Toteuttakaa joitakin JUnit-yksikkötestejä, integraatiotestejä ja end-to-end -testejä ja kuvailkaa lyhyesti

- miten ne tehtiin ja
- millä perusteella ne ovat juuri yksikkö/integraatio/E2E-testejä.

Integraatiotestit:

Käytössä SpringBootTest, h2 testitietokanta ja MockUser ylläpitäjäoikeuksilla.

Test-kansion alle application-test.properties ja luokat lippujen ja myyntien testaamiseen.

Mitä testataan: Lipun luominen oikeilla syötteillä

Syöte:

```
String uusiLippuJson = """
        {
            "myynti_id": %d,
            "tapahtuman_lipputyypit_id": %d,
            "hinta": 30.00
        }
        """.formatted(myyntiId, tapahtumanLipputyyppiId);
```

Odotettu lopputulos: Json-tyyppinen vastaus, status isCreated ja sisältää koodi-attribuutin.

Mitä testataan: Lipun luominen väärillä syötteillä

Syöte:

```
{
                    "myynti_id": 9999,
                    "tapahtuman_lipputyypit_id": %d


                }
                """.formatted(tapahtumanLipputyyppiId);
```

Odotettu lopputulos: status badRequest ja virheviesti

Mitä testataan: kaikkien lippujen hakemista, yhden lipun hakemista, kaikkien myyntien hakemista ja yhden myynnin hakemista

Syöte: ei syötettä

Odotettu lopputulos: status isOk, sisältö Json-muodossa

Mitä testataan: Myynin luominen oikeilla syötteillä

Syöte:

```
{
            "liput": [],
            "myyntipiste": {
                "myyntipisteId": %d
            },
            "maksutapa": {
                "maksutapa_id": %d
            }
        }
        """.formatted(myyntipisteId, maksutapaId);
```

Odotettu lopputulos: Json-tyyppinen vastaus ja status isCreated.

Mitä testataan: Myynnin luominen väärillä syötteillä

Syöte:

```
{
                    "liput": [],
                    "myyntipiste": {
                        "myyntipisteId": %d
                    },
                    "maksutapa": {
                        "maksutapa_id":999
                    }
                }
                """.formatted(myyntipisteId);
```

Odotettu lopputulos: status NotFound ja virheviesti

Miksi integraatiotestejä: Testataan sovelluksen ja tietokannan välistä toimintaa:

- oikeat statuskoodit ja virheilmoitukset
- oikein muodostetut pyynnöt menevät läpi - esim. lipun tallennus
- virheelliset pyynnöt eivät tallennu tietokantaan

JUnit yksikkötestit:

Testattiin Lippu ja Myynti luokkien toimintaa.

Pienillä yksikkötesteillä testataan, että ohjelman pienet osaset toimivat kuten pitää. Ne ovat yksikkötestejä nimenomaan niiden tarkan rajauksen vuoksi.

Lippu luokkaa testattiin luomalla uusi lippu, tallentamalla se repositorioon ja lopuksi haettiin tallennettu lippu ja tarkastettiin oliko se saanut id:n ja koodin.

Toisessa tapauksessa luodaan uusi lippu ja testataan löydetäänkö luotu lippu LippuRepositorion findByKoodiilla. Haettua lippua verrataan lopuksi aluksi luotuun lippuun.

Kolmannessa tapauksessa testattiin miten lipun hinta käyttäytyy, jos siihen laitetaan raja-arvoja sekä kiellettyjä arvoja, eli negatiivisia arvoja.

Myynti-luokkaa testattiin luomalla uusi myynti, tallentamalla se ja hakemalla tallennetun myynnin id.

Myynnit luokan myyntien listan hakua testattiin luomalla kaksi myyntiä, tallentamalla ne ja sen jälkeen hakemalla myyntien lista. Tämän jälkeen testataan, että listan koko on suurempi kuin 1.

Myynti-luokassa testattiin myös myynnin päivämäärää. Myynti luokasta haettiin yksi myynti ja tämän jälkeen testattiin, että myynnin päivämäärä on tyyppiä LocalDate.

End-to-End testit

Testasimme end-to-end testit manuaalisesti Rahti-ympäristössä. Testauksen kohteet olivat:

- Sisään- ja uloskirjautuminen
- Lippujen myyminen
- Lippujen hakeminen
- Lippujen selaaminen
- Lippujen käytetyksi merkkaaminen