



Upravljanje datotekama i stream- ovima u .NET-u

System.IO Namespace

System.IO Namespace sadrži tipove koji omogućavaju:

- Čitanje datoteka
- Pisanje u datoteke
- Stream-ove podataka
- Osnovnu podršku za datoteke i direktorijume.

Manipulacija datotekama

Tipovi za manipulaciju datotekama:

- File – sadrži statičke metode za kreiranje, kopiranje, brisanje, premeštanje i otvaranje datoteka. Pomaže u kreiranju FileStream objekata.
- Directory – sadrži statičke metode za kreiranje, premeštanje i prolazak kroz direktorijume i poddirektorijume.

Manipulacija datotekama

- FileInfo – Pruža informacije o datotekama. Omogućava kreiranje, kopiranje, brisanje, premeštanje i otvaranje datoteka. Podržava kreiranje FileStream objekata.
- DirectoryInfo – Pruža informacije o direktorijumima. Omogućava kreiranje, premeštanje i prolazak kroz direktorijume i poddirektorijume.

Čitanje iz datoteke

- `File.ReadAllText` – otvara tekstualnu datoteku i celokupan tekst iz nje učitava u string
- `File.ReadAllLines` - otvara tekstualnu datoteku i iz nje učitava sve redove u niz stringova.
- `File.ReadAllBytes` - otvara binarnu datoteku i iz nje učitava sadržaj u niz bajtova
- `File.ReadLines` – čita pojedinačne redove tekstualne datoteke i upisuje ih u niz stringova. Metod pogodan za velike tekstualne datoteke.

Upisivanje u datoteku

- `File.WriteAllText` - Kreira novu tekstualnu datoteku i upisuje u nju tekstualni sadržaj.
- `File.WriteAllLines` - Kreira novu datoteku i upisuje u nju jedan ili više stringova
- `File.WriteAllBytes` - Kreira novu binarnu datoteku i upisuje u nju prosleđeni niz bajtova

Tokovi podataka (Streams)

- Stream-ovi su nizovi bajtova koji se mogu koristiti za upisivanje i čitanje iz različitih vrsta skladišta podataka.

Prednosti stream-ova

- Inkrementalno procesiranje podataka
- Apstrakcija rada sa skladištima podataka
- Fleksibilnost
- ...

Stream klasa

- Apstraktna klasa Stream podržava čitanje i pisanje bajtova. Sve klase koje predstavljaju stream-ove nasleđuju klasu Stream.
- Klasa Stream i njene izvedene klase apstrahuju detalje operativnog sistema i hardware-a
- Klasa Stream implementira IDisposable interfejs. Pošto koristi eksterne resurse, stream se čisti garbage collector-om, te ga je potrebno ručno počistiti korišćenjem Dispose() metode.

FileStream klasa

Implementira stream za datoteku, podržavajući operacije čitanja i pisanja u datoteku i iz datoteke.

MemoryStream klasa

- Implementira stream za memoriju, podržavajući operacije čitanja i pisanja u memoriju i iz memorije.

Čuvanje datoteka u bazi podataka

- DBMS sistemi imaju mogućnost čuvanja datoteka u tabelama najčešće korišćenjem BLOB tipova.
- Niz bajtova (`byte[]`) može da se koristi za čuvanje fajlova u memoriji i za potrebe upisivanja u bazu podataka
- Memory Stream može lako da se konvertuje u `byte[]` (i obrnuto)

MemoryStream i WCF

- Parametri koji se prosleđuju kroz WCF i sadrže MemoryStream objekte treba i sami da implementiraju IDisposable interfejs
- Dispose metod treba da izvrši Dispose za sve stream-ove i ostale IDisposable objekte
- Implementacije ServiceContract metoda koje sadrže IDisposable parametre treba da imaju dodeljen parametar

[OperationBehavior(AutoDisposeParameters = true)]
(default je true)

Monitorisanje sistema datoteka

- Klasa FileSystemWatcher osluškuje notifikacije o promenama u sistemu datoteka i pokreće event-e kada se direktorijum ili datoteka u direktorijumu promene.
- Događaji za koje se osluškuju notifikacije:
 - Created
 - Changed
 - Renamed
 - Deleted
 - Error

Zadatak

- Razvija se aplikativni sistem za razmenu datoteka između serverske i klijentskih aplikacija
- Aplikativni sistem se sastoji od sledećih aplikacija:
 - ✓ Service – serverska aplikacija koja prima stream-ove i smešta ih u različite vrste skladišta (direktorijum, baza podataka...)
 - ✓ UploaderClient – klijentska aplikacija koja čita datoteke iz definisanog direktorijuma i prosleđuje ih Service aplikaciji putem WCF-a
 - ✓ DownloaderClient – klijentska aplikacija koja šalje serverskoj aplikaciji upit na osnovu kog serverska aplikacija vraća streamove. Na klijentskoj strani datoteke se upisuju u definisani direktorijum.

Zadatak

- UploaderClient ima mogućnost monitorisanja datoteka kroz osluškivanje događaja upisa u direktorijum, i kroz stalnu proveru postojanja datoteka.
- Service ima mogućnost čuvanja datoteka u in-memory bazi podataka i u file system-u

Zadatak

- Potrebno je implementirati metode za manipulaciju datotekama i stream-ovima. Metode koje je potrebno implementirati označene su sa „IMPLEMENTIRATI“
- Potrebno je uraditi izmene u kodu koje se tiču performansi i čišćenja memorije.



Izvori

<https://learn.microsoft.com/en-us/dotnet/api/system.io?view=net-7.0>



Hvala!

