

Use Case

Das System soll es ermöglichen mit dem Framework HTMX Websites aufzubauen, ohne ein eigenes Backend entwickeln zu müssen. Das System bietet ein fertiges NoSQL basiertes Backend mit eingebauter Nutzer- und Berechtigungsverwaltung. Zudem ist es möglich neue Collections über die API anzulegen und diese zu manipulieren. Ebenfalls soll es möglich sein statischer Content hochzuladen und HTML Templates im Handlebar Syntax zu hinterlegen. Das Backend kann dann bei einer API Request welche Daten aus einer collection liest nicht nur in JSON antworten sondern auch hinterlegte HTML Templates befüllen. Abschliessend soll es möglich sein eigene URL's zu definieren aus welchen Pfadteile und Queryparameter genutzt werden um aus verschiedenen collections daten zu filtern und diese an ein HTML Template zu binden. So soll es einfach möglich sein eine Art CMS aufzubauen.

Funktionale Requirements

ID	Beschreibung
CL-01	Über die API müssen mit einem JSON Schema neue Collections in der Datenbank angelegt werden können.
CL-02	Über die API müssen Daten über JSON POST in erstellte Collections eingefügt werden können.
CL-03	Über die API müssen Daten über HTML Form POST in erstellte Collections eingefügt werden können.
CL-04	Die API muss beim Einfügen der Daten das JSON-Schema validieren und mit einem HTTP 400 Fehler antworten, wenn die Validation fehlschlägt.
CL-05	Über die API müssen Daten in erstellten Collections anhand eines Schlüssels aus erstellten Collections gelöscht werden können
CL-06	Über die API müssen Daten in erstellten Collections eine MongoDB Query gesucht werden können.
CL-07	Über die API müssen Daten über HTML Form POST in erstellten Collections durchsucht werden können.
CL-08	Über die API müssen erstellte Collections gelöscht werden können.
CP-01	Über die API müssen Berechtigungen für Durchsuchen mit MongoDB Query von erstellten Collections vergeben werden können
CP-02	Über die API müssen Berechtigungen für Einfügen von Daten in erstellte Collections vergeben werden können
CP-03	Über die API müssen Berechtigungen für das Verändern von Daten in erstellten Collections vergeben werden können
CP-04	Über die API müssen Berechtigungen für das Löschen von Daten in erstellten Collections vergeben werden können
CT-01	Über die API müssen Templates für die erstellten Collections angelegt werden können welche ein Template im Handlebar Syntax enthalten
CT-02	Ein Collection Template muss über seine ID als Slug angewendet werden können (GET /collectionId/templateId)
CT-03	Über die API müssen Templates gelöscht werden können
CT-04	Über die API muss ein Standard Template für eine erstellte Collection festgelegt werden können
CT-05	Einem Template müssen immer der aktuell eingeloggte Nutzer sowie die gefilterten Daten aus der Collection zur Verfügung gestellt werden
AT-01	Über die API müssen sich Benutzer anmelden können und ein JWT erhalten welches ihre Berechtigungen widerspiegelt

AT-02	Bei erfolgreichen Login muss über ein query-parameter einstellbar sein ob das JWT per response Body oder Cookie zurückgegeben wird
AT-03	Über die API muss es mit administrativer Berechtigung möglich sein Nutzerkonten zu deaktivieren
AT-04	Über die API muss es möglich sein sein Passwort zu ändern
AT-05	Über die API muss es möglich sein sich zu registrieren
AR-01	Über die API muss es möglich sein neue Gruppen anzulegen
AR-02	Über die API muss es möglich sein Gruppen zu löschen
AR-03	Über die API muss es möglich sein Gruppen Berechtigungen hinzuzufügen
AR-04	Über die API muss es möglich sein Gruppen Berechtigungen zu entziehen
AR-05	Über die API muss es möglich sein Nutzern Gruppen hinzuzufügen
AR-06	Über die API muss es möglich sein Nutzer aus Gruppen zu entfernen
AR-07	Über die API muss es möglich sein Nutzern Berechtigungen hinzuzufügen
AR-08	Über die API muss es möglich sein Berechtigungen von Nutzern zu entfernen
AR-09	Über die API muss es möglich sein Nutzer zu Administratoren zu machen
RT-01	Über die API muss es möglich sein neue URL Routen zu definieren
RT-02	Für Routen muss es möglich sein eine Collection festzulegen zu welcher diese auflösen
RT-03	Für Routen muss es möglich sein Collections dynamisch über einen Teil der URL aufzulösen
RT-04	Für Routen muss es möglich sein ein festes HTML Template zu definieren
RT-05	Für Routen muss es möglich sein ein Collection Templates dynamisch über einen Teil der URL aufzulösen
RT-06	Für Routen müssen Queryparameter definiert werden können durch mit welcher die aufgelöste Collection nach einzelnen Feldern durchsucht werden kann.
RT-07	Für Routen muss es möglich sein ein MongoDB Query Template in Handlebars Syntax zu definieren was anstatt von Feldern benutzt wird und über QueryParameter befüllt wird
SC-01	Über die API muss es möglich sein statischen Content als Blob hochzuladen
SC-02	Über die API muss es möglich sein statischen Content zu löschen
SC-03	Über die API muss es möglich sein statischen Content abzurufen
SC-04	Über die API muss es möglich sein Berechtigungen zum Lesen von statischen Content festzulegen
SC-05	Über die API muss es möglich sein Berechtigungen zum Löschen von statischen Content festzulegen

