|  |  |  |  |
| --- | --- | --- | --- |
| Revizyon Tarihçesi | | | |
| Tarih | Versiyon | Hazırlayan | Açiklama |
| 25.02.2023 |  | Özgür Yatmaz | Camel Transport App dokumnai hazirlandi |

**İçindekiler**

[1 API Hakkında Genel Bilgi 3](#__RefHeading___Toc1209_801178615)

[1.2 My Sql Database Ayarları 3](#__RefHeading___Toc19292_943653133)

[1.3 Databasede Log Tablosu Olusturma 4](#__RefHeading___Toc4420_1627051137)

[1.*4 Test Adımları* 4](#__RefHeading___Toc6703_3143813541)

[2 Servis operasyonları 5](#__RefHeading___Toc11629_1023073916)

[2.1. GetTransportedApples Operasyonu 5](#__RefHeading___Toc13755_1543940018)

[2.1.1. Operasyon Request JSON / REST Mesaj 6](#__RefHeading___Toc73825_1966934515)

[2.1.2. Operasyon Response JSON / REST Mesaj 6](#__RefHeading___Toc73827_1966934515)

[3 Loglarin Yönetilmesi 7](#__RefHeading___Toc4424_1627051137)

[4 API Durumu Izleme 7](#__RefHeading___Toc5039_1627051137)

# 1 **API Hakkında Genel Bilgi**

Bu api iki kısımdan oluşmaktadır. Birinci kısmı standart rest apidir. Bu apide tek bir operasyon bulunmaktadır. Bu operasyon iki parametreye (elma sayisi ve mesafe) bağlı olarak bir deve ile kaç adet elma taşınabileceği bilgisini consümera dönmektedir. Şayet taşıma mümkün değil ise bunun sebebini dönmektedir. Ayni zamanda tüm sorgular cevapları ile birlikte MySql data base loglanmaktair. Loglama işlemi için Log4j2 kütüphanesi kullanılmıştır.

API nin ikinci kısmı spring mvc ile yapıldığından sadece web browser kullanılarak databasedeki loglara erişim sağlamaktadır. Ayni zamanda dilenilen loglarin silinebilmesi için her logun yanında delete butonu bulunmaktadır. Loglarin görüntülenmesi ve silinmesi işlemleri spring data jpa kullanılarak yapılmaktadır.

## 1.2 **My Sql Database Ayarları**

Database credentialleri Application.properties dosyasından yapılmaktadır.

|  |
| --- |
| **Application.properties** |
| spring.mvc.view.prefix=/WEB-INF/jsp/  spring.mvc.view.suffix=.jsp  spring.datasource.url=jdbc:mysql://localhost:3306/<DB adi>  spring.datasource.username=<kullanici adi>  spring.datasource.password=<kullanici sifresi>  spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  spring.jpa.properties.hibernate.dialect =org.hibernate.dialect.MySQL8Dialect |

| **Alan Adi** | **Açiklama** |
| --- | --- |
| spring.mvc.view.prefix | Spring mvc nin hangi clasore bakacagını belirtir (degistirmeyin) |
| spring.mvc.view.suffix | Spring mvc nin hangi uzantılı dosyaları isleme alacagini belirtir (degistirmeyin) |
| spring.datasource.url | Database urli |
| spring.datasource.username | Data base kullanıcı adı |
| spring.datasource.password | Data base kullanıcı sifresi |
| spring.datasource.driver-class-name | Sql driver adı |

## 1.3 **Databasede Log Tablosu Olusturma**

Asagıdaki sql script calıstırılarak loglar icin gerekli tablo olusturulmalıdır.

