

Koduppgift

Översiktlig beskrivning

Du skall bygga en bankapplikation. Applikationen använder en databas för att lagra konton.

Varje konto har:

- Namn
- Nuvarande saldo
- Transaktionslista, en lista på varje belopp som är insatt eller uttaget.

Applikationen skall vara en console applikation där man interagerar med hjälp av tangentbordsinmatning.

Funktion

Applikationen startas.

[Fråga efter kontonamn]

Applikationen skall fråga efter namn på bankkonto.

Man skall sedan mata in namnet på kontot. Om ett tomt kontonamn anges så går applikationen vidare till steget **[Visning av bankens totala balans]**

Applikationen kommer att fråga efter ett belopp att sätta in eller ta ut.

Man skall nu mata in ett belopp som är ett heltal, positivt (insättning) eller negativt (uttag) eller noll.

Om man anger ett felaktigt belopp, dvs. en textsträng som inte är ett heltal enligt definitionen ovan skall den meddela att beloppet är felaktigt och fråga efter belopp igen.

När du angivit ett korrekt belopp skall bankkontot uppdateras, alternativt skapas (om kontonamnet inte redan finns) med ny balans och beloppet som angivits skall registreras i transaktionslistan.

Bankkontots balans efter insättning/uttag skall nu visas för användaren.

Applikationen skall nu återgå till **[Fråga efter kontonamn]** och användaren ges möjlighet att göra nya insättningar/uttag på bankkonton.

[Visning av bankens totala balans]

Applikationen visar nu summan av alla bankkontons balans. Väntar på att användaren trycker ENTER innan den slutligen avslutas.

[SLUT]

Tekniska förutsättningar

- Applikationen skall byggas i C# som console applikation.
- Du får använda vilken .NET version som helst, men referens projektet är utvecklat i .NET framework 4.7.2
- Databasen som används skall vara MongoDB
- Connectionsträngen till mongodb servern skall ligga i App.config
- Insättning/uttag skall vara atomiska (Dvs. fungera i en multi-användarmiljö där samma konto skulle kunna hanteras samtidigt av flera användare)
- Du får använda vilka tredjepartskomponenter du vill

Översiktligt om MongoDB

Mongodb är en dokumentdatabas som lagrar dokument i något som heter collections (närmast jämförbart med tabeller) som ligger i en databas, som i sin tur ligger på en mongodb server.

MongoDb jobbar med BSON dokument (förenklat binär JSON), oftast räcker det att tänka att det helt enkelt är JSON dokument som lagras.

MongoDb är atomisk på dokumentnivå.

Mongodb går att ladda ner gratis (community edition) från:

<https://www.mongodb.com/download-center/community>

Det finns officiella nuget drivers för .NET: <https://www.nuget.org/packages/mongodb.driver>

Det finns flera GUI's för att interagera mot Mongodb. Den jag rekommenderar är Studio3T som man kan testa gratis ett antal dagar: <https://studio3t.com/download/>
(Tips, tryck på "More download options" och välj din version så slipper du signa upp på newsletter etc)

