**Mechanismen für die Authentifizierung und Autorisierung umsetzen**

Im program.js die JWT Konfiguration hinzufügen:  
Ein Bild, das Text, Screenshot, Software enthält.

Automatisch generierte Beschreibung

Im LoginController die Methode CreateToken erstellen, um ein JWT-Token für die authentifizierten Benutzer zu erstellen. Der Token enthält verschiedene Claims wie die Benutzer-ID(NameId), Benutzernamen(UniqueName) und die Benutzerrolle(Role) ist der Benutzer ein Admin so bekommt er die admin Rolle wenn nicht kriegt er die Benutzerrolle:

Ein Bild, das Text, Screenshot, Software enthält.

Automatisch generierte Beschreibung

In der allgemeinen Login Methode kann man nun die CreateToken Methode verwenden um für jeden autorisierten Nutzer einen individuellen JWT-Token zu erstellen:

Ein Bild, das Text, Screenshot, Software enthält.

Automatisch generierte Beschreibung



Im Frontend der Applikation kann man nun den JWT verwenden.

In dem Login File kann man folgende Funktionen erstellen: saveJWT, getJwtToken, getjwtHeader und parseJwt. Diese Funktionen dienen dazu den JWT-Token lokal zu speichern, den JWT-Token aus dem lokalen Speicher holen, den Authentifizeriungsheader für den JWT-Token zu erstellen(welchen man später in den http-Anfragen verwenden kann), den Inhalt des JWT-Tokens zu analysieren und dekodieren.



Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Nun kann man diese Funktionen verwenden um in der http-Anfrage an das Backend den JWT-Token mitzuschicken wenn einer verfügbar ist und der Server kann so dem autorisierten Benutzer Zugriff zur Funktion gewähren oder unautorisierten Benutzern den Zugriff verweigern. Hier ist ein Beispiel für die Funktion von handleSaveNew:

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