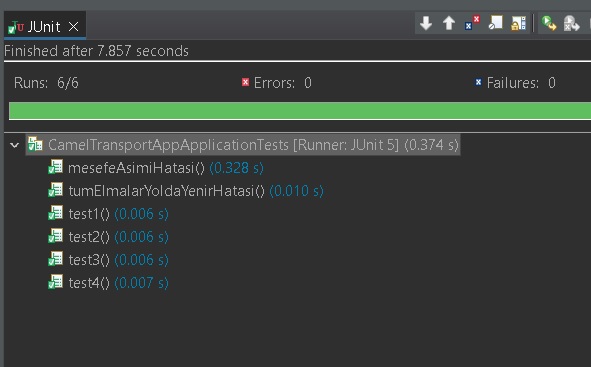
|  |
| --- |
| **Data Base log table olusturma sql scripti:** |
| CREATE TABLE `log` (  `ID` varchar(36) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL,  `DATE\_TIME` timestamp NOT NULL,  `CLASS\_INFO` varchar(255) NOT NULL,  `LEVEL` varchar(50) NOT NULL,  `MESSAGE` text,  `EXCEPTION\_INFO` text,  PRIMARY KEY (`ID`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci; |

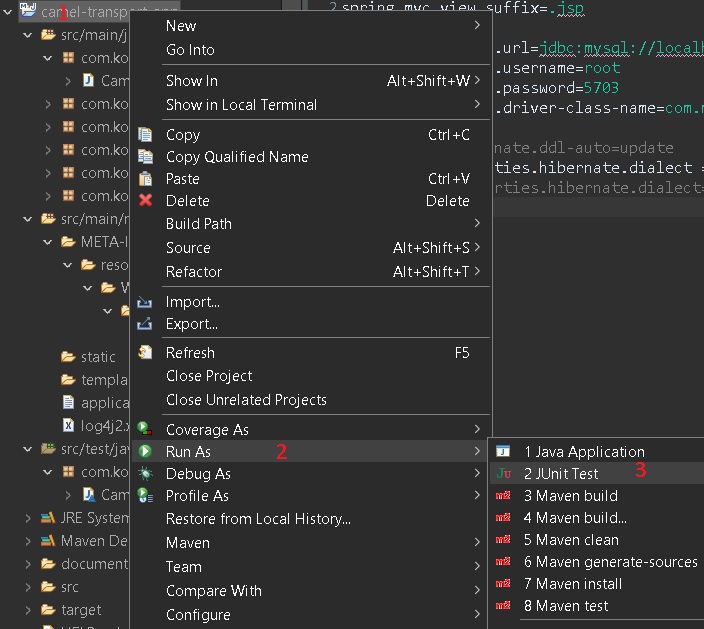
## 1.***4 Test Adımları***

Unit testler için görselde gösterildiği gibi proje Eclipse import edidikten sonra önce projeye sağ tıklanır ardından Run as Junit test denir. Ya da maven yardımı ile terminalden projenin root directorisine geldikten sonra “mvn test” komutu çalıştırılır.

Unit testler icin 6 adet farklı case test edildi ve bir hata gözlemlenmedi.

Ayrıca apinin kullanılmasi icin postman collection ekte saglanmistir.



******

# 2 Servis operasyonları

## **2.1. GetTransportedApples Operasyonu**

Deve ile belirtien mesafeye kac adet elma tasınabılecegini hesaplayan consiyondur.

| **Request** | | | |
| --- | --- | --- | --- |
| CamelTrasportRequest | | | |
| **Alan Adı** | **Tip** | **Zorunlu** | **Açıklama** |
| elmaSayısı | int | Evet | Yasima işlemi icin kac adet elma oldugu bilgisi |
| mesafe | int | Evet | Taşıma işleminin kaç kilo metreye yapılacağı bilgisi. |

| **Response** | | | |
| --- | --- | --- | --- |
| CamelTransportResponse | | | |
| **Alan Adı** | **Tip** | **Zorunlu** | **Açıklama** |
| tasinabilirElmaSayisi | int | Evet | Eger tasıma islemi yapılabilir ise kac adet elmanın tasinabilir olğu bilgisi |
| status | String | Evet | İslem durumu, tasima islemi yapılamaz ise sebebini bildirir, yapılabilir ise “succes” döner |

