

DreamRobot Programmierer Probe-Aufgaben

#1 Erstellen eines MariaDB Container

Um die aktuelle Probeaufgabe zu erfüllen, muss im ersten Schritt ein Maria DB Docker Container erstellt und eine Tabelle products mit dem folgenden Schema angelegt werden

id	title	description	category	state
1	Samsung galaxy s10	Mobile Phone from ..	Phone	true
2	Apple iPhone 13	Apple mobile phone ..	Phone	true
3	Apple watch 8	Apple Watch 8	Smartwatch	true
4	Garmin Approach S62	Garmin golf watch	Smartwatch	true
5	Apple air pods	ANC headphones from apple	Headphone	true

Die oben aufgeführten Produkte sind in der Datenbank anzulegen und der Container sollte so konfiguriert werden, dass dieser dauerhaft detached läuft und bei einem Absturz automatisch neugestartet wird. Aufgabe ist es das Dockerfile, das Container Startscript und die Datenbank mit Tabelle und Inhalt einzurichten. Bitte nutze ein offizielles Docker Image, soweit möglich.

#2 PHP-Anwendung mit Rest-API

Schreibe eine PHP-Anwendung, welche eine REST-API anbietet mit einem Framework deiner Wahl. Folgende Funktionalität soll die API anbieten:

1. Abrufen aller Produkte aus der Tabelle products (Aufgabe 1).
2. Abrufen eines bestimmten Produkts aus der products-Tabelle.
3. Hinzufügen eines neuen Produkts in der Tabelle products.
4. Hinzufügen von mehreren Produkten zur Tabelle products.
5. Aktualisieren eines konkreten Produkts in der Tabelle products.
6. Löschen eines Produkts aus der Tabelle products anhand eines eindeutigen Schlüssels.

Input- und Output-Format ist hierbei JSON.

#3 PHP-Anwendung in Container packen und betreiben

Die erstellte Applikation soll in einem Docker Container verpackt werden und in diesem ausgeführt werden. Bitte beachte die Port-Belegung und nutze hier einen sinnvollen Port.

#4 Rate-Limit Implementieren

Implementiere in der bestehenden API ein Rate-Limiting-System, das eine festgelegte Anzahl an Anfragen pro Minute erlaubt (z.B. 100 Anfragen pro IP pro Minute).

Anforderung I: Das System darf nicht auf einem bestehenden Framework basieren und sollte manuell in PHP umgesetzt werden.

Anforderungen II: Schreibe einen automatisierten Test, welcher das implementierte Rate-Limit ausreizt, validiert und visuell einen dritten anzeigt.

#5 Github

Bitte Lade deine Probearbeit in deinen Github-Account (kostenlose Variante ggfs. erstellen) und Teile uns den Link zum Repository mit, so dass wir deine Arbeit auschecken und starten können. Alle notwendigen Dateien, die für die Ausführung benötigt werden müssen im Repository eingecheckt werden und eine Readme.md beschreibt wie die Applikation zu starten und zu benutzen ist. Weiter sollten Beispiel Aufrufe der API zum einfachen Nachvollziehen dokumentiert sein und beschrieben werden wie der Test des Rate-Limits zu starten ist. Bitte denke daran dein Code sinnvoll und kurz zu dokumentieren, so dass deine Gedanken nachvollziehbar sind. Gerne fasse in der Readme.md zusammen, was du bei einer erneuten Entwicklung anders machen würdest. Uns ist bewusst das nicht alles Perfekt wird unter Zeitdruck es ist aber wichtig zu wissen, wo du Abstriche machen musstest.

Bitte nimm dir für die Aufgabe maximal 8 Stunden Zeit und Teile uns mit wann du startest und wann du fertig geworden bist mit der Aufgabe. Bitte dokumentiere deinen Code und schick uns die URL zum Code-Repository. Bei Rückfragen schreibe mich bitte an: therrmann@ecommerceone.de

Viel Spaß mit der Aufgabe.