

.NET 8

alati i veštine

Unapredite svoju karijeru tako što ćete savladati ključne .NET alate i veštine



.NET 8

alati i veštine

**Unapredite svoju karijeru tako što ćete
savladati ključne .NET alate i veštine**

Mark J. Prajs

Izdavač:



Obalskih radnika 4a
Beograd, Srbija

Tel: 011/2520272

e-pošta: kombib@gmail.com

veb-sajt: www.kombib.rs

Za izdavača:

Mihailo J. Šolajić, direktor

Autor:

Mark J. Prajs

Prevod: Nemanja Lukić

Recezent: Miroslav Ristić

Slog: Zvonko Aleksić

Znak Kompjuter biblioteke:

Miloš Milosavljević

Štampa: „Pekograf“, Zemun

Tiraž: 500

Godina izdanja: 2024.

Broj knjige: 580

Izdanje: Prvo

ISBN: 978-86-7310-6

Naslov originala:

Tools and Skills for .NET 8

ISBN 978-1-83763-520-7

Copyright © July 2024 Packt Publishing

Packt Publishing Ltd.

Birmingham, UK, packt.com

.NET 8
alati i veštine

Autorizovani prevod sa engleskog jezika.

Sva prava zadržana. Nijedan deo ove knjige se ne sme reprodukovati, čuvati u sistemu za pronalaženje ili prenositi u bilo kom obliku ili na bilo koji način, bez prethodne pismene dozvole izdavača, osim u slučaju kratkih citata ugrađenih u kritičke članke ili prikaze.

Tokom pripreme ove knjige uloženi su svi naponi da se obezbedi tačnost predstavljenih informacija. Međutim, informacije sadržane u ovoj knjizi se prodaju bez garancije, bilo izričite ili podrazumevane. Autori i izdavač neće biti odgovorni za bilo kakvu štetu prouzrokovanu ili navodno prouzrokovanu direktno ili indirektno ovom knjigom.

„Kompjuter biblioteka“ i „Packt Publishing“ su nastojali da obezbede informacije o zaštitnim znakovima o svim kompanijama i proizvodima pomenutim u ovoj knjizi korišćenjem odgovarajućeg načina njihovog pominjanja u tekstu. Međutim, ne možemo da garantujemo tačnost ovih informacija.

O AUTORU

Mark J. Prais je stručnjak kompanije Microsoft za C# programiranje i arhitekturu Microsoft Azure rešenja, sa više od 30 godina iskustva. Od 1993. godine, položio je više od 80 Microsoft programerskih ispita i specijalizovao se za pripremu kandidata za polaganje tih ispita. Između 2001. i 2003. godine, Mark je radio na pisanju zvaničnih kurseva za Microsoft u Redmondu, SAD. Njegov tim je pisao prve kurseve za C# dok je još uvek bio u ranoj, alfa verziji. Dok je radio za Microsoft, predavao je kurseve „obučavanje trenera” kako bi pripremio Microsoft Certified Trainers za C# i .NET. Mark je veći deo svoje karijere proveo obučavajući učenike svih generacija, od 16-godišnjih pripravnika do 70-godišnjih penzionera, s tim da su većina bili profesionalni programeri. Mark ima diplomu iz računarskih nauka sa izuzetnim rezultatima.

Hvala svim mojim čitaocima. Vaša podrška mi omogućava da pišem ove knjige i slavim vaše uspehe. Posebno sam zahvalan čitaocima koji mi daju korisne povratne informacije putem mog GitHub spremišta i imejla, kao i onima koji komuniciraju sa mnom i književnim zajednicama na Discord platformi. Vi pomažete da svaka moja naredna knjiga bude još bolja.

O RECENZENTIMA

Kiren Fut je samouki C# programer sa velikom strašću za učenjem novih tehnologija i tehnika. Neprestano istražuje najnovije trendove jezika C# i .NET SDK platforme, sa posebnim fokusom na veb programiranje. Kao vodeći softverski inženjer u ConnX Business Solutions, maloj softverskoj kompaniji sa sedištem u Velikoj Britaniji, ima priliku da primeni svoje znanje u praksi i pomaže drugima da steknu nove veštine. Uživa u pomaganju drugima na njihovom putu kroz C#/.NET i kao takav je aktivan član Packt Discord zajednice (zašto ne biste skenirali QR kod na sledećoj stranici i popričali s njim?).

