RYTLE GmbH
Konsul-Smidt-Straße 20
28217 Bremen

Coding Challenge
Fullstack Development

Stelle dir einen kleinen und einen großen Würfel vor. Ein kleiner Würfel kann jeweils in einem großen Würfel platziert werden. In einen großen Würfel passen insgesamt 4 kleine Würfel. Jeder Würfel verfügt über **eine einzigartige ID (String)**, einer **optionalen Besitzer ID (String oder null)** sowie **einem Typ (groß oder klein)**. Bitte implementiere folgende Problemstellung in TypeScript und verzichte auf die Verwendung von weiteren Packages (außer Jest und TypeScript).

Versuche alle Aufgaben mit den Werkzeugen zu implementieren, die Dir TypeScript zur Verfügung stellt. Betrachte die Aufgabe als Schnittstelle für andere Entwickler.

1. Schreibe eine asynchrone Funktion zur Erstellung eines Würfels (groß und klein). Als ersten Parameter übergibst Du jeweils den Typ des Würfels (groß oder klein). Der zweite Parameter beschreibt eine optionale Besitzer ID. Der letzte Parameter beschreibt ein optionales Array an einzigartigen Ids zur Erstellung von kleinen Würfeln (Rekursive Erstellung mit max. Tiefe 1).
    a. Die zu implementierende Funktion soll folgende Signatur besitzen:

    async createCube(id: string, ownerId?: string | null, childIds?: string[]) : Promise<Cube>

    b. Beachte, dass ein Würfel, der sich innerhalb eines anderen Würfels befindet, dessen Besitzer annimmt.
2. Schreibe eine asynchrone Funktion zum Löschen eines Würfels. Diese Funktion erhält als ersten Parameter eine Liste aller erstellten Würfel. Als zweiten Parameter die ID des Würfels, der gelöscht werden soll und einen dritten optionalen Parameter, der entscheidet, ob Würfel, die sich innerhalb des zu löschenden Würfels befinden, ebenfalls gelöscht werden. Dieser Parameter ist standartmäßig mit *false* angegeben. Sollen die Würfel, die sich innerhalb des betrachteten Würfels befinden, nicht gelöscht werden, dann erhalten sie einen leeren Besitzer. Wenn die angegebene Würfel ID nicht existiert, sollte ein Fehler auftreten.
    a. Die zu implementierende Funktion soll folgende Signatur besitzen:

    async deleteCube(cubes: Cube[], id: string, deleteChilds?: boolean) : Promise<Cube[]>
3. Überprüfe Deine Implementierung über Unit Tests mithilfe von Jest (https://www.npmjs.com/package/jest) . Dabei solltest du jedes mögliche Verhalten, dass du der Aufgabenstellung entnehmen kannst, testen.

Für diese Coding Challenge stellen wir Dir ein Template bereit.

Liebe Grüße und viel Erfolg wünscht dir das RYTLE Software Team