Datenmodelle

Legende

Titel: Friendlyname (Database Collection)

* = Unique Field

? = Nullable Type

EntityBase (alle Entitäten mit eigener Collection erben hiervon)

Feld	Typ	Beschreibung
_id*	ObjectId	Eindeutige Id des objektes

CreatedAt	TimeStamp	Wann das Objekt erstellt wurde
UpdatedAt	TimeStamp	Wann das Objekt zuletzt modifiziert wurde

User (users)

Feld	Typ	Beschreibung
PasswordHash	String	Hash des passworts vom nutzer
Username*	String	Nutzername
Email*	String	Email des Benutzers
IsLockoutEnabled	Bool	Ob der Account gesperrt ist
FirstName	String?	Vorname des Nutzers
LastName	String?	Nachname des Nutzers
AvatarUrl	String?	Link zum Avatar Bild des Benutzers
Groups	String[]	Liste an Gruppen zu denen der Nutzer gehört
Permissions	String[]	Liste an Berechtigungen die der Nutzer hat

Group (groups)

Feld	Typ	Beschreibung
Slug*	String	Eindeutiger url friendly name der Gruppe
Name	String	Anzeigename der Gruppe
Description	String?	Beschreibung der Gruppe
Permissions	String[]	Berechtigungen der Gruppe

Collection (collections)

Feld	Typ	Beschreibung
Slug*	String	Eindeutiger url friendly name der Collection
Name	String	Anzeigename der Collection
CacheRetentionTime	TimeSpan?	Ob und wie lange ergebnisse aus der Collection gecached werden dürfen
Schema	Json	Das Schema der Collection
Templates	Template[]	Liste an Templates der Collection
DefaultTemplate	String?	Slug des Tempaltes das standardmässig verwendet werden soll
QueryPermission	String?	Ob Query von Daten erlaubt sein soll und wie die Berechtigung heisst
InsertPermission	String?	Ob Einfügen von Daten erlaubt sein soll und wie die Berechtigung heisst
ModifyPermission	String?	Ob Modifizieren von Daten erlaubt sein soll und wie die Berechtigung heisst
DeletePermission	String?	Ob Löschen von Daten erlaubt sein soll und wie die Berechtigung heisst
ComplexQueryPermission	String?	Ob MongoDB Queries von Daten erlaubt sein sollen und wie die Berechtigung heisst

Template (collections)

Feld	Typ	Beschreibung
SingleItem	Bool	Ob das Template für ein oder mehrere Items aus der Collection gedacht ist
Slug*	String	Eindeutiger url friendly name des Templates
Template	String	Das Handlebars HTML Template

RouteTemplate (route-templates)

Feld	Typ	Beschreibung
UrlTemplate	String	Template der Url welche gematcht werden soll
CollectionId	ObjectId?	Id der festgebundenen Collection wenn nicht über Route gematcht
TemplateId	String?	Slug des Templates der gematchten Collection wenn nicht über Route gematcht
Fields	FieldMatch[]?	Liste an Mapping zwischen Query Values und Feldern welche verglichen werden sollen
QueryTemplate	String?	Alternative MongoDB Query Template in Handlebar Syntax für komplexe Abfragen

FieldMatch (route-templates)

Feld	Typ	Beschreibung
Name	String	Name des Feldes in der Datenbank
MatchKind	MatchKind	Wie das Feld mit dem Query Value verglichen werden soll
Type	BsonType	In welchen Bson Typen das Query Value vor dem Vergleich geparsed werden muss

MatchKind (enum)

Id	Name	Beschreibung
0	Equals	Die Felder haben den gleichen Wert
1	NotEquals	Die Felder haben nicht den gleichen Wert
2	Contains	Das Feld ist ein Array und beinhaltet den Wert
3	NotContains	Das Feld ist ein Array und beinhaltet den Wert nicht
4	LargerThan	Das Feld ist numerisch oder ein Datum und grösser als der Wert
5	LessThan	Das Feld ist numerisch oder ein Datum und kleiner als der Wert

StaticContent (static-content)

Feld	Typ	Beschreibung
Name	String	Name der Datei
MimeType	String	MimeType der Datei
Length	Int	Länge der Datei in Bytes
Slug*	String	Url freundlicher Name der Datei, muss mit VirtualPath zusammen eindeutig sein
VirtualPath	String?	Virtueller Verzeichnispfad der Datei ('/' bei null value)

Path	String*	Pfad zur Datei auf dem physischen Speichermedium
DeletePermission	String?	Name der Berechtigung zum Löschen der Datei
ReadPermission	String?	Name der Berechtigung zum Lesen der Datei

Techstack

Zum Speichern der Datenbank kommt MongoDB zum Einsatz, da diese einen flexiblen Umgang mit dem Datenschema erlaubt was das custom collection erstellen einfach macht. Für das Backend kommt asp.net mit C# zum Einsatz, da das Framework hohe Sicherheit, Stabilität, Skalierbarkeit und Performanz bietet. Zu dem gibt es für C# ein first party SDK zur Kommunikation mit MongoDB. Für das Deployment wird ein Dockerfile erstellt und für die Quellverwaltung kommt GitHub zum Einsatz.