

Criteria Tennis opdracht

Mate van functionaliteit

De regels van tennis was ik niet bekend mee, maar uiteindelijk is het mij toch gelukt om de puntentelling en verschillende statussen in werking te krijgen.

Bij HTTP/REST heb ik duidelijke http requests gebruikt voor de desbetreffende taken. Dat kun je zien in de presentation layer (GET, POST, PUT, etc.). De URL's zijn ook duidelijk vind ik, omdat ik eerst onderscheid heb gemaakt tussen objecten en daarna voor de desbetreffende requests path variables gebruik indien nodig.

De navigatie tussen layers heb ik volgens de regels geprobeerd uit te voeren. Bijvoorbeeld in de architectuur van de application layer wordt er bij SetService met de services van een andere component (bijvoorbeeld PlayerService) gepraat, en niet rechtstreeks met de onderliggende infrastructuur (PlayerRepository).

Voor de architectuur heb ik ook op de packagestructuur gelet, de classes en interfaces zijn onderverdeeld in de desbetreffende packages.

Ik heb opgelet om de verantwoordelijkheden te scheiden en ieder object in zijn eigen rol te laten werken.

Fouthandelingen worden netjes opgepakt met exceptions die ik heb aangemaakt, en ze worden met passende status-codes gestuurd wanneer het nodig is. Zoals PlayerNotFoundException, SetHasAlreadyEndedException.

De persistentie vindt plaats in h2 database, het werkt naar behoren.

Tests schrijven kwam ik er helaas niet aan toe, ik heb wel vertrouwen dat de tests ook goed zou lukken. Ik heb de must-haves eerst afgerond. Uiteindelijk heb ik ook de database erbij gepakt en de errorhandling.

Postman heb ik gebruikt om de requesten handmatig te testen.

Mijn code vind ik best netjes, omdat ik gebruik heb gemaakt van witregels en onderscheiding van verschillende stappen, waardoor de leesbaarheid een stuk verbeterd is. Ik vind ook dat de naamgeving duidelijk is.

Succes met het beoordelen,

Abdullah Sanli