

Bakery-Shop

Allgemein

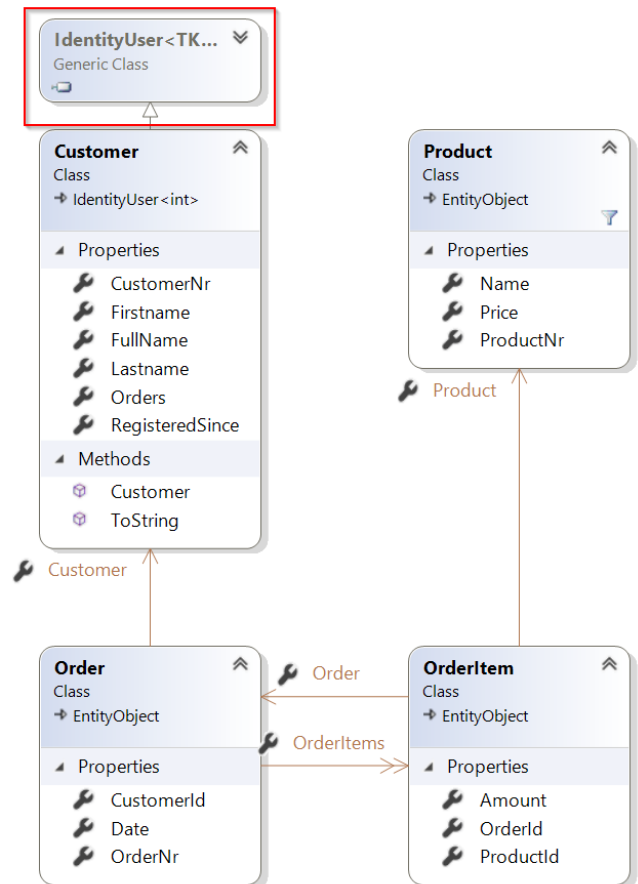
Es soll die bestehende Bäckerei-Software, welche die Kundenbestellungen verwaltet, per Microsoft Identity abgesichert werden, sodass sich Kunden in der Web Applikation anmelden können.

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: „Bakery.Persistence“
3. Startup Project: „Bakery.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 der Bäckerei-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!