### **2.1.1. Operasyon Request JSON / REST Mesaj**

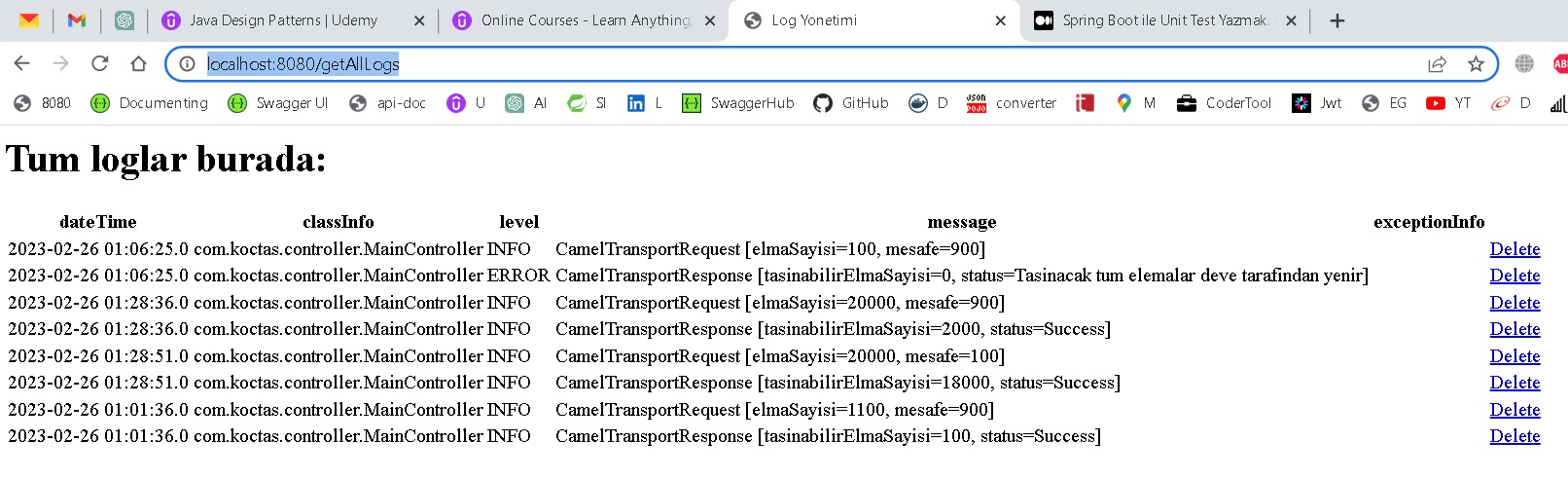
| **Request REST** |
| --- |
| **Request Url:** http://localhost:8080/api/v1/getTransportedApples |
| {      "elmaSayisi": 20000,      "mesafe": 100  } |

### **2.1.2. Operasyon Response JSON / REST Mesaj**

| **StartPayment Response REST** | |
| --- | --- |
| **Rest -** | |
| SUCCESS | {      "tasinabilirElmaSayisi": 18000,      "status": "Success"  } |
| FAILED | {      "tasinabilirElmaSayisi": 0,      "status": "Bir deve 1000 kilometreden fazla tasiyamaz"  } |
| {      "tasinabilirElmaSayisi": 0,      "status": "Tasinacak tum elemalar deve tarafindan yenir"  } |

# **3 Loglarin Yönetilmesi**

DB deki tüm loglar yalnızca web browsere “<http://localhost:8080/getAllLogs>” urli yazılarak görüntülenebilmektedir. Herhangi bir log delete buttonu ile silinebilmektedir.



# **4 API Durumu Izleme**

Api durumunu izleyebilmek için actuator eklendi. <http://localhost:8080/actuator> urlini kullanarak api hakkında tüm durum bilgilerine ulaşılarak apinin zafiyetleri konusunda bilgi sahibi olunabilir. Not etmek gerekir ki talep edilmediğinden spring security kullanılmadı. Bu sebeple api publictir(herkes ulasabilir) ve güvenliği spring bootun default sağladığı önlemler ile sınırlıdır.