Milan Jovanović je Microsoft MVP i radi u industriji više od sedam godina. Bio je softverski arhitekta u svojoj prethodnoj kompaniji. Danas je punopravni kreator sadržaja koji pomaže .NET programerima da unaprede svoje veštine arhitekture i dizajna softvera. Specijalizovao se za razvoj veb aplikacija i entuzijasta je dizajna vođenog domenom.

Milan je prethodno radio kao tehnički urednik na knjizi Čista aritektura sa .NET okruženjem (ISBN: 9780138203368).

Miroslav Ristić je redovni profesor na Prirodno-matematičkom fakultetu Univerziteta u Nišu, sa preko 25 godina iskustva u razvoju statističkog softvera. Posebno se ističe njegov rad na razvoju grafičkog korisničkog interfejsa R Commander za programski jezik R. Dugi niz godina recenzirao je značajan broj knjiga za izdavačku kuću Springer i časopis Journal of Applied Statistics. Od 2023. godine aktivno recenzira najaktuelnija izdanja izdavačke kuće "Kompjuter biblioteka". Nakon prevođenja, svako izdanje prolazi kroz njegovo stručno vrednovanje i recenziju prevoda, sa ciljem da se osigura da prevodi budu ne samo jasni, precizni i prilagođeni čitaocima, već i da održe visok kvalitet i stručnu relevantnost knjiga.

Predgovor

Postoje knjige o programiranju koje imaju za cilj da budu sveobuhvatni priručnici za pojedinačne teme kao što su bezbednost, ubrizgavanje zavisnosti, jedinično testiranje, veb testiranje, kontejnerizacija, razvoj u oblaku, projektni obrasci i socijalne veštine, uključujući pripremu za intervju.

Ova knjiga je drugačija. Ona je vodič, korak po korak, za učenje svih tih alata i veština za profesionalne .NET programere. Knjiga je sažeta i ima za cilj da bude brz i zabavan tekst prepun praktičnih uputstava za svaku od tema. Širina opšteg narativa ide na račun dubine, ali ćete naći mnoge putokaze za dalje istraživanje.

Iz mog iskustva, najteži deo kod učenja novog alata ili veštine je sam početak. Kada mi se objasne najvažniji ključni koncepti i kad uradim neke praktične zadatke, osećam se prijatno da samostalno istražujem zvaničnu dokumentaciju. Osetićete slobodu da eksperimentišete na svoju ruku nakon što vidite kako osnovne stvari ispravno funkcionišu.

Ova knjiga je najbolja za one koji već znaju osnove jezika C# i .NET biblioteka, koji su sami izradili neke aplikacije i usluge pomoću .NET okruženja, a koji sada žele da nauče veštine relevantne za timski rad u profesionalnoj organizaciji.

Pokriću najvažnije aspekte alata i veština za .NET profesionalce, da biste sa kolegama mogli da razgovarate o tim alatima i veštinama i brzo postanete produktivni.

Gde se nalaze kodovi rešenja

Postupna rešenja zadataka i vežbi možete preuzeti ili kopirati iz GitHub spremišta na sledećoj adresi: <https://github.com/markjprice/tools-skills-net8>.

Ako ne znate kako to da uradite uputstva se nalaze na kraju prvog poglavlja, *Predstavljanje .NET alata i veština*.

Šta ova knjiga pokriva

Ovaj predgovor ukratko predstavlja svako poglavlje. Duži opisi, kao i razlozi za pokrivanje svake od tema, nalaze se u prvom poglavlju.

Uvod

Poglavlje 1, Predstavljanje .NET alata i veština, bavi se podešavanjem vašeg razvojnog okruženja. Možete koristiti Visual Studio 2022, Visual Studio Code, JetBrains Rider ili bilo koji drugi uređivač koda. Takođe, pruža detaljniji uvid u to šta ostatak ove knjige pokriva i zašto. Kreirate neke biblioteke klasa za model entiteta i bazu podataka koje ćemo koristiti u projektima u narednim poglavljima. Poglavlje se završava uputstvom za upotrebu ove knjige sa .NET 9 okruženjem, koje očekujemo u novembru 2024.

