

Projekt i kursen Dataåtkomster i .NET

Bakgrund

Din uppgift är att utveckla en datamodell och en "code first" databas med .NET Entity Framework Core, samt en därtillhörande ASP.NET Core Restful WebApi server för en reseapplikation.

WebApiApp ska ha en modern mjukvaru-arkitektur i lager som formar en mjukvara stack. Applikationen skall förutom att använda EntityFrameworkCore använda ASP.NET Core för konfiguration, Loggning och dependency injection.

WebApi servern ska tillhandahålla ett antal tjänster som läser, skriver, uppdaterar och tar bort dataposter i databasen, så som orter, sevärdheter, kommentarer och användare.

En användare ska kunna använda alla av WebApi tjänster med en webbrowser genom WebApis Swagger gränssnitt och få resultatet i json dataformat.

Genom Swagger ska en användare kunna, genom att använda tjänsterna:

- Se sevärdheter och dess kategori (typ restaurang, cafe', arkitektur), rubrik och beskrivning.
- ange en ort och som resultat få alla rekommenderade sevärdheter, dess rubrik, kategori och beskrivning
- För valfri sevärdhet ska en användare kunna lägga till en kommentar.
- Se alla användares kommentarer för en viss sevärdhet.

Betyg

Du ser nedan vad som krävs för G, respektive VG

Projektinlämning

Projektet är individuellt. Du ska alltså lämna in en egen lösning, som innehåller.

- A. Ditt namn
- B. ER modellering och din förklaring till punkt 1 – G nivå.
- C. Zip fil med din Visual Studio Solution.
- D. I zip filen skall även innehålla din user-secret fil
- E. Jag ska kunna öppna, bygga och köra din WebApi tjänst i Visual Studio genom dess Swagger gränssnitt utan kompilator eller run-time fel.

Gå noga igenom ovan steg innan du lämnar in. Jag kommer att returnera inlämningar som inte uppfyller A-E.

Uppgifter för G nivå

1. Modellera databasen i C# för Entity Framework Core (EFC). Visa modelleringen i ett ERD diagram med crow-foot notation.
 - Förklara kortfattat hur du tänkt när du skapat C# klasserna som utgör tabellerna i databasen
2. Skapa en WebApiApp med en modern mjukvarustack. Du får använda den mall av mjukvaru stacken som jag ger i Lektion 1 och du får använda kod från undervisningen i ditt projekt.
3. Konfigurationsdata skall komma från appsettings.json samt från user-secrets.
4. Implementera databasen och dess tabeller, index och keys enligt EFC "code first" principen.
 - Anpassa databasen, t ex PrimaryKey, ForeignKey, Index, med EFC Annotations
5. Skapa WebApi tjänster, som lägger in testdata i databasen genom "robust seeding".
 - minst 50 användare,
 - minst 100 orter slumpmässigt fördelade över minst 4 länder
 - minst 1000 sevärdheter fördelade slumpmässigt över adresser på orterna
 - Varje Sevärdhet skall ha slumpmässigt mellan 0 och 20 kommentarer
6. Skapa WebApi tjänster som:
 - Visar alla sevärdheter filtrerade på kategori, rubrik, beskrivning, land, och ort
 - Visar alla sevärdheter som inte har någon kommentar
 - Visar en sevärdhets kategori, rubrik, beskrivning, och alla kommentarer
 - Visar alla användare och de kommentarer som användaren har lagt in
7. Skapa WebApi tjänster som tar bort all testdata från databasen.

Uppgifter för VG nivå (fortsättning på G nivå)

8. Projektet enligt G nivå.
9. Skapa WebApi tjänster som:
 - Lägger till en användare till databasen
 - Lägger till en sevärdhet till databasen
 - Låter en användare lägga till en kommentar till en sevärdhet
 - Ändra en sevärdhets kategori, land, ort, rubrik och beskrivning
10. Skapa WebApi tjänster som:
 - Tar bort en användare från databasen. Alla kommentarer som användaren skapat skall också tas bort.
 - Tar bort en sevärdhet från databasen. Alla kommentarer som hör till sevärdheten skall också tas bort.
 - Tar bort en kommentar från sevärdheten.

Lycka till!