

Projektni zadatak - Konzistencija

Ognjen Čavić E2 161/2024

November 26, 2025

Sadržaj

1	Uvod	2
2	Organizacija rešenja	2
2.1	ConsistencyService	2
2.1.1	ConsistencyService/ConsistencyService.cs	2
2.1.2	ConsistencyService/ConsistencyService.svc.cs	2
2.2	ConsistencyManager	3
2.2.1	ConsistencyManager/Program.cs	3
2.3	Sensors	3
2.3.1	Sensors/Program.cs	3
2.3.2	Sensors/Sensor.cs	3
2.4	SensorsDatabaseContext	3
2.4.1	SensorsDatabaseContext/DbContext.cs	3
3	Rezultati	4
3.1	Pokretanje	4
3.2	Rezultati	4
3.3	Resavanje potencijalnog SQL izuzetaka	6

1 Uvod

Potrebno je napraviti sistem u kojem 10 senzora na nasumičnim vremenskim intervalima generišu nasumične vrednosti, dok "menadžer" vrši njihovo poravnanje, tj. bira jednu od tih vrednosti koja će biti proglašena glavnom. Ovaj sistem se može opisati kao skup od 3 glavne komponente:

- **Simulator** - Pokreće po jednu nit za svaki senzor. Svaka nit na vremenskom intervalu između 1 i 10 sekundi generiše novu vrednost i upisuje je u sopstvenu tabelu u bazi podataka pomoću *Entity Framework*-a.
- **Menadžer** - Na svakih 60 sekundi vrši poravnanje, tako što bira čitanje senzora koje je najpovoljnije i upisuje u tabele svih senzora. Kriterijum za izbor najbolje vrednosti je najskorije očitavanje u prihvatljivom opsegu $\pm 5^{\circ}C$. Ukoliko takvo ne postoji uzima se najskorije.
- **WCF servis** - Definiše način komunikacije između menadžera i senzora.

2 Organizacija rešenja

Rešenje se sastoji od 4 manja projekta, gde svaki predstavlja jednu od prethodno navedenih celina, sa tim da dodati četvrti projekat opisuje način struktuiranja podataka za upis u bazu podataka:

- **ConsistencyService** - WCF servis
- **ConsistencyManager** - menadžer koji vrši poravnanje
- **Sensors** - pokretanje niti koje simuliraju senzore
- **SensorsDatabaseContext** - organizacija baze podataka

2.1 ConsistencyService

2.1.1 ConsistencyService/ConsistencyService.cs

Klasa **TemperatureInfo** opisuje oblik podataka koji se razmenjuju između korisnika servisa i formiraju ga 4 polja:

- **sensor_id** - Identifikacioni broj senzora, broj od 1 do 10
- **temperature** - Vrednost temperature
- **timestamp** - vremenski podatak
- **source** - izvor podatka, tj. ko ga upisuje u bazu, senzor ili menadžer tokom poravnanja

U ovom fajlu su samo još definisani interfejsi servisa. **IPublisher** definiše metod za upis u bazu, dok interfejs **ISubscriber** definiše metod za slanje upita.

2.1.2 ConsistencyService/ConsistencyService.svc.cs

Implementacija **IPublisher** i **ISubscriber** interfejsa unutar **ConsistencyService** klase. Bitno je primetiti da klasa nasleđuje oba interfejsa. Metoda **publishTemp** upisuje zadati **TemperatureInfo**, što se koristi da i menadžer i senzori upisuju u bazu, tako što **source** polje postave na odgovarajuću vrednost. Poslednje očitavanje senzora se dobija pozivom **querySensor** i prosleđivanjem odgovarajućeg rednog broja. Obe ove funkcije vrše odgovarajuću transformaciju tipova tako da korisnici servisa mogu da pozivaju operacije razmene informacija sa bazom podataka bez poznavanja unutrašnje organizacije u istoj.