Alati

Važni alati za profesionalne .NET programere obuhvataju uređivač koda, Git, alate za otklanjanje grešaka i alate za analizu memorije.

Poglavlje 2, Najbolja upotreba alata u uređivaču koda, bavi se ređe korišćenim alatima ugrađenim u Visual Studio, Code i Rider. Naučićete da prilagodite vaš uređivač i da kreirate isečke koda i funkcije za refaktorisanje.

Poglavlje 3, Upravljanje izvornim kodom pomoću sistema Git, pokriva najčešće zadatke koje obavljate sa Git sistemom da biste upravljali vašim izvornim kodom, posebno kada radite u timu .NET programera.

Poglavlje 4, Otklanjanje grešaka i rešavanje problema sa memorijom, govori o korišćenju alata za otklanjanja grešaka u vašem uređivaču koda i o upotrebi alata za praćenje korišćenja memorije da bi se poboljšale aplikacije i usluge.

Poglavlje 5, Upozorenja, metrike i prikupljanje zapisa radi praćenja, objašnjava kako instrumentirati kod kako bi se omogućila vidljivost tokom testiranja i proizvodnje, pomoću telemetrije.

Veštine

Važne veštine za profesionalne .NET programere obuhvataju dokumentovanje, dinamički kod, zaštitu podataka i koda i integraciju **veštačke inteligencije** u projekte.

Poglavlje 6, Dokumentovanje koda, API interfejsa i usluga, raspravlja o tome kako najbolje dokumentovati kod da bi drugi programeri mogli da ga održavaju u budućnosti oslanjajući se na komentare, i kako da dokumentujete vaše usluge i API interfejsa da bi ih drugi programeri koristili na ispravan način.

Poglavlje 7, Dinamičko praćenje i menjanje izvršavanja koda, predstavlja neke uobičajene tipove koji su uključeni u .NET za izvođenje refleksije koda i primenu i čitanje atributa; rad sa stablima izraza; i kreiranje generatora izvornog koda.

Poglavlje 8, Zaštita podataka i aplikacija pomoću kriptografije, govori o zaštiti vaših podataka od pregleda zlonamernih korisnika pomoću šifrovanja, i od manipulacije ili oštećenja putem heširanja i potpisivanja. Takođe ćete učiti o autentifikaciji i autorizaciji za zaštitu aplikacija od neovlašćenih korisnika.

Poglavlje 9, Izgradnja prilagođene usluge konverzacije zasnovane na velikim jezičkim modelima, pokriva izgradnju prilagođene usluge konverzacije koja integriše veštačku inteligenciju zasnovanu na **velikom jezičkom modelu**.

Testiranje

Važni alati i veštine za testiranje za profesionalne .NET programere obuhvataju zavisnosti usluga i sve vrste testiranja, od jediničnog testiranja do testiranja veb korisničkog interfejsa.

Poglavlje 10, Ubrizgavanje zavisnosti, kontejneri i trajanje usluge, govori o ukidanju krutih veza između komponenti, što je posebno važno za praktično testiranje. To vam takođe omogućava da bolje upravljate promenama i složenošću softvera.

Poglavlje 11, Jedinično testiranje i lažiranje, predstavlja prakse testiranja koje će poboljšati kvalitet vašeg koda. Pri jediničnom testiranju je lako napraviti grešku, što podriva poverenje tima. Kada se uradi ispravno, uštedeće vreme i novac i olakšati proces razvoja.

Poglavlje 12, Testovi integracije i bezbednosti, predstavlja dva viša nivoa testiranja, integraciono i bezbednosno, koja se primenjuju na sve komponente rešenja.

Poglavlje 13, Procene performansi, testiranje opterećenja i otpornosti pri ekstremnim uslovima, predstavlja vam pravilnu upotrebu biblioteke BenchmarkDotNet za praćenje vašeg koda radi merenja performansi i efikasnosti. Zatim ćete videti kako da obavite testiranje opterećenja i izdržljivosti na vašim projektima da biste predvideli potrebne resurse i procenili troškove implementacije u proizvodnji.