Beenden mit Eingabetaste ...
  
```

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

Bakery.Web Home

Bestellübersicht

Nachname Suchen

Neue Bestellung anlegen

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

© 2020 - Bakery.Web

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 Suchen

Neue Bestellung anlegen

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!

Bakery.WebApp Home Customers Orders API

Neue Bestellung anlegen

Bestellung

Bestellnummer

Bestellnummer darf nicht leer sein!

Datum

20.03.2021

Kunde

Admin User

[Speichern](#)

[Zurück](#)

© 2020/21 - Bakery.WebApp

Bakery.WebApp Home Customers Orders API

Neue Bestellung anlegen

Bestellung

Bestellnummer

2020/1

Bestellung mit Nummer 2020/1 existiert bereits.

Datum

20.03.2021

Kunde

Admin User

[Speichern](#)

[Zurück](#)

© 2020/21 - Bakery.WebApp

Bearbeiten/Löschen einer Bestellung

Bestellung bearbeiten

Bestellnummer 2020/1
Datum 01.01.2020
Kunde Herbert Berger
Bestellsumme 15,40

Bestellpositionen

Produkt	Preis	Bestellmenge	
Cola	1,00	3	Entfernen
Fanta	0,90	5	Entfernen
Almdudler	0,79	10	Entfernen

Bestellposition hinzufügen

Almdudler [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.

Bakery.WebApp Home Customers Orders API Logout (admin@htl.at)

Übersicht

Nachname

Nachname	Vorname	Username	IsAdmin	RegisteredSince	
Berger	Herbert	herbert.berger@htl.at	<input type="checkbox"/>	19.03.2021 10:25:44	Bearbeiten Details
Bregar	Martin	martin.bregar@htl.at	<input checked="" type="checkbox"/>	19.03.2021 10:25:44	Bearbeiten Details
Fürlinger	Daniela	dani@htl.at	<input type="checkbox"/>	20.03.2021 05:52:19	Bearbeiten Details
Fürlinger	Josef	joe@htl.at	<input type="checkbox"/>	19.03.2021 10:54:23	Bearbeiten Details
Müller	Tanja	tanja.mueller@htl.at	<input type="checkbox"/>	19.03.2021 10:25:44	Bearbeiten Details
Sammer	Robert	robert.sammer@htl.at	<input type="checkbox"/>	19.03.2021 10:25:44	Bearbeiten Details
User	Admin	admin@htl.at	<input checked="" type="checkbox"/>	19.03.2021 10:28:09	Bearbeiten Details

© 2020/21 - Bakery.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:

Bakery.WebApp Home Customers Orders API

Edit

Customer

Id

UserName

Vorname

Das Feld 'Vorname' ist verpflichtend!

Nachname

Das Feld 'Nachname' ist verpflichtend!

IsAdmin ☒

[Zurück](#)

© 2020/21 - Bakery.WebApp

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

Bakery.WebApp Home Customers Orders API

Edit

Customer

Id

UserName

Vorname

Der Vorname muss mind. aus 2 Zeichen bestehen!

Nachname

Der Nachname muss mind. aus 2 Zeichen bestehen!

IsAdmin ☒

[Zurück](#)

© 2020/21 - Bakery.WebApp

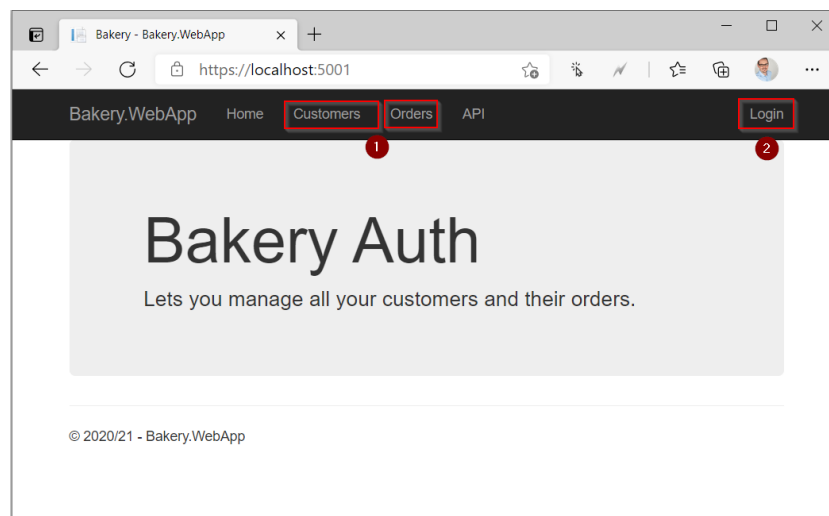
Initialisierung

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!

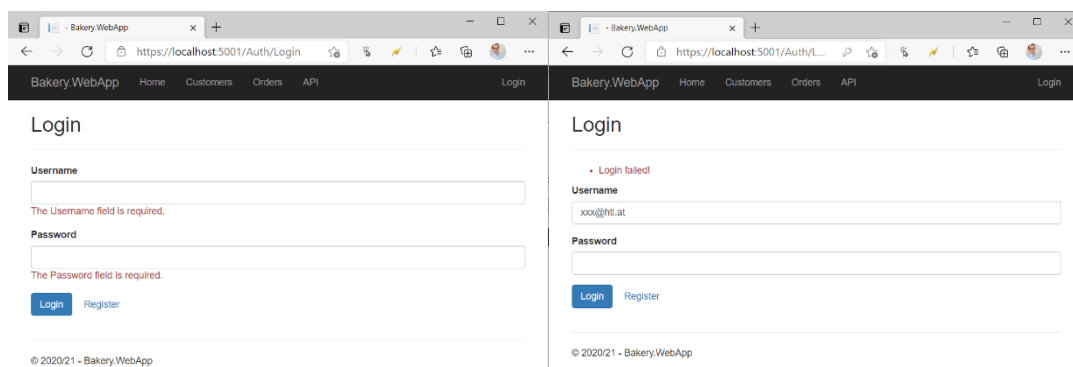
Tipp: Sehen Sie sich dazu das TODO-Kommentar in Program.cs an.

Login

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



Beim Login werden folgende Validierungen durchgeführt:



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:

The screenshot shows the Register page of Bakery.WebApp. The page has a dark navigation bar with links: Bakery.WebApp, Home, Customers, Orders, API, and Login. The main content area is titled "Register" and contains four input fields: Firstname, Lastname, Username, and Password. Each field has a red error message below it: "The Firstname field is required.", "The Lastname field is required.", "The Username field is required.", and "The Password field is required.". A blue "Register" button is at the bottom left. The footer shows "© 2020/21 - Bakery.WebApp".

The screenshot shows the Register page of Bakery.WebApp with the input fields filled: Firstname (Susanne), Lastname (Mustermann), Username (sumu@nt.at), and Password (empty). Above the fields, there are four bullet points for password requirements: "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').", and "Passwords must have at least one uppercase ('A'-'Z').". A blue "Register" button is at the bottom left. The footer shows "© 2020/21 - Bakery.WebApp".

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

REST-Abfragen

Es soll zusätzlich die Möglichkeit geschaffen werden, folgende Daten per Rest-Services abfragen zu können:

- **alle Bestellungen (/api/orders)**
 - o Benutzer mit der Rolle „Admin“ sehen alle Bestellungen in der Datenbank
 - o Standard-Benutzer sehen nur ihre eigenen Bestellungen!
- **alle Bestellungen zu einem bestimmten Kunden (/api/orders/ordersByCustomerId/5)**
 - o Kundeld wird übergeben
 - o Nur Benutzer mit der Rolle „Admin“ können diese Action aufrufen

Request URL	
http://localhost:64535/api/Orders	
Server response	
Code	Details
200	<p>Response body</p> <pre>[{ "orderedItems": [{ "id": 1, "productName": "Cola", "productPrice": "1,00", "amount": 3 }, { "id": 2, "productName": "Fanta", "productPrice": "0,90", "amount": 5 }, { "id": 3, "productName": "Almdudler", "productPrice": "0,79", "amount": 10 }], "sales": 15.4, "orderId": 1, "orderNr": "2020/1", "date": "2020-01-01T00:00:00", "customerName": "Herbert Berger" }, { "orderedItems": [{ "id": 13, "productName": "Kaffee", "productPrice": "3,00", "amount": 8 }], "sales": 24, "orderId": 7, "orderNr": "2020/5", "date": "2020-01-04T00:00:00", "customerName": "Martin Bregar" }, { "orderedItems": [{ "id": 14, "productName": "Kaffee", "productPrice": "3,00", "amount": 9 }], "sales": 27, "orderId": 8, "orderNr": "2020/8", "date": "2020-01-09T00:00:00", "customerName": "Martin Bregar" }]</pre>

Request URL	
http://localhost:64535/api/Orders/ordersByCustomerId/4	
Server response	
Code	Details
200	<p>Response body</p> <pre>[{ "orderedItems": [{ "id": 13, "productName": "Kaffee", "productPrice": "3,00", "amount": 8 }], "sales": 24, "orderId": 7, "orderNr": "2020/5", "date": "2020-01-04T00:00:00", "customerName": "Martin Bregar" }, { "orderedItems": [{ "id": 14, "productName": "Kaffee", "productPrice": "3,00", "amount": 9 }], "sales": 27, "orderId": 8, "orderNr": "2020/8", "date": "2020-01-09T00:00:00", "customerName": "Martin Bregar" }]</pre>