

Uppgift Xamarin – last.fm

Beskrivning

Skapa med hjälp av Xamarin en mobilapplikation där man kan söka efter artistinformation mot last.fm. Svaret ska visas i användargränssnittet för applikationen. Följande information ska visas:

- Name
- Content summary
- YearFormed
- URL
- Published
- Någon av bilderna som returneras i svaret från last.fm

För att kommunicera mot last.fm finns ett fritt API som man kan använda (kräver att man registrerar ett konto så att man får en nyckel som skickas med vid anrop som parameter "api_key").

Alternativ

För de som vill finns det möjlighet att jobba med en egen uppgift (inte individuellt – ni ska arbeta i grupper enligt indelningen nedan - men alltså med en annan uppgift/app än den som beskrivs här). Den egna uppgiften ska då ungefär motsvara uppgiften nedan i omfattning. Väljer ni att arbeta med en egen uppgift, stäm av med Jens-Peter innan ni börjar.

Valfria tillägg

Inte obligatoriska, man kan välja mellan att göra något av tilläggen nedan, alla eller inget alls.

- Visa också albuminformation inkl. bilder på omslag
- Lägg till informationen från XML-elementet "similar" och lägg de artister som listas där som länkar i applikationen (d.v.s. när man väljer en "similar" så görs automatiskt en sökning på denna artist)
- Lägg in en länk till artistens hemsida på last.fm (URL ingår i svaret från last.fm) och möjlighet att navigera till länken
- Utnyttja andra API-funktioner mot last.fm och lägg in fler funktioner i programmet
- Utöka sökfunktionen så att man kan söka på albumnamn, år och artist var för sig eller kombinerat

Bedömning

Bedömning sker utifrån tre olika kriterier:

- Funktionalitet, de obligatoriska och valfria delarna ovan samt eventuella övriga funktioner
- Kvalitet och design kodmässigt
- Hur bra lösningen fungerar som app, alltså användarvänlighet, prestanda, hur visuellt genomarbetad den är o.s.v.

Frågor

Eftersom kursen bedrivs i form av självstudier finns inte möjlighet till teknisk support, programmeringshjälp etc. Tolkningsfrågor runt uppgiften eller praktiska frågor runt inlämning el. dyl. hanteras via mail med Jens-Peter eller Mikael.

Gruppindelning

Arbeta i följande grupper:

Belal – Kristofer - Elisabeth
Alexander A – Kenny - Mats-Ola
Christopher – Isak – Hannah
Charles – Isabelle - David
Christian – Samuel - Rasmus
Richard – Tommy - Max
Bashar – Adam - Björn
Robin – Daniel - Bart
Pontus – Mikael – Alexander W
Malin – Oskar - Jim
Patrik – Greger

Examination

Denna uppgift utgör hela examinationen och hela betygsunderlaget. Alla i gruppen kommer få samma betyg (IG/G/VG). Lösningen ska vara inlämnad i kursrummet i Fronter senast 17 november kl. 8.00. Lämna in zippad källkod, det räcker givetvis att en gruppmedlem skickar in koden.

För de som får IG eller inte blir klara p.g.a. sjukdom el. dyl. finns *en* möjlighet till komplettering (skickas med e-post till jens-peter.olsson@plushogskolan.se senast den 30 november). Därefter ges kursbetyget IG om man inte lämnat in en godkänd lösning.

Kod

Om man t.ex. använder RestSharp och vill få automatisk konvertering (genom att anropa `Execute<Artist>` o.s.v.) mellan texten som returneras via http och klasser i C# måste man skapa klasser enligt exakt den (ibland något märkliga) struktur som last.fm arbetar med. Nedan en grundläggande mappning som fungerar och som ni kan utgå ifrån för artistinformation.

```
public class image
{
    public string Size { get; set; }
    public string Value { get; set; }
}

public class Biography
{
    public string Summary { get; set; }
    public string Content { get; set; }
    public int YearFormed { get; set; }
    public DateTime Published { get; set; }
}

public class ArtistImageCollection : List<image> { }
```

```
public class Artist
{
    public string Name { get; set; }
    public string Mbid { get; set; }
    public string Url { get; set; }
    public Biography Bio { get; set; }
    public ArtistImageCollection Image { get; set; }
}
```