Poglavlje 14, Funkcionalno i testiranje s kraja na kraj veb stranica i usluga, predstavlja funkcionalno i testiranje s kraja na kraj usluga API interfejsa i veb korisničkih interfejsa.

Poglavlje 15, Docker kontejnerizacija, predstavlja koncept kontejnerizacije i konkretnu upotrebu Docker platforme za virtualizaciju domaćina usluga u rešenjima složenih arhitektura.

Poglavlje 16, Razvoj u oblaku pomoću skupa komponenti .NET Aspire, predstavlja .NET Aspire, prethodno definisan način upravljanja simuliranim razvojnim okruženjem u oblaku na vašem lokalnom računaru.

Dizajn i karijera

Veštine dizajna potrebne profesionalnom .NET programeru obuhvataju uobičajene projektne obrasce kodiranja, implementacije algoritama i arhitektonske šablone. Cilj je da ostvarite karijeru kojoj težite. Da biste to postigli, potrebni su vam impresivni dokumenti za prijavu na posao, kao i nastup na intervjuu.

Poglavlje 17, Obrasci i principi dizajna, predstavlja SOLID projektne obrasce, kao i druge uobičajene projektne obrasce koji se koriste u .NET okruženju, kao što su Unikat i Fabrika.

Poglavlje 18, Osnove arhitekture softvera i rešenja, predstavlja arhitekturu softvera i rešenja.

Poglavlje 19, Vaša karijera, timski rad i intervjui, predstavlja rad u profesionalnom timu, prijavljivanje za poslove i postupak intervjuisanja za poziciju .NET softverskog inženjera i srodne poslove. Ovo poglavlje sadrži 60 najčešće postavljanih pitanja na intervjuima. Prva tri pitanja imaju predložene odgovore u štampanoj knjizi. Kompletan skup od 60 pitanja i predložene odgovora dostupan je u PDF formatu koji možete preuzeti sa sledeće adrese: <https://github.com/markjprice/tools-skills-net8/blob/main/docs/interview-qa/readme.md>.

Epilog predstavlja opcije za dalje učenje, da biste savladali alate i veštine potrebne da postanete svestran profesionalni .NET programer.

Dodatak, Odgovori na pitanja za proveru znanja, sadrži odgovore na pitanja za testiranje znanja sa kraja svakog poglavlja.

Dodatak se nalazi na sledećoj adresi: <https://packt.link/isUsj>.

Šta vam je potrebno za ovu knjigu

Možete razvijati i implementirati .NET projekte pomoću razvojnog okruženja Visual Studio 2022, uređivača koda Visual Studio Code ili pomoću nekog alata treće strane, kao što je JetBrains Rider. Code, Rider i alati komandne linije rade na većini operativnih sistema, uključujući Windows, macOS i mnoge varijante Linuxa. Visual Studio je dostupan samo za Windows, jer je Visual Studio za Mac računare povučen iz upotrebe, zvanično ne podržava .NET 8, a avgust 2024. godine označava kraj njegove upotrebe.

Preuzimanje slika u boji iz ove knjige

Takođe, na raspolaganju vam je PDF datoteka koja sadrži kolorne slike snimaka ekrana i dijagrama korišćenih u ovoj knjizi. Slike u boji će vam pomoći da bolje razumete promene izlaznih rezultata.

Možete preuzeti ovu datoteku na adresi: [//packt.link/gbp/9781837635207](https://packt.link/gbp/9781837635207).

Konvencije

U ovoj knjizi korišćene su razne tekstualne konvencije.

KoDuTekstu: Ukazuje na delove koda u tekstu, imena tabela baza podataka, imena direktorijuma, imena datoteka, ekstenzije datoteka, putanje, lažne URL adrese, korisnički unos i Twitter (X) naloge. Na primer: „Direktorijumi `Controllers`, `Models` i `Views` sadrže ASP.NET Core klase i `.cshtml` datoteke za izvršavanje na serveru.”