2.2 ConsistencyManager

2.2.1 ConsistencyManager/Program.cs

Sadrži dve funkcije, **Main** i **findBest**. **Main** funkcija na svakih 60 sekundi šalje upit bazi podataka za poslednji podatak iz svake tabele, potom pomoću funkcije **findBest** određuje koje od čitanja senzora je bilo najbolje. Nakon toga se izmenjuju polja **source** i **timestamp** tako da se označi da je podatak upisan kao rezultat poravnjanja i vreme kada je on upisan, takav podatak se upisuje u svaku od tabela. Potrebno je napomenuti da menadžer instancira *subscribera* i *publisher*, koje koristi za upite i upise u bazu, respektivno.

2.3 Sensors

2.3.1 Sensors/Program.cs

Svrha ovog fajla je da instancira 10 objekata klase **Sensors** i pokreće 10 niti u vidu taskova.

2.3.2 Sensors/Sensor.cs

Definicija klase **Sensor**, koja sadrži 5 ali praktično 4 polja.

- **sensorID** - redni broj senzora
- **_random** - statičko polje za generator nasumičnih brojeva
- **maxTimeDelay** - maksimalni vremenski razmak do generisanja sledeće vrednosti
- **random** - generator koji je u konstruktoru postavljen na statički generator
- **publisherClient** - klijent servisa koji će pisati u bazu podataka
- **_randomLock** - objekat koji se koristi za zaključavanje dela koda koji generiše nasumične brojeve.

Polje **random** je postavljeno da bude jednako statičkom **_random** zato što ako svaka instanca senzora ima svoj generator, desiće se to da postoji 10 generatora nasumičnih brojeva sa vrlo sličnim "semenom" tj. **seed-om**. To znači da će sekvence generisanja brojeva za sve senzore biti identične zato što se implicitno koristi seed koji zavisi od vremena inicijalizacije. Međutim, pošto objekat koji pravi nasumične brojeve može da pukne ukoliko ga više od jedne niti koristi u bilo kom trenutku, potrebno je zaključati deo koda koji poziva metode tog objekta.

2.4 SensorsDatabaseContext

2.4.1 SensorsDatabaseContext/DbContext.cs

Definicija kako su organizovani podaci unutar baze podataka. Klasa **DBInfo_Sensor** opisuje kako izgleda jedan red baze, odnosno kako se jedno očitavanje senzora predstavlja unutar baze i sadrži 4 polja:

- **id** - redni broj podatka unutar tabele, nije povezano sa rednim brojem senzora
- **temperature** - temperatura koju je senzor očitao
- **timestamp** - vreme očitavanja
- **source** - ko je upisao podatak, senzor ili menadžer

Pošto je potrebno imati odvojenu tabelu za svaki od senzora, a *Entity Framework* zahteva da svaka tabela ima posebno definisan tip, jednostavno je napravljeno dodatnih 10 klasa **DBInfo_Sensor1**, ..., **DBInfo_Sensor10** i svaka od njih nasleđuje **DBInfo_Sensor**. **SensorsDBContext** je izveden iz **DbContext** i sadrži 10 polja, gde svako od njih predstavlja po jednu od tabela unutar baze podataka.

