

Figaro

Allgemein

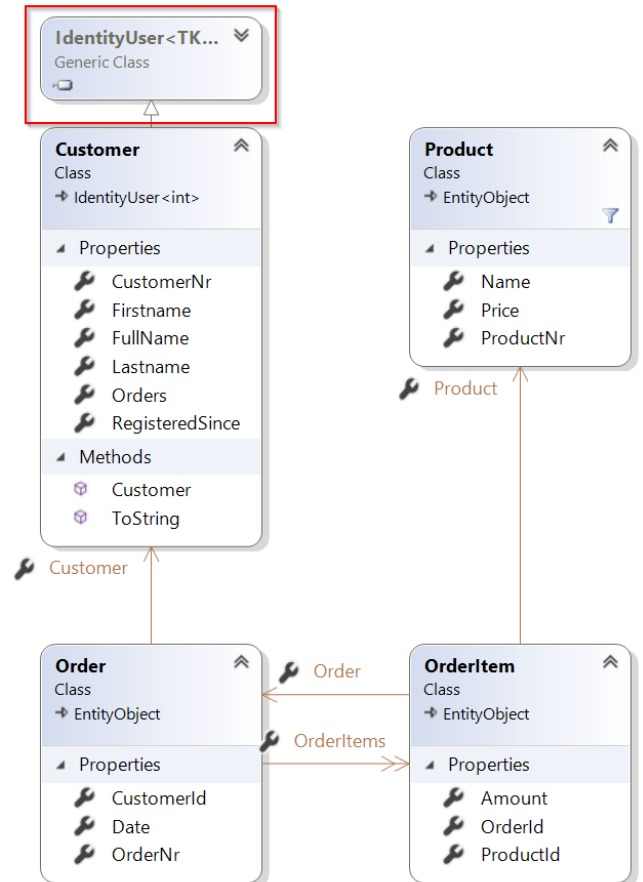
Es soll die bestehende Friseurbedarf-Software „Figaro“, welche die Kundenbestellungen verwaltet, per Microsoft Identity abgesichert werden, sodass sich Kunden in der Web Applikation anmelden können.

Persistenz

Bereiten Sie die Customer-Klasse zur Verwendung mit dem Identity Framework vor.

Erstellen Sie danach dann einen Entity Framework Migration und wenden Sie die Änderungen in der Datenbank an.

1. Keine Build Errors!
2. Default Project: „Figaro.Persistence“
3. Startup Project: „Figaro.Web“
4. NuGet Package Manager Konsole öffnen
5. Add-Migration InitialMigration
6. Update-Database
7. Datenbank kontrollieren („SQL Server Object Explorer“)



ImportConsole:

Der Großteil der Importlogik ist bereits vorhanden. Fügen Sie noch die Initialisierungen in der Customer-Klasse für das Microsoft Identity Framework hinzu sodass sich Customers im System anmelden können.

```

*****
Import
*****
Import der Figaro-Daten in die Datenbank
Datenbank löschen
Datenbank migrieren
Daten werden von csv-Dateien eingelesen
  12 Produkte eingelesen
Produkte werden in Datenbank gespeichert (persistiert)
Es wurden 12 Produkte gespeichert!
Es wurden 4 Kunden gespeichert!
Es wurden 8 Bestellungen gespeichert!
Es wurden 14 Bestellpositionen gespeichert!
  
```

Web Application (Razor Pages)

In einer Razor Page-Webanwendung sollen Bestellungen verwaltet werden können. Es können nur authentifizierte Benutzer die Bestellungsverwaltung aufrufen. Wenn der aktuell angemeldete Benutzer die Rolle „Admin“ besitzt so werden in der Startmaske alle Bestellungen sortiert nach der

Bestellübersicht

Nachname

Bestellnummer	Kunde	
2020/1	Herbert Berger	Bearbeiten
2020/2	Tanja Müller	Bearbeiten
2020/3	Herbert Berger	Bearbeiten
2020/4	Robert Sammer	Bearbeiten
2020/5	Martin Bregar	Bearbeiten
2020/6	Robert Sammer	Bearbeiten
2020/7	Tanja Müller	Bearbeiten
2020/8	Martin Bregar	Bearbeiten

Bestellnummer in einer Liste dargestellt.

Neben jeder Bestellung soll ein Link zum Bearbeiten (inkl. Löschen) der jeweiligen Bestellung platziert werden, außerdem soll auf der Startmaske ein Link zum Neuanlegen einer Bestellung verfügbar sein.

Im oberen Bereich soll nach einem Nachnamen gesucht werden können. Dazu kann ein Teil eines Nachnamens eingegeben werden, wenn auf den „Suchen“-Button geklickt wird, wird die Liste entsprechend aktualisiert:

Wenn der Benutzer die Rolle „Admin“ nicht besitzt, so sollen nur die persönlichen Bestellungen des Kunden angezeigt werden!