Blok koda je postavljen na sledeći način:

```
// Skladištenje stavki na indeks pozicijama.  
names[0] = "Kejt";  
names[1] = "Džek";  
names[2] = "Rebeka";  
names[3] = "Tom";
```

Kada želimo da skrenemo vašu pažnju na određeni deo bloka koda, relevantne linije ili stavke su istaknute:

```
// Skladištenje stavki na indeks pozicijama.  
names[0] = "Kejt";  
names[1] = "Džek";  
names[2] = "Rebeka";  
names[3] = "Tom";
```

Svi unosi i izlazi iz komandne linije napisani su na sledeći način:

```
dotnet new console
```

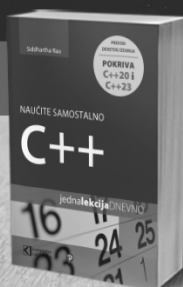
Podobljana slova: Označavaju novi **termin**, važnu **reč** ili reči koje vidite na ekranu, na primer, u menijima ili dijalozima. Na primer: „Klikom na dugme **Next** prelazite na sledeći ekran.”

Dobra praksa

ili

Važne napomene

Prikazani su ovako.



Postanite član Kompjuter biblioteke

Kupovinom jedne naše knjige stekli ste pravo da postanete član Kompjuter biblioteke. Kao član možete da kupujete knjige u pretplati sa 40% popusta i učestvujete u akcijama kada ostvarujete popuste na sva naša izdanja. Potrebno je samo da se prijavite preko formulara na našem sajtu.

Link za prijavu: kombib.rs/kblista.php

Skenirajte QR kod
registrujte knjigu
i osvojite nagradu



1

Predstavljanje .NET alata i veština

U ovom prvom poglavlju, cilj je da shvatite koncept alata i veština koje ćete naučiti u ovoj knjizi, da podesite vaše razvojno okruženje za Visual Studio 2022, Visual Studio Code ili JetBrains Rider, a zatim da podesite bazu podataka i projekte koje ćemo koristiti u ostalim poglavljima. U ovoj knjizi ću koristiti nazive **Visual Studio**, **Code** i **Rider** za ova tri uređivača koda.

Koristim termin **moderan .NET** za .NET 8 i prethodne verzije, kao što je .NET 6, proizašle iz .NET Core okruženja. Termin **zastareli .NET** koristim za .NET Framework, Mono, Xamarin i .NET Standard. Moderan .NET je ujedinjenje tih zastarelih platformi i standarda.

Svako poglavlje u ovoj knjizi predstavlja određene alate i veštine, ali su neka poglavlja više fokusirana na određeni alat ili određenu veštinu.

Ovo poglavlje pokriva sledeće teme:

- Uvod u ovu knjigu i njen sadržaj
- Podešavanje vašeg razvojnog okruženja
- Efikasna upotreba GitHub spremišta za ovu knjigu
- Gde potražiti pomoć
- Podešavanje baze podataka i projekata za ovu knjigu
- Upotreba .NET 9 uz ovu knjigu

Napomena

GitHub spremište za ovu knjigu sadrži rešenja za kompletne projekte aplikacija za sve zadatke sa kodom:

<https://github.com/markjprice/tools-skills-net8/>

Kad odete na GitHub spremište, jednostavno pritisnite taster . (tačka) na vašoj tastaturi, ili promenite .com u .dev da biste pretvorili spremište u uređivač koda u realnom vremenu zasnovan na uređivaču Code, pomoću GitHub Codespaces. Uređivač Code u veb pregledaču odlično će raditi paralelno sa vašim izabranim uređivačem koda, dok budete izvršavali zadatke kodiranja iz knjige. Možete porediti vaš kod sa kodom rešenja i lako kopirati i umetati delove ako bude potrebno.