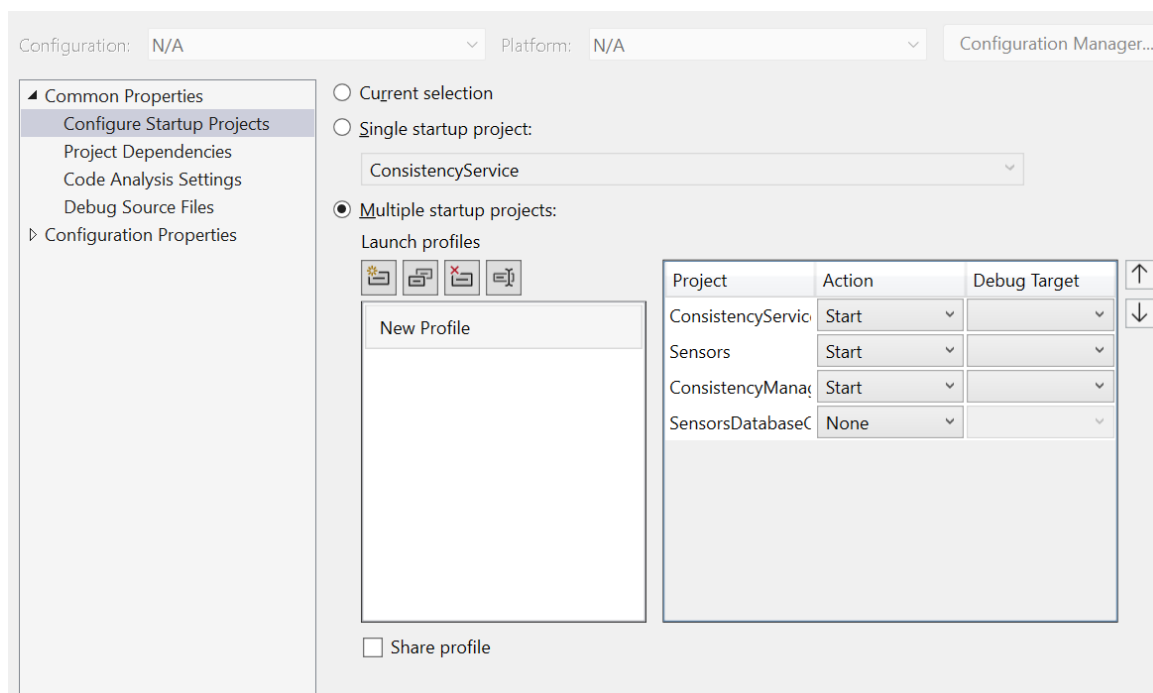
3 Rezultati

3.1 Pokretanje

Za pravilno funkcionisanje napisanog koda, potrebno je pokrenuti projekte (desnim klikom na projekat *Debug*→*Start new instance*) u sledećem redosledu:

1. **ConsistencyService** - WCF servis
2. **Sensors** - pokretanje niti koje simuliraju senzore
3. **ConsistencyManager** - menadžer koji vrši poravnanje

Jedna olakšica jeste to što visual studio ima opciju da se na start dugme podigne proizvoljan broj projekata u proizvoljnom redosledu. Desnim klikom na naziv celog rešenja tj. **solution-a**, i odabirom opcije properties otvara se prozor gde je potrebno sa leve strane odabrati **Configure Startup Projects** i konfigurisati ga kao na slici:



