Tekniktest

Instruktioner och övningar

Innehållsförteckning

[1 Uppgifter 2](#_Toc19799773)

[2 Att tänka på: 3](#_Toc19799774)

# Uppgifter

1. I klassen LibraryRepository.cs finns det en hårdkodad länk till databasen (.xml filen) vi vill att värdet inte sätts i kod utan flyttas ut till web.config filen eller en app setting för API’t så att en användare själv kan ändra sökvägen vid behov utan att kompilera om koden.
2. I klassen BookInfoController finns det en exempelmetod som söker efter en bok via ISBN koden (streckkod på baksidan av en bok) vi vill att du bygger en enkel internetsida som kan skicka och ta emot informationen mot API kontrollen.
   1. Sidan får vara byggd i som en enkel html sida, en ASP.NET sida eller en MVC sida
   2. Förfrågan mot API’t ska skickas och hanteras via javascript, JQuery eller AngularJS
   3. Sidan ska klara av att presentera informationen som kommer tillbaka

***PS. Det går att få mer information om API controllern genom att köra det bifogade projektet och klicka till API fliken och sedan klicka på metoden***

1. Till en början så accepterar API’t alla anrop utan någon typ av felhantering. Vi vill att strängen som skickas in valideras mot null och tomma strängar.
   1. Ifall en sträng är null, tom eller endast innehåller blanksteg ska ett felmeddelande returneras i svaret och applikationen ska inte krascha.
   2. I nuvarande läge i klassen LibraryRepository.cs så skickas en bok tillbaka med titeln ”Book not found!” det här behövs hanteras på din hemsida så det inte presenteras på samma sätt som att en bok är funnen, dvs att om du väljer att presentera objektet så skall det inte stå ”Title: Book not found!” utan det ska vara tydligt att ett fel har inträffat. Bygg om på valfritt sätt i API’t eller på din hemsida.
2. Funktionen för sök är inte tillräckligt bra för din hemsida. Sökfunktionaliteten hämtar just nu ut endast en bok, dvs. en specifik serienummersökning på ISBN numret. Vi vill att det även ska gå att söka på titel eller författare av boken och få tillbaka alla träffar och inte endast en. Självklart vill vi fortfarande kunna söka på ISBN nummer.
3. En kund har begärt att vi utökar API’t med funktionalitet för att skapa nya objekt i biblioteket. Bygg en funktion som klarar av att föra in nya böcker i .xml filen.
   1. Metoden ska vara av typen POST
   2. Metoden ska från kontrollen anropa en SET metod i LibraryRepository
   3. SET metoden i repositoryt ska endast kunna anropas inom assemblyt
   4. Metoden ska ta emot följande parametrar author, title och ISBN
4. Den nya metoden i API’t är nu implementerad, bygg en ny flik, vy eller dylikt på din hemsida som kan skicka nya värden till API’t
   1. Den ska validera att inga värden får vara tomma eller innehålla blanksteg.
   2. Ifall något fält är tomt eller innehåller blanksteg så ska ett felmeddelande dyka upp på användarens sida
   3. Funktionaliteten för att hantera detta ska byggas i javascript, JQuery eller AngularJS
5. Bygg en funktion på hemsidan som skickar in bokförslag till API’t om vad som bör läggas till i biblioteket.
   1. Anropet ska ta emot namnet på personen som skickar in förfrågan, författarens namn och en boktitel.
   2. API’t ska kontrollera att det finns en mapp i projektet för dagens datum, om det inte finns så skapar den en.
   3. I den existerande/nyskapade mappen så ska en fil skapas av valfri filtyp och ska innehålla titeln och avsändaren
   4. Filnamnet ska bestå av första bokstaven i personen som skickar in förfrågans namn följt av ett \_ och datumet följt av ett \_ samt första bokstaven i författarens namn.
   5. Ifall en liknande fil redan finns så läggs en filändelse till med ett s.k. auto-increment (räknare).

***Filnamnet ska alltså se ut som följande om Peter Barsaum skickar in frågan om boken ”Programmering” av ”Harry Kodknackare” P\_2019-01-31\_H. Om filen redan finns blir filnamnet P\_2019-01-31\_H1, P\_2019-01-31\_H2 osv.***

# Att tänka på:

* Försök att inte bygga för stora metoder utan försök att tänka på att man vill kunna återanvända koden där man kan.
* Lägg inte för mycket tid på att designa hemsidan du bygger, vi är mer intresserade av hur du bygger funktionaliteten än hur sidan ser ut. Vi uppskattar självklart en fin sida men det är inte syftet.
* Om du finner ett syfte i att lägga fler .xml element i filen gör det, ni är fria att använda hela projektet som ni vill i scopet av denna uppgift.
* Det viktiga är inte att ni klarar av att göra alla uppgifter utan att ni utför dem så bra som möjligt.
* Dem uppgifter ni inte klarar av ser vi gärna att ni försökt påbörja så kan vi bedöma delen av det ni klarat av.
* Se till att när projektet skickas tillbaka så ska det gå att bygga projektet utan problem.
* När ni lägger till böcker är det inte viktigt att riktiga böcker läggs till, det är nog bara jag som läser för mycket.