

# ClubAdministration

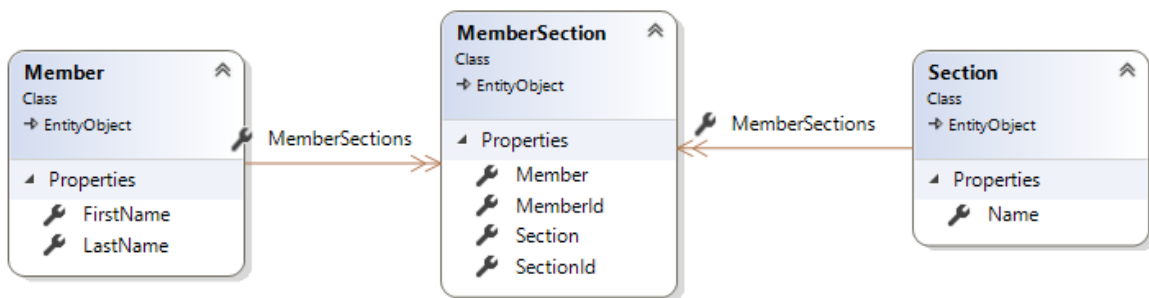
## Lehrziele

- Wiederholung ASP.NET Core Razor Pages
- Wiederholung ASP.NET Core WebApi
- Wiederholung Entity Framework Core
- WPF Validierung

Es ist eine einfache Mitgliederverwaltung für einen Verein mit mehreren Sektionen zu erstellen. Ein Mitglied (Member) kann in mehreren Sektionen (Section) angemeldet sein. Die Zuordnung wird über die assoziative Entität MemberSection verwaltet.

## Core

Die Entitätsklassen sind bereits angelegt. Auch die Annotationen zur Definition (inkl. der Migrationen) der Datenbank sind bereits implementiert.



Im Corelayer sind die Contracts für die Repositories bedarfsgerecht zu erweitern

## Import / Persistence

Es werden die 22 Mitglieder mit ihren 41 Mitgliedschaften in 6 Sektionen aus der Datei members.csv importiert.

Die Migration wurde bereits angelegt und muss nur verwendet werden, wenn an den Entitätsklassen Änderungen vorgenommen werden:

- Persistence als StartupProject
- Add-Migration InitialMigration
- UpdateDatabase mit Kontrolle, ob DB angelegt wurde
- Die Daten über die ImportConsole importieren.

Die ConnectionStrings wurden in den relevanten Projekten schon in der appSettings.json festgelegt.

Die Repositories müssen noch mit den fehlenden Methoden erweitert werden.

```

C:\Program Files\dotnet\dotnet.exe
Import der Sections und Members in die Datenbank
Datenbank löschen
Datenbank migrieren
Members werden von members.csv eingelesen
Es wurden 22 Members eingelesen
Es wurden 6 Sections eingelesen
Es wurden 41 Mitgliedschaften eingelesen. Speichern in Datenbank ...
69 Datensätze wurden in Datenbank gespeichert!
Beenden mit Eingabetaste ...
  
```

## ASP.NET Core Razor Pages (ClubAdministration.Web)

Ergänzen Sie das Template um die entsprechenden Pages.

### Page „Index“

#### Anforderungen:

1. Auswahl der Sektion über Select-Tag
  - a. Absenden des Formulars direkt nach der Auswahl (Spezialistenaufgabe) oder alternativ über einen Submit-Button
2. Tabelle der Sektionsmitglieder
3. Spalten der Tabelle
  - a. Mitgliedernamen
  - b. Anzahl der Sektionen, in denen die Person Mitglied ist
  - c. Link zu Bearbeitungsseite des Mitgliedes mit der Id als Querystring-Parameter
4. Sortierung der Tabelle nach Anzahl der Mitgliedschaften (absteigend)

LastName	FirstName	CountSections	Edit Member
Akgül	Oktay	3	<a href="#">Edit Member</a>
Axmann	Rene	3	<a href="#">Edit Member</a>
Ernst	Florian	3	<a href="#">Edit Member</a>
Ferrari	Rene	3	<a href="#">Edit Member</a>
Herbrik	Julia	3	<a href="#">Edit Member</a>
Klösler	Daniel	3	<a href="#">Edit Member</a>
Lehner	Dominic	3	<a href="#">Edit Member</a>
Leimgruber	Florian	2	<a href="#">Edit Member</a>
Rechberger	Simon	1	<a href="#">Edit Member</a>
Selig	Stefan	1	<a href="#">Edit Member</a>
Wolfsberger	Daniel	1	<a href="#">Edit Member</a>

### Page „Members/Edit“

#### Anforderungen:

Der Vor- und Zuname des Mitgliedes mit der per Querystringparameter übergebenen Id kann auf dieser Seite bearbeitet werden.

Folgende Randbedingungen sind dabei zu beachten:

- Sowohl FirstName als auch LastName dürfen nicht leer sein
- FirstName und LastName müssen aus zumindest zwei Zeichen bestehen
- Spezialistenaufgabe:
  - Es darf der geänderte Name nicht mit bereits in der Datenbank befindlichen anderen Mitgliedern übereinstimmen.