Förväntat resultat efter körning av applikation

Nedan är ett screenshoot från en körning av referensapplikationen med tom databas

```
C:\buildroot\cypoint\MongoDbCodeTest\MongoDbCodeTest\bin\Debug\MongoDbCodeTest.exe
Welcome to the bank

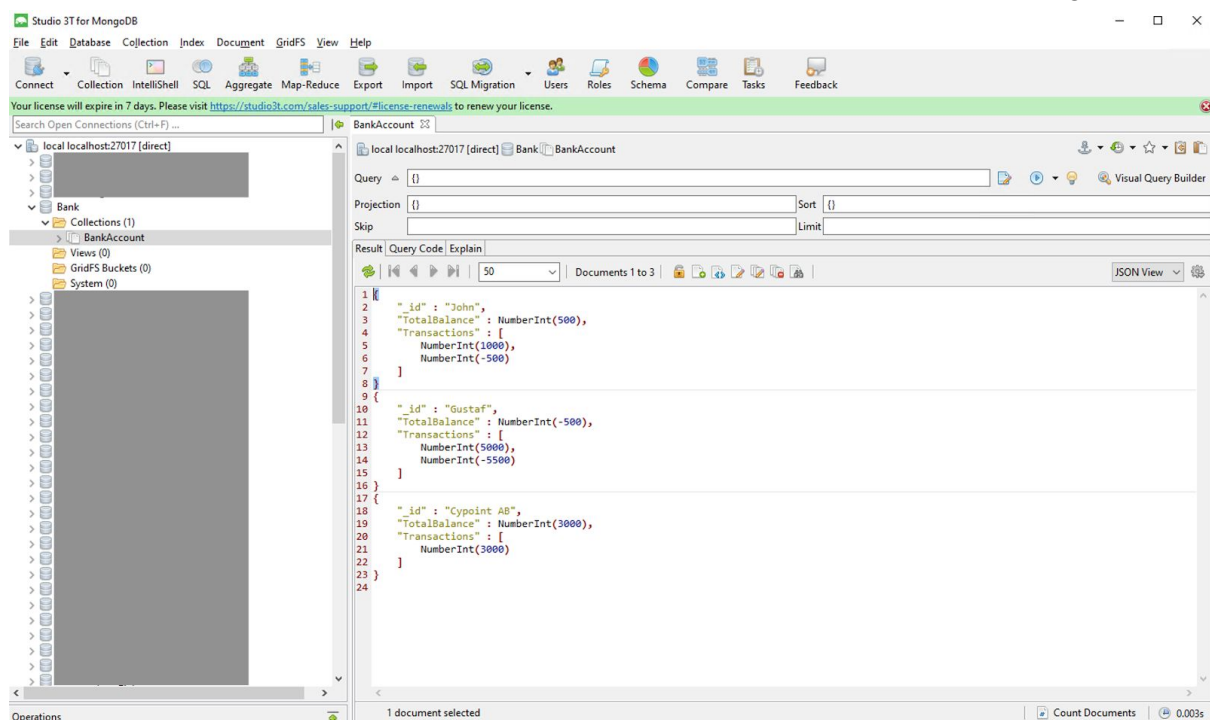
Please enter account name [empty to exit]: John
Please enter amount: test
Invalid amount, please try again
Please enter amount: 1000
Current balance: 1000
Please enter account name [empty to exit]: John
Please enter amount: -500
Current balance: 500
Please enter account name [empty to exit]: Gustaf
Please enter amount: 5000
Current balance: 5000
Please enter account name [empty to exit]: Cypoint AB
Please enter amount: 3000
Current balance: 3000
Please enter account name [empty to exit]: Gustaf
Please enter amount: -5500
Current balance: -500
Please enter account name [empty to exit]:

Bank total balance: 3000

Please press [ENTER] to exit
```

Efter körning skall det finnas en databas som heter “Bank”, och i databasen skall det finnas en collection som heter “BankAccount”.

Såhär skall innehåller i “BankAccount” collectionen se ut efter ovanstående körning



Förväntningar

- Resultatet skall vara en .ZIP-fil med källkod utan några binärer (.exe .dll etc) och skall kunna kompileras/köras på en windows maskin med Visual Studio 2017 (Om du endast har senare version så löser vi det)
- Utveckla applikation på ungefär samma sätt som du skulle gjort om en kund hade bett dig göra den. Exempelvis om du har några patterns du brukar följa eller andra kod principer är det bra att använda dessa OM det är applicerbart. Det är inte meningen att det skall bli någon Enterprise FizzBuzz :)
(<https://github.com/EnterpriseQualityCoding/FizzBuzzEnterpriseEdition>)

Extrauppgifter

Koduppgiften är ovanstående. Men om det skulle visa sig vara för enkelt finns det några extrauppgifter du kan välja att göra, men de är på intet sätt obligatoriska.

Extrauppgift A

Implementera huvuduppgiften med hjälp av valfri dependency injection container

Extrauppgift B

Använd mongodb's aggregation framework för att lösa någon del av huvuduppgiften.

Extrauppgift C

Skriv unittester som motionerar och verifierar koden mot databasen.