Uvod u ovu knjigu i njen sadržaj

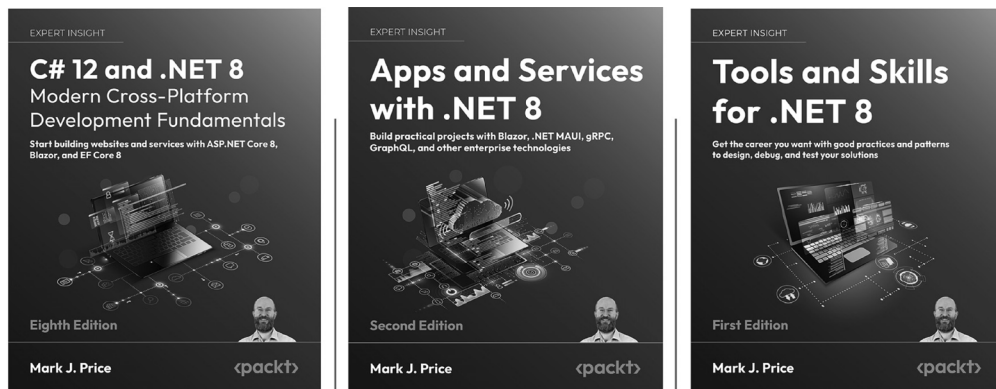
Pre nego što pređemo na pregled ove knjige, postavićemo kontekst činjenicom da je ovo jedna od tri knjige o .NET 8 koje sam napisao i koje pokrivaju gotovo sve što početnik treba da zna o .NET okruženju.

Prateće knjige za upotpunjavanje vašeg procesa učenja

Ova knjiga je treća od tri knjige u trilogiji koja upotpunjava vaše učenje okruženja .NET 8:

1. Prva knjiga, *C# 12 i .NET 8 – Osnove modernog razvoja za više platformi*, predstavlja osnove C# jezika, .NET biblioteka i upotrebu okruženja ASP.NET Core i Blazor za razvoj veb aplikacija. Koncipirana je tako da se čita po redu jer se veštine i znanje iz svih poglavlja nadograđuju i potrebne su za razumevanje ostatka knjige.
2. Druga knjiga, *Aplikacije i usluge sa .NET 8*, predstavlja specijalizovane .NET biblioteke, kao što je internacionalizacija, i popularne pakete trećih strana, uključujući Serilog i Noda Time. Učite da gradite mašinske, **unapred kompajlirane (AOT)** usluge sa ASP.NET Core Minimal APIs i da poboljšate performanse, skalabilnost i pouzdanost pomoću keširanja, redova i pozadinskih usluga. Implementirajte više usluga pomoću alata GraphQL, gRPC, SignalR i Azure Functions. Na kraju, učite da gradite grafičke korisničke interfejsa za veb sajtove, desktop i mobilne aplikacije alatima Blazor i .NET MAUI.
3. Ova, treća knjiga, predstavlja važne alate i veštine koje profesionalni .NET programer treba da ima. To obuhvata projektne obrasce i arhitekturu rešenja, otklanjanje grešaka, analizu memorije, sve važne tipove testiranja, bilo jedinično, integraciono, testiranje performansi ili veb korisničkog interfejsa, a zatim teme testiranja rešenja u oblaku na vašem lokalnom računaru, kao što su kontejnerizacija, Docker i .NET Aspire. Na kraju dolazi priprema za intervju, da biste ostvarili karijeru .NET programera koju želite.

Rezime trilogije .NET 8 i najvažnije teme prikazani su na slici 1.1:



- **C# language**, including new C# 12 features, object-oriented programming, debugging, and unit testing.
- **.NET libraries**, including numbers, text, regular expressions, collections, file I/O, and data with EF Core and SQLite.
- **Websites and web services** with ASP.NET Core and Blazor.

- **More libraries**: Internationalization, multitasking, and third-party packages.
- **More data**: SQL Server and Cosmos DB.
- **More services**: Minimal APIs, caching, queuing, GraphQL, gRPC, SignalR, and Azure Functions.
- **More user interfaces**: ASP.NET Core MVC, Blazor, and .NET MAUI.

- **Tools**: IDEs, debugging, memory analysis, and AI assistants.
- **Tests**: Unit, integration, performance, security, and web, including DI and IoC.
- **Develop**: Docker and .NET Aspire.
- **Design**: Patterns, principles, software and solution architecture.
- **Career**: Teamwork and interviews.