Nach erfolgreicher Bearbeitung des Mitgliedes ist wieder die Übersichtsseite aufzurufen.

First Name: Oktay

Last Name: Akgül

[Save](#)

[Back to List](#)

### Validierungsvorgaben:

The image shows three side-by-side screenshots of the 'Edit Member' form in ClubAdministration.Web, illustrating different validation states:

- Left Screenshot:** The 'FirstName' field is empty, and a red error message 'The FirstName field is required.' is displayed below it. The 'LastName' field contains 'Mustermann'. A 'Save' button and a 'Back to List' link are at the bottom.
- Middle Screenshot:** Both 'FirstName' and 'LastName' fields contain the letter 'M'. A red error message 'FirstNames minimum length is 2' is displayed below each field. A 'Save' button and a 'Back to List' link are at the bottom.
- Right Screenshot:** The 'FirstName' field contains 'Florian' and the 'LastName' field contains 'Ernst'. A red error message 'Es existiert bereits ein Mitglied mit dem Namen' is displayed below the 'LastName' field. A 'Save' button and a 'Back to List' link are at the bottom.

Each screenshot includes a copyright notice '© 2019 - ClubAdministration.Web' at the bottom.

Nach erfolgreicher Bearbeitung des Mitgliedes ist wieder die Übersichtsseite aufzurufen.

### ASP.NET WebApi (ClubAdministration.Web)

Die API-Controller sind im dazu vorgesehenen Verzeichnis zu implementieren.

#### Route „/api/members“

- Die Namen der Mitglieder sind alphabetisch sortiert auszugeben
- Der Rückgabetyt ist ein String-Array

Curl

```
curl -X GET "https://localhost:5001/api/Members" -H "accept: text/plain"
```

Request URL

```
https://localhost:5001/api/Members
```

Server response

Code	Details
200	<p>Response body</p> <pre>[   "Akçul Oktay",   "Axaann Rene",   "Dobroka Andreas",   "Ernst Florian",   "Ferrari Rene",   "Hengstschläger Valentin",   "Herbrik Julia",   "Hingerl Fabian",   "Hoffelner Lisa",   "Klösl Daniel",   "Lehner Dominic",   "Leimgruber Florian",   "Masinovic Faruk",   "Moser Michael",   "Rechberger Simon",   "Schörkmayr Lukas",   "Selig Stefan",   "Slaby Roman",   "Strasser Stefan",   "Watzl Tobias",   "Windhager Daniel",   "Wolfsberger Daniel" ]</pre> <p>Download</p>

### Route „/api/members/lastname/firstname/sections“

- Liefert die Namen der Sektionen des Mitglieds zurück
- Sortiert nach Sektionsnamen aufsteigend
- Gibt es das Mitglied nicht, wird NotFound zurückgegeben
- Gibt es das Mitglied, hat es aber keine Zuordnungen, wird ein leeres Array geliefert

Request URL

```
https://localhost:5001/api/Members/Ernst/Florian/sections
```

Server response

Code	Details
200	<p>Response body</p> <pre>[   "Fußball",   "Tennis",   "Wandern" ]</pre> <p>Download</p>

## ClubAdministration.Wpf

Über die Startseite der App wird, wie bei der Webanwendung die Übersicht der Mitglieder ausgegeben:

- „Die Spalte Sections gibt über die Anzahl der Sektionen des Mitglieds Auskunft
- Der Text des Bearbeiten-Buttons aktualisiert sich mit der Auswahl eines Mitglieds
- Sortierung wieder nach Zu- und Vornamen

### Dialog zum Bearbeiten eines Mitglieds:

Über diesen Dialog können die Daten eines bestehenden Mitglieds bearbeitet werden.

Die Validierungen sind entsprechend der Musterausgaben vorzunehmen.

**Top Screenshot (Validation Error):**

Lastname: q  
 Firstname: Steffen  
 Error: Lastname must be at least two characters long  
 Button: Save Member

**Bottom Screenshot (Duplicate Error):**

Lastname: Ernst  
 Firstname: Florian  
 Error: Member Ernst Florian existiert bereits  
 Button: Save Member

Club Administration

Section: Fußball

Lastname	Firstname	Sections
Ferrari	Rene	3
Max	Mustermann	1
Ernst	Florian	2
Wolfsberger	Daniel	1
Herbrik	Julia	3
Lehner	Dominic	3
Klösler	Daniel	3
Axmann	Rene	3
Akgül	Oktay	3
Leimgruber	Florian	2

Member Max Mustermann bearbeiten

- Die Daten der ursprünglich gewählten Sektion bleiben erhalten
- Die Änderungen sind sofort sichtbar (Tabelle, Button)

### Hinweise

- Verwenden Sie zum Sicherstellen der Datenkonsistenz (siehe Screenshots der Website) die notwendigen Validierungsattribute.
- Verwenden Sie dort wo sinnvoll DataTransferObjects
- Achten Sie auf eine korrekte Schichtentrennung (Core, Persistence und Web)
- Verwenden Sie das UnitOfWork-Muster
- Dependency Injection (IoC) verwenden
- Erweitern Sie, wo notwendig die Repositories