

Coding Challenge für Backend Developer

Um einen Eindruck von deiner Arbeitsweise zu bekommen, führen wir als OBI eine Coding Challenge im Rahmen des Bewerbungsprozesses durch.

Wähle aus den folgenden Aufgaben Eine aus, die du innerhalb von 7 Tagen nach Erhalt bearbeitest. Stelle uns deine Lösung in einer für dich geeigneten Form zur Verfügung. Sie sollte für uns ausführbar sein. Anschließend werden mit dir einen Folgetermin vereinbaren, um über die Lösung in einem Fachgespräch gemeinsam zu sprechen.

(Alternativ: Du kannst auch auf die Coding Challenge verzichten und im zweiten Gespräch eine (Mob Programming) Live Coding Session teilen.)

Aufgaben

Wähle eine der folgenden Aufgaben und löse sie im Backend. Für die Aufgabe können beliebige Technologien verwendet werden. Programmiersprache würden wir allerdings gern auf Java oder Node.JS einschränken.

Warenkorb

Es soll ein rudimentärer Warenkorb implementiert werden. Es reicht aus, wenn die Funktionalitäten über eine API angesprochen werden.

Folgende User Stories bilden den MVP:

- Als Kunde möchte ich einen Artikel in einer freien Anzahl(Ganzzahl) in den Warenkorb legen, um diesen später bestellen zu können.
 - Ist Anzahl > Bestand des Artikels, wird der Artikel nicht in den Warenkorb gelegt
 - Ist Anzahl < Mindestanzahl des Artikels, wird der Artikel in der Mindestanzahl in den Warenkorb gelegt
- Wird ein Artikel in den Warenkorb gelegt, der sich bereits im Warenkorb befindet, wird die Anzahl des Artikels im Warenkorb um die hinzugefügte Anzahl, wenn möglich, erhöht.
- Als Kunde möchte ich meinen Warenkorb anhand der Warenkorb-ID anzeigen(ausgeben), um den Inhalt überprüfen zu können.
 - Die Ausgabe enthält alle Positionen(Artikel, Anzahl, Preis)
 - Die Ausgabe enthält einen Gesamtpreis. Dieser wird aus der Summe der Positionspreise berechnet. Ein Positionspreis ist das Produkt aus dem Artikelpreis und der Anzahl

Optional:

- Als Kunde möchte ich einen Artikel aus dem Warenkorb löschen
- Als Kunde möchte ich den kompletten Warenkorb leeren können

Zombieangriff

Es findet ein Zombiangriff auf eine Unterkunft der Überlebenden statt und wird von Überlebenden

abgewehrt. Entwickle ein Programm, das den folgenden Zombiangriff simuliert.

- Als Simulierender möchte ich folgende Parameter einstellen können
 1. Initiale Zombieanzahl
 2. Initiale Überlebendenanzahl
 3. Wahrscheinlichkeit X, dass ein Überlebender einen Zombie tötet (1-99)
 4. Wahrscheinlichkeit Y, dass ein Zombie einen Überlebenden verwandelt (1-99)
- Die Simulation dauert so lange bis nur noch Überlebende oder nur noch Zombies übrig bleiben.
 - Auf der Konsole wird die Gesamtdauer der Simulation und die Gewinnerfraktion mit der Anzahl der Übriggebliebenen ausgegeben
- Als Zombie stelle ich mich vor die Unterkunft und versuche von einem Überlebenden wahrgenommen zu werden
 - Ein Zombie erzeugt ein "Grrrrr" auf der Konsole, wenn er sich anstellt
- Als Überlebender versuche ich den Zombie zu töten. Dies gelingt mir mit einer Wahrscheinlichkeit von X % (Parameter 3)
 - Ein getöteter Zombie erzeugt ein "Klatsch" auf der Konsole
 - Ein Überlebender, der nicht trifft, erzeugt "Mist!" auf der Konsole
- Als Überlebender, der den Zombie nicht getroffen hat, werde ich vom Zombie angegriffen und mit einer Wahrscheinlichkeit von Y % (Parameter 4) in einen Zombie verwandelt.
 - Ein verwandelter Zombie stellt sich, wie Andere, vor die Unterkunft und wartet auf Überlebende
 - Ein Überlebender, der dem Angriff entkommt, erzeugt "Juhu" auf der Konsole und versucht den nächsten Zombie zu töten.

Rahmenbedingungen:

- Es sollen mehrere Überlebende und Zombies gleichzeitig agieren können.
- Kein Zombie darf von mehreren Überlebenden angegriffen werden

Optional:

- Ein Überlebender, der einem Zombie entkommt bekommt einen Bonus von 3% auf die Trefferwahrscheinlichkeit
- Es gibt unterschiedlich starke Zombies
- Es gibt unterschiedlich starke Waffen und Rüstungen bei den Überlebenden

User-Verwaltung

Aufgabe ist es, eine Webanwendung (Backend + UI) für eine User-Verwaltung zu implementieren. Folgende User Stories repräsentieren die Anforderungen an das Produkt:

- Als Kunde möchte ich, dass User-Datensätze mindestens aus Name, Adresse, E-Mail Adresse und Geburtsdatum bestehen, damit alle Geschäftsprozesse damit abbildbar sind.
- Als Kunde möchte ich, dass im UI alle existierenden User mit Name, Adresse, E-Mail Adresse und Geburtsdatum auf einer Seite einsehbar sind, damit ich einen Überblick über alle existierenden User habe.
- Als Kunde möchte ich, dass im UI über ein Formular neue User angelegt werden können, damit ein einfacher Weg zur Erstellung neuer User existiert.
- Als Kunde möchte ich, dass existierende User über das UI gelöscht werden können, um schnell auf DSGVO Löschanfragen reagieren zu können.