Slika 1.1: Prateće knjige za učenje .NET 8 za čitaoce početnog i srednjeg nivoa

Pružamo vam PDF datoteku koja sadrži kolorne slike snimaka ekrana i dijagrama korišćenih u ovoj knjizi. Možete preuzeti ovu datoteku na adresi <https://packt.link/gbp/9781837635207>.

Ciljna grupa za ovu knjigu

Ova knjiga ima dve ciljne grupe:

- Čitaoce koji su pročitali moju knjigu o osnovama jezika C#, .NET biblioteka i upotrebe ASP.NET Core okruženja za veb razvoj, *C# 12 i .NET 8 – Osnove modernog razvoja za više platformi*, a sada žele da nadgrade svoje znanje.
- Čitaoce koji poseduju osnovne veštine i znanje o C# i .NET i žele da steknu praktične veštine i da upoznaju uobičajene alate da bi dostigli nivo profesionalnog .NET programera, a posebno one koji žele da se pridruže timu .NET programera.

Pogledajmo jednu analogiju:

- Prvo, kuvar amater može da kupi knjigu i nauči osnovne veštine, koncepte i terminologiju koje svaki kuvar treba da zna da bi pravio uobičajena jela.
- Drugo, kuvar amater, takođe, može da kupi knjigu sa receptima da nauči da primeni stečeno znanje i veštine za pripremu kompletnih obroka.
- Treće, da bi postao profesionalac, kuvar bi trebalo da razume uloge u profesionalnoj kuhinji, da upozna specijalizovane alate i veštine potrebne pri kuvanju obroka za više ljudi u profesionalnom okruženju i rad u timu kuvara.

Ova tri scenarija su razlog zašto sam napisao tri knjige o .NET okruženju.

Predgovor ukratko predstavlja svako poglavlje, ali mnogi čitaoci preskaču predgovor. Zato, hajde sada da vidimo zašto je svaka od tema detaljno obrađena.

Alati

Postoji mnogo alata koje profesionalni .NET programer treba da poznaje. Neki su ugrađeni u većinu uređivača koda, kao alat za otklanjanje grešaka ili integracija sa kontrolom izvornog koda, a neki zahtevaju posebne aplikacije i usluge, kao analiza memorije i telemetrije.

Napomena

Čak i programeri početnici poznaju osnovne alate uređivača koda, kao što su glavni prozor za uređivanje, upravljanje datotekama u projektu, kako postaviti tačku prekida i započeti otklanjanje grešaka, zatim pregledati kod iskaz po iskaz, i kako pokrenuti projekat, pa te teme neće biti pokriveno u ovoj knjizi.

Poglavlje 2: Najbolja upotreba alata u uređivaču koda, govori o ređe korišćenim alatima ugrađenim u Visual Studio, Code i Rider. Glavni alati, kao što je alat za otklanjanje grešaka ili alat za analizu memorije obrađeni su u narednim, zasebnim poglavljima. Ovo poglavlje pokriva teme poput refaktorisanja i prilagođavanja uređivača koda pomoću standarda kao što je `.editorconfig`.

Poglavlje 3: Upravljanje izvornim kodom pomoću sistema Git, pokriva najčešće zadatke koje biste obavljali pomoću sistema Git za upravljanje vašim izvornim kodom, posebno kada radite u timu .NET programera. Git je distribuirani sistem kontrole izvornog koda, tako da programeri imaju lokalnu kopiju celog spremišta. To omogućava rad bez interneta, brzo grananje i spajanje. Git je najpopularniji sistem kontrole izvornog koda za .NET projekte, a postoje alati i proširenja dostupna za besprekornu integraciju sa svim uređivačima koda i alatima komandne linije. GitHub je popularna Microsoft platforma za skladištenje Git spremišta i saradnju na softverskim projektima.

Poglavlje 4: Otklanjanje grešaka i rešavanje problema sa memorijom, govori o korišćenju alata za otklanjanje grešaka u vašem uređivaču koda. Naučićete da koristite ugrađene funkcije za otklanjanje grešaka, da ukasite svoj kod atributima da bi bilo lakše videti šta se dešava tokom otklanjanja grešaka i da u vašem uređivaču koda koristite alate za praćenje upotrebe memorije da biste poboljšali svoje aplikacije i usluge.