Slika 1: Podešavanje pokretanja projekta

3.2 Rezultati

Nakon pokretanja, otvaraju se dva prozora. Na jednom prozoru se ispisuju poruke o tome koji senzor je očitao koju vrednost, dok na drugom prozoru se ispisuju poruke menadžera nakon što izvrši poravnanje.

```

Sensor 6 measured 22.1683818344811
Sensor 10 measured 28.4593702938684
Sensor 5 measured 23.5745196387053
Sensor 7 measured 20.2676822199801
Sensor 8 measured 11.2017151067041
Sensor 3 measured 12.9976688432496
Sensor 1 measured 22.5696471298903
Sensor 8 measured 27.7514681954642
Sensor 3 measured 25.2829374350994
Sensor 9 measured 27.1514615589527
Sensor 3 measured 19.3743173169784
Sensor 2 measured 17.8054660222519
Sensor 1 measured 25.7824806197465
Sensor 4 measured 12.4102589871782
Sensor 3 measured 23.3265732477077
Sensor 4 measured 16.2928145035602
Sensor 6 measured 20.3477869882937
Sensor 10 measured 12.8899271240877
Sensor 7 measured 25.3505158402727
Sensor 5 measured 27.6224005490646
Sensor 2 measured 19.3369317750153
Sensor 3 measured 13.3263997097157
Sensor 9 measured 23.7991339963857
Sensor 4 measured 17.1065351772618
Sensor 8 measured 29.7522344066539
Sensor 2 measured 12.114325452124
Sensor 8 measured 13.410850383067
Sensor 1 measured 29.6705989724354
Sensor 6 measured 22.9491575495103
Sensor 7 measured 11.9147929371869
Sensor 3 measured 12.8192190932199
Sensor 2 measured 22.8913141847082
Sensor 6 measured 27.7143745951887
Sensor 10 measured 21.0751071903273
Sensor 4 measured 12.5695865706399
Sensor 5 measured 12.2599373116437
4: 19.1187566374982, 11/26/2025 2:54:29 PM
5: 12.997720634098, 11/26/2025 2:54:28 PM
6: 12.2593017491788, 11/26/2025 2:54:27 PM
7: 18.1795501933338, 11/26/2025 2:54:29 PM
8: 12.4528714280822, 11/26/2025 2:54:29 PM
9: 10.6936799086135, 11/26/2025 2:54:26 PM
10: 27.1981433300293, 11/26/2025 2:54:28 PM
AVG: 20.0664752336563
BEST: id:4, temp:19.1187566374982, time: 11/26/2025 2:54:29 PM

1: 17.6052542718152, 11/26/2025 2:55:28 PM
2: 22.5718619639854, 11/26/2025 2:55:28 PM
3: 24.4795754805578, 11/26/2025 2:55:26 PM
4: 26.2262246181379, 11/26/2025 2:55:27 PM
5: 20.2514016862267, 11/26/2025 2:55:28 PM
6: 28.6411654290935, 11/26/2025 2:55:22 PM
7: 17.7683124354893, 11/26/2025 2:55:23 PM
8: 25.3997985345311, 11/26/2025 2:55:27 PM
9: 20.9415282732535, 11/26/2025 2:55:25 PM
10: 29.9996578134595, 11/26/2025 2:55:27 PM
AVG: 25.3951255740206
BEST: id:5, temp:20.2514016862267, time: 11/26/2025 2:55:28 PM

1: 20.5937804796704, 11/26/2025 2:56:26 PM
2: 12.410544251283, 11/26/2025 2:56:29 PM
3: 11.7328808650993, 11/26/2025 2:56:29 PM
4: 10.6154552104955, 11/26/2025 2:56:28 PM
5: 22.4461423477373, 11/26/2025 2:56:27 PM
6: 24.8797854105382, 11/26/2025 2:56:25 PM
7: 27.0439278600011, 11/26/2025 2:56:25 PM
8: 11.9312981804513, 11/26/2025 2:56:28 PM
9: 13.6886377556662, 11/26/2025 2:56:29 PM
10: 28.6617742100087, 11/26/2025 2:56:28 PM
AVG: 20.9399352144972
BEST: id:9, temp:13.6886377556662, time: 11/26/2025 2:56:29 PM

```

Slika 2: Prozori sa porukama o merenjima senzora i izvršenim poravnanjima menadžera