Bestellübersicht

Nachname b

Bestellnummer	Kunde
2020/1	Herbert Berger
2020/3	Herbert Berger
2020/5	Martin Bregar
2020/8	Martin Bregar

Beim Anlegen einer neuen Bestellung soll das Datum mit dem heutigen Tag vorbelegt sein und das Kundenfeld mit dem Namen des aktuell angemeldeten Kunden befüllt sein (read-only).

- Die Bestellnummer muss angegeben sein (client-seitige Validierung).
- Die Eingabe der Bestellnummer ist gegen die Datenbank zu validieren, da die Bestellnummer eindeutig sein muss!

Neue Bestellung anlegen

Bestellung

Bestellnummer

Bestellnummer darf nicht leer sein!

Datum

Kunde

Speichern

Zurück

Neue Bestellung anlegen

Bestellung

Bestellnummer

Bestellung mit Nummer 2020/1 existiert bereits.

Datum

Kunde

Speichern

Zurück

Bearbeiten/Löschen einer Bestellung

Bestellung bearbeiten

Bestellnummer 2020/1
Datum 01.01.2020
Kunde Herbert Berger
Bestellsumme 2105,20

Bestellpositionen

Produkt	Preis	Bestellmenge	
Wella Thermal Rundbürste	18,90	3	Entfernen
Wella Haartrockner Sahira Color	149,90	5	Entfernen
Hot Tools Professional Black Gold Volumiser-Set	129,90	10	Entfernen

Bestellposition hinzufügen

Hinzufügen

[Bestellung Löschen](#) | [Zurück](#)

In der Detailmaske zu einer Bestellung werden alle Bestelldaten inklusive der Bestellsumme über alle Bestellpositionen angezeigt (read-only). Weiters können beliebig viele Bestellpositionen hinzugefügt und gelöscht werden!

Vor dem endgültigen Löschen einer Bestellung soll noch eine Sicherheitsrückfrage angezeigt werden:

Nur angemeldete Benutzer dürfen auf diese Seiten wechseln!

Bestellung löschen

Diese Bestellung wirklich löschen?

Bestellnummer 2020/1
Datum 01.01.2020
Kunde Herbert Berger

[Löschen](#) | [Zurück](#)

Kundenübersicht

Die Kundenübersicht soll ausschließlich für Benutzer der „Admin“-Rolle zugänglich sein.

Figaro.WebApp

HomeCustomersOrdersAPI

Logout (admin@htl.at)

Übersicht

Nachname

Nachname	Vorname	Username	IsAdmin	RegisteredSince	
Berger	Herbert	herbert.berger@htl.at	<input type="checkbox"/>	02.04.2021 11:17:02	Bearbeiten Details
Bregar	Martin	martin.bregar@htl.at	<input type="checkbox"/>	02.04.2021 11:17:02	Bearbeiten Details
Müller	Tanja	tanja.mueller@htl.at	<input type="checkbox"/>	02.04.2021 11:17:02	Bearbeiten Details
Sammer	Robert	robert.sammer@htl.at	<input type="checkbox"/>	02.04.2021 11:17:02	Bearbeiten Details
User	Admin	admin@htl.at	<input checked="" type="checkbox"/>	02.04.2021 11:19:01	Bearbeiten Details

© 2020/21 - Figaro.WebApp

In der Spalte „IsAdmin“ soll dargestellt sein, ob ein bestimmter Kunde sich in der Admin-Rolle befindet.

Kunde bearbeiten

Administratoren dürfen einzelne Kunden nach klick auf „Bearbeiten“ in der Kundenansicht auch verändern:

- Vorname
- Nachname
- Is Admin

Validierungen Sie folgende Bedingungen:

Edit

Customer

Id

8

UserName

dani@htl.at

Vorname

Das Feld 'Vorname' ist verpflichtend!

Nachname

Das Feld 'Nachname' ist verpflichtend!

IsAdmin

☒

Speichern

Zurück

Edit

Customer

Id

8

UserName

dani@htl.at

Vorname

x

Der Vorname muss mind. aus 2 Zeichen bestehen!

Nachname

x

Der Nachname muss mind. aus 2 Zeichen bestehen!

IsAdmin

☒

Speichern

Zurück

© 2020/21 - Bakery.WebApp

Bei ausgewählter Checkbox „IsAdmin“ muss der Kunde in die Admin-Rolle aufgenommen werden.

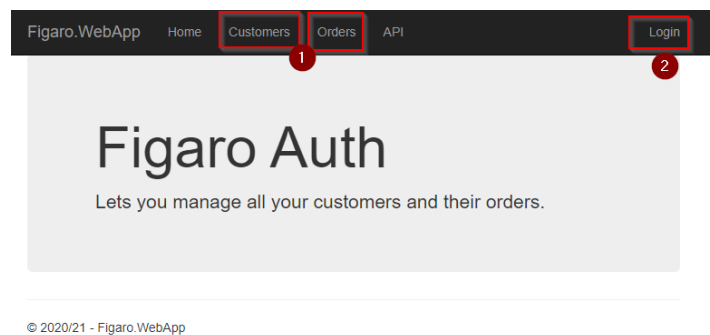
Initialisierung

Prüfen Sie beim Start der Web Anwendung ob es bereits einen Benutzer admin@htl.at gibt. Wenn nicht so soll dieser angelegt werden sowie gleich die zu erstellende Rolle „Admin“ zugewiesen werden!

