

# Višekorisnička chat aplikacija u C#-u

Antonio Kovačić

Janko Marohnić

Lana Arzon

26. lipnja 2014.

# Sadržaj

<b>1</b>	<b>Opis problema</b>	<b>2</b>
<b>2</b>	<b>Implementacija</b>	<b>2</b>
2.1	Server . . . . .	2
2.1.1	Kreiranje . . . . .	2
2.1.2	Instaliranje . . . . .	3
2.1.3	Pokretanje . . . . .	3
2.2	Klijent . . . . .	3

# 1 Opis problema

Potrebno je napraviti aplikaciju za gdje se ljudi mogu međusobno dopisivati u realnom vremenu. Dvije ili više osoba se treba moći "naći na jednom mjestu", i komunicirati zajedno putem tekstualnih poruka.

Kada se osoba prijavi u chat, treba upisati svoje korisničko ime, kojim će se ta osoba identificirati. Ako je postoji druga osoba u istoj "sobi" koja već ima željeno korisničko ime, osoba bi trebala izabrati drugo korisničko ime. Nakon što je upisala korisničko ime, osoba mora moći vidjeti koji su sve drugi korisnici trenutno prijavljeni u istu "sobu".

Kada osoba pošalje poruku u grupni chat (napiše tekst i stisne "enter"), svaka druga osoba mora primiti tu istu poruku (i treba pisati od koga dolazi). Na taj način svatko zna što su svi ostali rekli. I poruke moraju stizati redom kojim su poslane.

Kada se osoba odjavi iz chata, ostalim osobama treba doći poruka da se ta osoba odjavila iz chata.

## 2 Implementacija

### 2.1 Server

#### 2.1.1 Kreiranje

Windows servis je program koji zapravo radi u pozadini dok je korisnik prijavljen na korisnički račun unutar Windows operativnog sustava. Podešavanjem konfiguracija u **Command Promptu** Windows servis postaje aktivan. Samo kreiranje Windows servisa (detaljnim opisom što radi) se vrši kodiranjem u *C#*. Najprije kreiramo novi projekt u *C#*, odabirući opciju *Windows Service*. Zatim rekonstruiramo metodu *OnStart* - dodajemo kôd za otvaranje (instanciranje) kanala, registraciju objekata (dio tog koda *C#* doda automatski), itd. Zatim se instalacija vrši na jednostavan način preko Visual Studia, odnosno **Command Prompta**.

```
1
2     protected override void OnStart(string [] args) {
3         BinaryServerFormatterSinkProvider provider = new
4             BinaryServerFormatterSinkProvider();
5         provider.TypeFilterLevel = System.Runtime.
6             Serialization.Formatters.TypeFilterLevel.Full;
7         IDictionary props = new Hashtable(); props["port"]
8             = 12345;
9         TcpChannel channel = new TcpChannel(props, null,
            provider);
        ChannelServices.RegisterChannel(channel);
        RemotingConfiguration.RegisterWellKnownServiceType(
            typeof(ChatServer), "Chat.ChatServer",
            WellKnownObjectMode.Singleton);
    }
```

### 2.1.2 Instaliranje

Za postavljanje servisa smo napomenuli da ga moramo konfigurirati najprije preko **Command Prompta**. Nakon kompajliranja koda Windows servisa, dobivamo binarnu (izvršnu) datoteku **Chat.exe**. Najprije u **Command Promptu** trebamo otići u direktorij **.NET Framework v. 4** naredbom:

```
cd "C:\Windows\Microsoft.NET\Framework64\v4.0.30319"
```

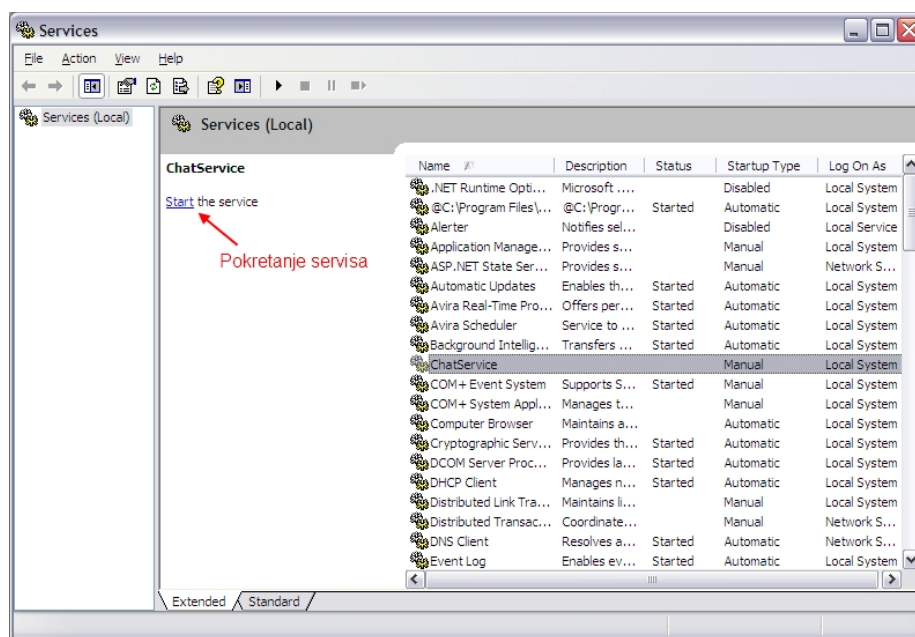
Potom naredbom:

```
installutil "C:\Ciljni_direktorij\Chat\bin\Debug\Chat.exe"
```

instaliramo dani windows servis.

### 2.1.3 Pokretanje

Pokretanje servisa se vrši na sljedeći način. Pokrenemo **Administrative tools** preko **Control Panela** (ili preko **Search programs and files** na startnoj liniji Windowsa. Zatim odaberemo opciju **Services** u kojoj nađemo **ChatService** (tako smo nazvali naš servis). Lijevim odabirom na opcijisu **Start** pokrećemo dani servis.



Slika 1: Pokretanje servisa ChatService

## 2.2 Klijent

...