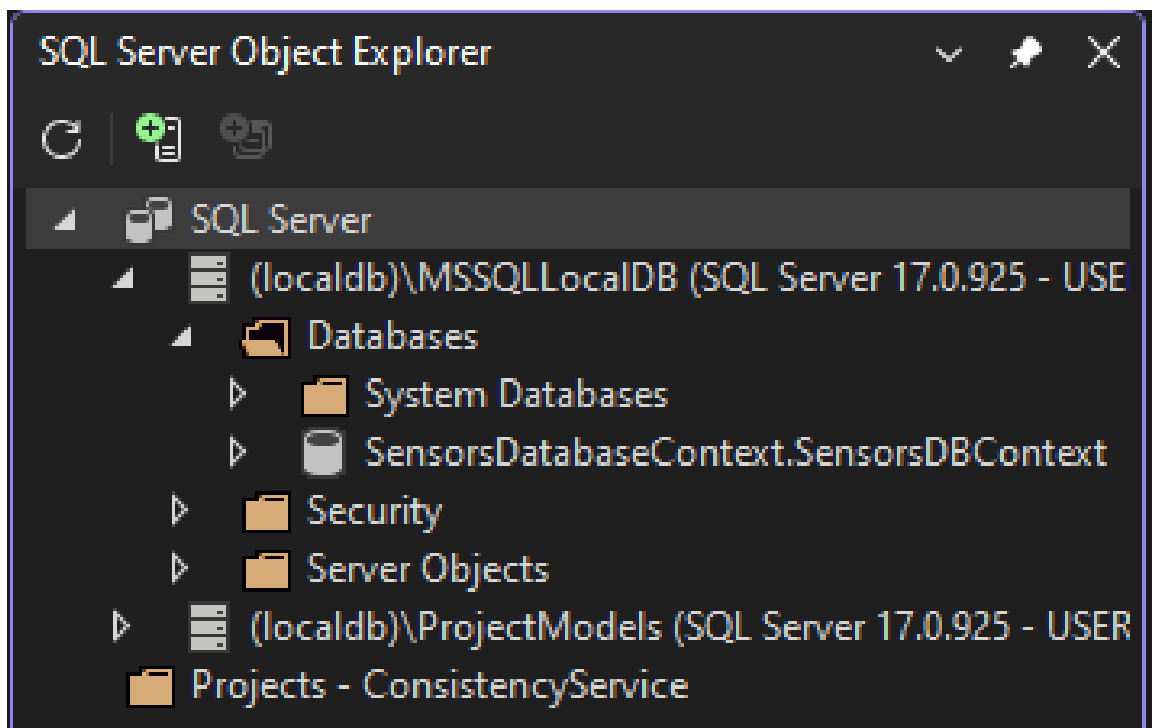
Druga stvar koju je potrebno proveriti je da li su podaci upisani u bazu podataka i da li su validni. Na slici 3 se može videti da se vremenski sve poklapa, tj. da senzor upisuje u svoju tabelu na nasumičnom intervalu između 1 i 10 sekundi, dok menadžer upisuje poravnatu vrednost svakog minuta.

