

Inlämningsuppgift 1 – Web API för TopStyle

Beskrivning

Uppgiften handlar om att skapa ett web api som körs i Azure. Företaget TopStyle som säljer kläder och skor online behöver en webbshop och du skall bygga backenddelen för den. Lösningen skall skrivas med ASP.NET Web api och gå mot en SQL databas som sätts upp i Azure. Du måste använda Entity framework för kommunikationen med databasen.

Krav för att lösningen skall bli godkänd

- Det skall gå att söka produkter. En produkt har ett namn, en beskrivning, pris och tillhör en kategori. Exempel på kategorier är tex byxor, skjortor/blusar, tröjor. Titta gärna på exempel från någon webbshop som säljer kläder.
- Det skall gå att skapa ett användarkonto och sedan logga in med det kontot. Alla användare skall hanteras via databasen och det skall returneras en JSON web token när inloggningen är klar. En inloggad kund skall bara kunna se sina egna uppgifter.
- En beställning/order skall kunna skapas som består av ett antal produkter. För att göra detta måste användaren vara inloggad dvs det måste skickas med en token för att metoden skall kunna köras.
- Alla uppgifter sparas i en databas som skall deployas till Azure. Kommunikationen med databasen skall ske med Entity framework och du skall använda code first.
- API:et skall deployas till Azure och du måste öppna upp så att det kan kommunicera med databasen. Connectionsträngen skall hanteras via en Azure keyvault.
- Det skall ha stöd för all denna funktionalitet och vara uppbyggt enligt de principer som vi pratat om på förra kursen och dokumenteras med Swagger.
- Github skall användas för att lagra och versionshantera all kod. Det är tillåtet att titta på tutorials men inte att följa dessa steg för steg eller kopiera färdiga lösningar.

Extra krav för väl godkänd

- Användarhanteringen skall göras med Core Identity. Det skall finnas en Admin inloggning och loggar man in med den skall bara admin funktionalitet visas. En admin skall kunna lägga till, uppdatera och avaktivera produkter. Om en produkt inte finns med på någon beställning skall den kunna tas bort. Den skall också kunna komma åt alla beställningar och ta bort och uppdatera en beställning.
- Alla endpoints skall vara asynkrona. Det betyder att koden skall var asynkron och TPL skall användas.

- Loggning skall aktiveras i Azure och applikationen skall skriva till loggen vid felhantering.
- Större krav på bra arkitektur och kod enligt de riktlinjer vi gått igenom på lektionerna.

Lösningen ***måste göras objektorienterat***. Koden skall fungera och web api:et skall gå att köra utan fel dvs alla metoder skall kunna anropas via POSTMAN och de skall returnera lämpliga koder och data .

Betyg

Det går att få G, IG och VG på uppgiften. Den är obligatorisk för att kunna få betyg på kursen och du måste göra VG delen för att kunna få VG på hela kursen.

Redovisning

Redovisning sker genom att du visar läraren den fungerande uppgiften. Skicka INTE in något via mail! Senaste tidpunkt för redovisning är sista dagen sista veckan på kursen.