Tipp: Sehen Sie sich dazu das TODO-Kommentar in Program.cs an.

Login

Folgende Bereiche der Web Applikation führen (wenn noch nicht angemeldet) zum Login-Dialog:



Beim Login werden folgende Validierungen durchgeführt:

<p>Login</p> <p>Username</p> <input type="text"/> <p>The Username field is required.</p> <p>Password</p> <input type="password"/> <p>The Password field is required.</p> <p>Login Register</p>	<p>Login</p> <p>• Login failed!</p> <p>Username</p> <input type="text" value="xxx@htl.at"/> <p>Password</p> <input type="password"/> <p>Login Register</p>
--	--

Über den Button „Register“ können sich neue Benutzer registrieren.

Register

Neue Benutzer können sich für die Anmeldung im System registrieren. Folgende Validierungen werden dabei durchgeführt:

Per Klick auf den Button „Register“ soll ein „Lokaler Redirect“ auf die Login-Seite durchgeführt werden!

Register

Firstname

The Firstname field is required.

Lastname

The Lastname field is required.

Username

The Username field is required.

Password

The Password field is required.

Register

Register

- Passwords must be at least 4 characters.
- Passwords must have at least one non alphanumeric character.
- Passwords must have at least one digit (0-9).
- Passwords must have at least one uppercase (A-Z).

Firstname

Lastname

Username

Password

Register

ASP.NET Web API

OrdersController

Es soll zusätzlich die Möglichkeit geschaffen werden, folgende Daten gesichert per JWT-Token über einen Rest-Services abfragen zu können

- **alle Bestellungen (/api/orders)**
 - o Benutzer mit der Rolle „Admin“ sehen alle Bestellungen in der Datenbank
 - o Orientieren Sie sich bei der Implementierung auf die zurückzugebenden StatusCodes.
 - o Standard-Benutzer sehen nur ihre eigenen Bestellungen!
- **alle Bestellungen zu einem bestimmten Kunden (/api/orders/ordersByCustomerId/5)**
 - o Kundeld wird übergeben
 - o Orientieren Sie sich bei der Implementierung auf die zurückzugebenden StatusCodes.
 - o Nur Benutzer mit der Rolle „Admin“ können diese Action aufrufen

```
Request URL
https://localhost:44347/api/Orders

Server response
Code    Details
200

Response body
{
  "orderedItems": [
    {
      "id": 15,
      "productName": "Air Haartrockner",
      "productPrice": "105,90",
      "amount": 1
    },
    {
      "id": 16,
      "productName": "Comair Nackenwedel Jumbo",
      "productPrice": "6,90",
      "amount": 4
    },
    {
      "id": 17,
      "productName": "Jaguar Kammset",
      "productPrice": "30,30",
      "amount": 2
    }
  ],
  "sales": 194.1,
  "orderId": 9,
  "orderNr": "2021/10",
  "date": "2021-04-02T00:00:00",
  "customerName": "Josef Furlinger"
}
```

```
Request URL
https://localhost:44347/api/Orders/ordersByCustomerId/4

Server response
Code    Details
200

Response body
[
  {
    "orderedItems": [
      {
        "id": 13,
        "productName": "Comair Pneumatikbürste mit Wildschweinborsten/Stifte",
        "productPrice": "6,55",
        "amount": 8
      }
    ],
    "sales": 52.4,
    "orderId": 7,
    "orderNr": "2020/5",
    "date": "2020-01-04T00:00:00",
    "customerName": "Martin Bregar"
  },
  {
    "orderedItems": [
      {
        "id": 14,
        "productName": "Comair Pneumatikbürste mit Wildschweinborsten/Stifte",
        "productPrice": "6,55",
        "amount": 9
      }
    ],
    "sales": 58.949999999999996,
    "orderId": 8,
    "orderNr": "2020/8",
  }
]
```


Figaro (Auth)

Curl

```
curl -X POST "https://localhost:44344/api/Auth/register" -H "accept: */*" -H "Content-Type: application/json" -d '{"username":"ludwig.hirsch@htl.at","password":"x","firstname":"Ludwig","lastname":"Hirsch"}'
```

Request URL

https://localhost:44344/api/Auth/register

Server response

Code	Details
400	Error: Response body <pre>{ "status": "Error", "message": "Passwords must be at least 4 characters., Passwords must have at least one non alphanumeric character., Passwords must have at least one digit ('0'-'9')., Passwords must have at least one uppercase ('A'-'Z')." }</pre> Response headers <pre>content-type: application/json; charset=utf-8 date: Sun14 Mar 2021 08:01:11 GMT server: Microsoft-IIS/10.0 x-powered-by: ASP.NET</pre>

Responses

Code	Description	Links
200	Success	No links

Hinweis:

Dekorieren Sie die Action-Methode mit den entsprechenden Response-Type Attributen (200, 400)!