id	temperature	timestamp	source
917	28.775242361601091	11/26/2025 2:55:05 PM	SENSOR
918	10.789626660193143	11/26/2025 2:55:10 PM	SENSOR
919	17.348495799744732	11/26/2025 2:55:17 PM	SENSOR
920	24.479575480557781	11/26/2025 2:55:26 PM	SENSOR
921	20.251401686226671	11/26/2025 2:55:30 PM	CONSISTENCY_MANAGER
922	26.927766668110976	11/26/2025 2:55:31 PM	SENSOR
923	19.140785582894825	11/26/2025 2:55:36 PM	SENSOR
924	29.770986661208319	11/26/2025 2:55:37 PM	SENSOR
925	11.25051907321928	11/26/2025 2:55:38 PM	SENSOR
926	21.664544023471189	11/26/2025 2:55:43 PM	SENSOR
927	27.239518471639379	11/26/2025 2:55:49 PM	SENSOR
928	12.4095964349851	11/26/2025 2:55:58 PM	SENSOR
929	11.961666272003979	11/26/2025 2:56:04 PM	SENSOR
930	12.382465844220699	11/26/2025 2:56:10 PM	SENSOR
931	10.753716435634399	11/26/2025 2:56:18 PM	SENSOR
932	29.635491203346984	11/26/2025 2:56:21 PM	SENSOR
933	11.732880865099318	11/26/2025 2:56:29 PM	SENSOR
934	13.688637755666225	11/26/2025 2:56:30 PM	CONSISTENCY_MANAGER
935	12.997668843249636	11/26/2025 2:56:34 PM	SENSOR
936	25.28293743509936	11/26/2025 2:56:35 PM	SENSOR
937	19.374317316978384	11/26/2025 2:56:36 PM	SENSOR
938	23.326573247707714	11/26/2025 2:56:38 PM	SENSOR
939	13.326399709715695	11/26/2025 2:56:42 PM	SENSOR
940	12.819219093219944	11/26/2025 2:56:48 PM	SENSOR
941	23.282713393346739	11/26/2025 2:56:56 PM	SENSOR
942	21.056140172787075	11/26/2025 2:57:01 PM	SENSOR
943	23.033525139574671	11/26/2025 2:57:10 PM	SENSOR
944	25.183242613069361	11/26/2025 2:57:19 PM	SENSOR
945	20.856724647272713	11/26/2025 2:57:23 PM	SENSOR
946	21.771100662542089	11/26/2025 2:57:24 PM	SENSOR
947	10.441536400672764	11/26/2025 2:57:25 PM	SENSOR
948	25.107364717455283	11/26/2025 2:57:29 PM	SENSOR
949	19.533700956745864	11/26/2025 2:57:30 PM	CONSISTENCY_MANAGER
950	13.413952432299942	11/26/2025 2:57:38 PM	SENSOR
951	14.404728638196703	11/26/2025 2:57:39 PM	SENSOR
939	20.043290448395204	11/26/2025 2:56:21 PM	SENSOR
940	24.879785410538215	11/26/2025 2:56:25 PM	SENSOR
941	13.688637755666225	11/26/2025 2:56:30 PM	CONSISTENCY_MANAGER
942	22.168381834481089	11/26/2025 2:56:32 PM	SENSOR
943	20.347786988293652	11/26/2025 2:56:39 PM	SENSOR
944	22.949157549510318	11/26/2025 2:56:46 PM	SENSOR
945	27.714374595188712	11/26/2025 2:56:48 PM	SENSOR
946	10.453214273067756	11/26/2025 2:56:53 PM	SENSOR
947	28.125458833819934	11/26/2025 2:56:58 PM	SENSOR
948	16.818245363802763	11/26/2025 2:57:04 PM	SENSOR
949	12.506214735333907	11/26/2025 2:57:06 PM	SENSOR
950	16.20075149750372	11/26/2025 2:57:14 PM	SENSOR
951	27.797117940055728	11/26/2025 2:57:21 PM	SENSOR
952	22.492602594426181	11/26/2025 2:57:26 PM	SENSOR
953	19.533700956745864	11/26/2025 2:57:29 PM	SENSOR
954	19.533700956745864	11/26/2025 2:57:30 PM	CONSISTENCY_MANAGER
955	11.477721445950555	11/26/2025 2:57:37 PM	SENSOR
956	24.987304059317012	11/26/2025 2:57:42 PM	SENSOR
957	14.574761439382453	11/26/2025 2:57:51 PM	SENSOR
958	22.505246117946342	11/26/2025 2:57:56 PM	SENSOR
959	13.641897683796426	11/26/2025 2:58:02 PM	SENSOR
960	10.120657337885657	11/26/2025 2:58:05 PM	SENSOR
961	27.429429049337948	11/26/2025 2:58:09 PM	SENSOR
962	17.4780338106109	11/26/2025 2:58:12 PM	SENSOR
963	17.740929707764149	11/26/2025 2:58:13 PM	SENSOR
964	12.597123935165406	11/26/2025 2:58:19 PM	SENSOR
965	14.046750107801868	11/26/2025 2:58:24 PM	SENSOR
966	21.631703102789679	11/26/2025 2:58:30 PM	CONSISTENCY_MANAGER
967	26.525755374005882	11/26/2025 2:58:31 PM	SENSOR
968	10.083240950518865	11/26/2025 2:58:33 PM	SENSOR
969	23.466373762798668	11/26/2025 2:58:34 PM	SENSOR
970	17.860063448483153	11/26/2025 2:58:38 PM	SENSOR
971	24.597771556395003	11/26/2025 2:58:41 PM	SENSOR
972	19.719384019132416	11/26/2025 2:58:44 PM	SENSOR
973	26.917857023383426	11/26/2025 2:58:49 PM	SENSOR

Slika 3: Prikaz redova u bazi podataka

3.3 Resavanje potencijalnog SQL izuzetaka

Prilikom prebacivanja koda na drugi računar moguće je da se pojavi izuzetak pri pokretanju projekta. Rešenje za taj problem jeste obrisati bazu podataka u **View**→ **SQL Server Object Explorer** i obrisati bazu koju ovaj projekat kreira **SensorsDatabaseContext.SensorsDBContext**.



Slika 4: Pregled baza u SQL Server Object Explorer