>
</div>
</div>
</div>
```

Add those lines to **phonebook.component.html**::

// Other Code lines...

```
<button (click)="editPerson(person)" title="{{ 'Edit' | localize }}" class="b
  <i class="fa fa-plus"></i>
</button>
<button *ngIf="'Pages.Tenant.PhoneBook.EditPerson' | permission" (click)="e
  <i class="fa fa-pencil"></i>
</button>
<button id="deletePerson" (click)="deletePerson(person)" title="{{ 'Delete' |
  <i class="fa fa-times"></i>
</button>
```

// Other Code lines...

```
<createPersonModal #createPersonModal(modalSave)="getPeople()"></createPersonModal>
<editPersonModal #editPersonModal (modalSave)="getPeople()"></editPersonModal>
```

Feedback



Controller

Create edit-person-modal.component.ts:



ASP.NET CORE ANGULAR

DOCUMENTS

```
import { PersonServiceProxy, EditPersonInput } from '@shared/service-proxies/service-proxies';
import { AppComponentBase } from '@shared/common/app-component-base';

@Component({
  selector: 'editPersonModal',
  templateUrl: './edit-person-modal.component.html'
})
export class EditPersonModalComponent extends AppComponentBase {

  @Output() modalSave: EventEmitter<any> = new EventEmitter<any>();

  @ViewChild('modal') modal: ModalDirective;
  @ViewChild('nameInput') nameInput: ElementRef;

  person: EditPersonInput = new EditPersonInput();

  active: boolean = false;
  saving: boolean = false;

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

  show(personId): void {
    this.active = true;
    this._personService.getPersonForEdit(personId).subscribe((result) => {
      this.person = result;
      this.modal.show();
    });
  }

  onShown(): void {
    // this.nameInput.nativeElement.focus();
  }

  save(): void {
    this.saving = true;
    this._personService.editPerson(this.person)
      .subscribe(() => {
```





ASP.NET CORE ANGULAR

DOCUMENTS

```
    });  
    this.saving = false;  
  }  
  
  close(): void {  
    this.modal.hide();  
    this.active = false;  
  }  
}
```

Add those lines to **phonebook.module.ts**:

```
import { EditPersonModalComponent } from './edit-person-modal.component';  
  
// Other Code lines...  
  
declarations: [  
  PhoneBookComponent,  
  CreatePersonModalComponent,  
  EditPersonModalComponent  
]  
  
// Other Code lines...
```

Feedback



Next

- [Multi Tenancy](#)