

Avancerad webbutveckling

Övningsuppgift 2

Nedan följer beskrivning av övningsuppgiften med en given handling - nämligen geografiska platser. Det är fritt fram att hitta på en en kontext, men se gärna till att ni får med de olika delarna/rubrikerna nedan.

Grunduppsättning

Skapa en tom applikation i form av ASP.NET Core MVC 2.0. Med hjälp av Bower, eller annan pakethanterare, hämta hem Bootstrap.

Skapa Layout, ViewStart och ViewImports-filer och använd dessa på korrekt sätt.

Default route

Ställ in din routing så att en annan controller än Home (med action method Index) används som start-route.

Model, interface, repository och dependency injection

Skapa en domänmodell (Location) med följande egenskaper:

- Id (int)
- Name (string)
- Latitude (string)
- Longitude (string)

Skapa ett interface (ILocationRepository) som kommer ligga till grund för repositoryet som senare ska returnera en IEnumerable<Location>

Skapa sedan ett FakeLocationRepository som implementerar interfacet och returnerar x antal hårdlödda platser.

Registrera ert IRepo och FakeRepo som servisar för dependency injection. Se sedan till att er controller får tillgång till repot. Skicka sedan vidare datat till en vy och presentera datat i någon form av lista och använd er av några av Bootstraps klasser.

Entity Framework

Skapa nu ett "riktigt repository" och se till så att detta repo hämtar sin data från MS SQL. Seed data i databasen eller kör INSERT-statements i SSMS för att få data på plats. Den här delen kräver att ni aktiverar Migrations och använder er av Package Manager Console och olika kommandon. Tips: Kom ihåg *IDesignTimeDbContextFactory*.

Partial view

Skapa en partial view som har till uppgift att presentera platsinformation om en geografisk plats. Se sedan till att denna partial view används i foreachen som itererar över modellens platser i vyn.

Tag helper

Skapa en tag helper som har till uppgift att presentera en statisk kartbild från Google Maps kart-API via ett vanligt -element tillsammans med den övriga informationen om en geografisk plats i er applikation.

Exempel på src-värde på img-elementet skulle kunna se ut så här:

<https://maps.googleapis.com/maps/api/staticmap?center=57.188182,%2014.036845&markers=color:red|label:C|57.188182,%2014.036845&zoom=16&size=400x200> ger följande bild:



Tag helpern ska ha stöd för att dynamiskt (från vyn) sätta bredd, höjd och zoom. En marker ska placeras på kartan med avseende på vad modellens värden på Latitude och Longitude är. Sedan ska även första bokstaven på platsens namn placeras i knappnålen. Ovan ser vi **C**ampus Värnamo, och i knappnålen så är bokstaven **C**.

Unit testing

Skapa upp ett xUnit projekt parallellt med det andra projektet och installera Moq och skapa några test-metoder som testar något i en controller och i ett (fake)repository.

Förväntat resultat

Bilden nedan genereras av en tag helper och producerar följande HTML:

```

```

Tag helpern i vyn se ut i stil med detta, dock med påhittade attribut och värden:

```
<img a="1" b="2" c="3" d="4" />
```

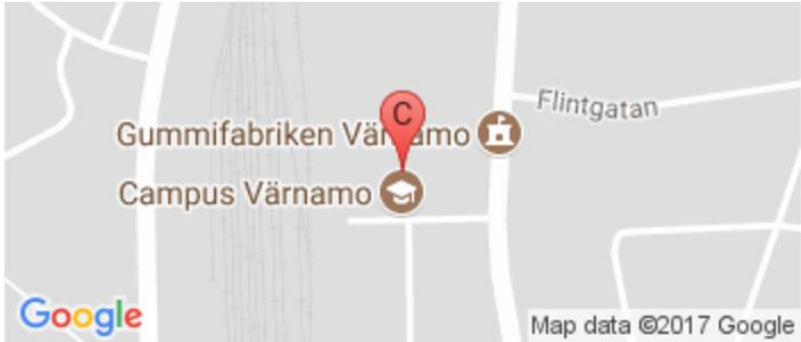
Se nästa sida för resultatet.

Locations

localhost:1657


ApparBlandatMiscKunderÖvriga bokmärken

Campus Värnamo



Map data ©2017 Google

Juneporten



Map data ©2017 Google

Nybergs Bil Jönköping