Java 822

AutoRest: Spring Boot REST API

רקע

התקבלתם כמפתחי תוכנה לחברת AutoRest הנותנת שירותי השכרת רכב ללקוחות פרטיים.
שרתי AutoRest מאחסנים מידע אודות רכבים זמינים להשכרה ורכבים מושכרים, פעילים בשטח.

בכדי לשרת את לקוחותיה, סוכני החברה משתמשים בטאבלט עליו מותקנת אפליקצית צד-לקוח המתקשרת עם שרתי החברה לשם קבלת מידע אודות רכביה.
בהסתמך על המידע המתקבל, יכול הטאבלט להציג את מסכי האפליקציה ובכך להקהל על הסוכן בעת מתן שירות ללקוח.

AutoRest

המטרה

כתיבת שירות REST API אשר יספק לצד-לקוח את כל המידע הנדרש לשם תפקודו.

מבנה הפרוייקט

סביבת פיתוח: IntelliJ Ultimate

סוג הפרוייקט: Maven

כלי עזר ליצירת הפרוייקט: Spring Initializr

שפה וגרסא: Java 17

תלויות הפרוייקט:

- Web
- Data JPA
- Dev Tools
- Lombok
- MySQL or PostgreSQL Driver

AutoRest

ישויות - Entities

Vehicle

תיאור: רכב בודד משלל רכבי AutoRest.

יכיל את השדות הבאים:

- מזהה ייחודי בדטא-בייס.
- מזהה ייחודי אוניברסלי - כמו UUID, למשל.
- שם יצרן - למשל Audi.
- שם דגם - למשל A7.
- האם הרכב זמין להשכרה כעת.
- מחיר השכרת הרכב ליום.
- סך הק"מ שעשה הרכב.
- הלקוח, Client, המחזיק כעת ברכב - יוסבר בהמשך.
- זמן יצירת הרכב בדטא-בייס, Timestamp למשל.
- זמן עדכון הרכב בדטא-בייס, Timestamp למשל.
- מס' גרסא.

Client

תיאור: לקוח AutoRest

- מזהה ייחודי בדטא-בייס.
- מזהה ייחודי אוניברסלי - כמו UUID, למשל.
- שם הלקוח.
- הרכב, Vehicle, המוחזק בידי הלקוח.
- מועד החזרת הרכב, למשל LocalDateTime.
- זמן יצירת הלקוח בדטא-בייס, Timestamp למשל.
- זמן עדכון הלקוח בדטא-בייס, Timestamp למשל.
- מס' גרסא.

Data Transfer Objects

אפליקציית צד הלקוח של AutoRest תתקשר עם השרת באמצעות שליחת DTOs וקבלת DTOs. באופן שכזה השרת לא יחשוף פרטים אודות אופן אחסון הנתונים בדטא-בייס ולפיכך לא יידרוש מידע מיותר.

VehicleDto

תיאור: אובייקט המתפקד כנושא מידע המגדיר רכב.

- מזהה ייחודי אוניברסלי.
- שם יצרן.
- שם דגם.
- סך הק"מ שעשה הרכב.
- מחיר השכרת הרכב ליום.

ClientDto

תיאור: אובייקט המתפקד כנושא מידע המגדיר לקוח.

- מזהה ייחודי אוניברסלי.
- שם הלקוח.

נקודות קצה - Endpoints

ה URL הבסיסי יהיה */api/autorest/*

תיאור	מתודה	URL
קבלת כל הלקוחות המחזיקים כעת ברכב	GET	clients/active
קבלת כל הרכבים הפנויים - ניתן יהיה לשלוח פרמטר המתאר שם יצרן ובכך לקבל את כל הרכבים הפנויים המתאימים לשם היצרן	GET	vehicles/available

		המבוקש
vehicles/belowPrice/{price}	GET	קבלת כל הרכבים בעלי מחיר נמוך מזה המצויין במשתנה price
clients/late	GET	קבלת כל המזהים האוניברסלים השייכים ללקוחות אשר מחזיקים ברכב מעבר לזמן המוסכם להחזירו
checkin/{clientUuid}/{vehicleUuid}	POST	שידוך רכב ללקוח - בנתיב הפנייה יועברו המזהים האוניברסלים של הלקוח והרכב על פי הסדר בהתאמה. לאחר פעולה זו הרכב יסומן כלא זמין. עבור מזהים אוניברסלים לא קיימים יש לזרוק חריגות מותאמות אישית - כאלו היורשות מ RuntimeException ומתארות את הבעיה. עבור רכב שאינו זמין יש גם כן לזרוק חריגה על פי אותם החוקים.

AutoRest

הוראות מבנה

הקפידו על המודל התלת שכבתי:

Repositories

- ClientRepository
- VehicleRepository

Service

הממשק AutoRestService יגדיר את כל פונקציות השירות והמימוש AutoRestServiceImpl יחזיק כ reference את ClientRepository ו VehicleRepository.

בשכבה זו השתמשו ב Mapper מתאים שיאפשר המרה מ ClientDto ל Client ולהפך ובדומה מ VehicleDto ל Vehicle ולהפך.

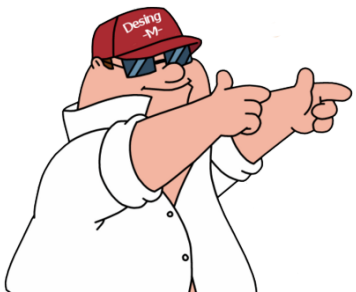
בונוס: למדו כיצד להשתמש בספרייה MapStruct בכדי לכתוב את ה Mapper.

חריגות: בשכבה זו תזהו מקרים חריגים ותזרקו חריגות מתאימות - כאלו היורשות מ RuntimeException.
למשל, עבור מזהה אוניברסלי של לקוח לא קיים, זרקו חריגה בשם, למשל, NoSuchClientException.

השאירו את ה Controller נקי ככל האפשר - בהמשך נראה כיצד ניתן לנתב חריגה למקום המטפל בה ומחזיר לצד הלקוח משוב מתאים.

Controller

המחלקה AutoRestController תשלוט בנתיב הבקשה ותפנה לפונקציות מתאימות בשכבת ה Service.



AutoRest

בהצלחה!