

Komplexaufgabe Angular

Die Aufgabe soll im Team erledigt werden. Richten Sie sich hierzu ein entsprechendes Git-Repository ein, auf dem Sie gemeinsam entwickeln können.

Es soll eine Anwendung entwickelt werden, welche Zeitbuchungsdaten erfasst, speichert und auswertet. Zunächst soll das Backend in Form einer Asp.net-Anwendung bereitgestellt werden.

Aufgabe 1:

Entwickeln Sie ein Datenmodell mit EF Core mit folgenden Entitäten:

- Benutzer (Vor- und Zuname, Personalnummer, E-Mailadresse, ...)
- Vertrag (erforderliche Arbeitszeit pro Wochentag, Pausenregelung)
- Zeitbuchung (Datum, Uhrzeit, Art der Buchung d.h. Kommen, Gehen, ...)

Aufgabe 2:

Rollen Sie das Datenbankmodell auf eine Datenbank Ihrer Wahl aus und entwickeln Sie die entsprechenden API-Funktionen (Hinzufügen, Löschen, Ändern, Lesen), sowie die DTO-Klassen für die Entitäten! Nutzen Sie dabei einen Auto-Mapper!

Aufgabe 3:

Binden Sie Swagger, sowie die Windows-Authentifizierung ein und testen Sie Ihre Anwendung! Es soll auch ein passender Logging-Mechanismus implementiert werden. Lassen Sie sich die Client-API (inkl. der DTOs!) über die swagger.json generieren!

Nun soll das Frontend in Angular entwickelt werden.

Aufgabe 4:

Erstellen Sie eine neue Angular-Anwendung und binden die generierte Client-API ein!

Aufgabe 5:

Entwickeln Sie eine ansprechende Willkommensseite, die eine Registrierung anbietet! Ist der Benutzer bereits registriert, so erfolgt eine automatische Authentifizierung über das Windows-Benutzerkonto und es sollen die Benutzerinformationen übersichtlich angezeigt werden. Entwickeln Sie auch eine Komponente für die Registrierung inkl. sinnvoller Validierung!

Bei erfolgreicher Authentifizierung sollen folgende Optionen verfügbar sein:

- Zeitenanalyse
- Zeiten anzeigen und bearbeiten
- Zeiten erfassen

Aufgabe 6:

Entwickeln Sie zunächst die Komponente „Zeiten erfassen“. Hier soll der Benutzer die Möglichkeit bekommen, die Zeiten für einen ausgewählten Arbeitstag zu erfassen und zu speichern. Sollten bereits Buchungen an diesem Tag vorhanden sein, sollten diese nach Rückfrage überschrieben werden. Es gibt die Möglichkeit, die Zeiten aus einem Textformat zu laden, oder von einer geöffneten Website abzurufen. Die eingelesenen Zeiten sollen auf Plausibilität überprüft werden. Entwickeln Sie hierzu auch einen passenden Service, der auch in den folgenden Aufgaben verwendet werden kann!

Aufgabe 7:

Entwickeln Sie nun die Komponente „Zeiten anzeigen und bearbeiten“. Der Benutzer wählt hierzu ein Datum der Zeiterfassung aus und die Daten werden übersichtlich in einer Tabelle dargestellt. Der Benutzer kann die Zeitbuchungen auch bearbeiten. Führen Sie jedoch auch hier eine Plausibilitätsprüfung durch!

Aufgabe 8:

Entwickeln Sie zuletzt die Komponente „Zeitanalyse“, welche die Bekannten Statistiken erzeugt und die Daten auch grafisch darstellt z.B. Kreisdiagramm, Fortschrittsbalken, ... Seien Sie kreativ!

Aufgabe 9:

Entwickeln Sie für alle Services und Komponenten entsprechende Tests!