Poglavlje 5: Upozorenja, metrike i prikupljanje zapisa radi praćenja, govori o tome kako da instrumentišete svoj kod da omogućite praćenje, metrike i kreiranje zapisa tokom proizvodnje, i kako da implementirate telemetriju pomoću OpenTelemetry platforme da biste pratili izvršavanje koda i omogućili vidljivost.

Veštine

Pored poznavanja upotrebe alata, profesionalni .NET programer treba da ima veštine kao što su dokumentovanje koda, iskorišćavanje dinamičkog koda i implementacija kriptografskih tehnika za zaštitu koda i podataka. Takođe ćete napraviti prilagođen četbot koji koristi model poboljšan prilagođenim funkcijama.

Poglavlje 6: Dokumentovanje koda, API interfejsa i usluga, objašnjava kako najbolje dokumentovati svoj kod da bi drugi programeri mogli da ga održavaju u budućnosti oslanjajući se na komentare, i kako dokumentovati svoje usluge i API interfejse da bi drugi programeri mogli da ih koriste kako je predviđeno. Prilikom dokumentovanja koda često ćete primetiti gde biste mogli da poboljšate dizajn i refaktorisaćete svoj kod i API interfejse.

Poglavlje 7: Dinamičko praćenje i menjanje izvršavanja koda, predstavlja neke uobičajene tipove za refleksiju koda uključene u .NET, primenu i čitanje atributa, rad sa stabilima izraza i, što je najkorisnije, kreiranje generatora izvornog koda.

Poglavlje 8: Zaštita podataka i aplikacija pomoću kriptografije, govori o zaštiti vaših podataka od pregleda zlonamernih korisnika, pomoću šifrovanja, i od manipulacije ili oštećenja pomoću heširanja i potpisivanja. Takođe ćete naučiti da koristite autentifikaciju i autorizaciju za zaštitu aplikacija od neovlašćenih korisnika.

Poglavlje 9: Izgradnja prilagođene usluge konverzacije zasnovane na velikim jezičkim modelima, pokriva izgradnju prilagođene usluge za konverzaciju koja integriše veliki jezički model (LLM) i prilagođene funkcije zasnovane na poslovnim podacima.

Poglavlje 10: Ubrizgavanje zavisnosti, kontejneri i trajanje usluge, govori o smanjenju čvrste povezanosti između komponenti, što je posebno važno za praktično testiranje. To takođe omogućava bolje upravljanje promenama i složenosti softvera.

Testiranje

Jedan od najvažnijih alata i veština za .NET programera je testiranje radi obezbeđivanja kvaliteta. Testiranje pokriva ceo ciklus razvoja, od testiranja malih jedinica ponašanja do testiranja s kraja na kraj korisničkog iskustva i integracije sistema. Što ranije u procesu razvoja testirate, to je manji trošak ispravljanja grešaka koje otkrijete.

Poglavlje 11: Jedinično testiranje i lažiranje, uvodi vas u prakse testiranja koje će poboljšati kvalitet vašeg koda. Jedinično testiranje je lako pogrešno primeniti i može postati beskorisno, što narušava poverenje tima. Ako ga pravilno sprovedete, uštedecete vreme i novac i olakšacete proces razvoja.

Poglavlje 12: Testovi integracije i bezbednosti, predstavlja viši nivo testiranja svih komponenti rešenja i vrste testiranja potrebne za održavanje bezbednosti vaših projekata.

Poglavlje 13: Procene performansi, testiranje opterećenja i otpornosti, objašnjava kako se pravilno koristiti biblioteka BenchmarkDotNet za praćenje vašeg koda radi merenja performansi i efikasnosti. Zatim ćete videti kako da izvršite testiranje opterećenja i izdržljivosti na svojim projektima da biste predvideli potrebne resurse i procenili troškove implementacije u proizvodnji pomoću alata Bombardier i radnog okvira NBomber.

Poglavlje 14: Funkcionalno i testiranje s kraja na kraj veb stranica i usluga, predstavlja vam testiranje API interfejsa i veb korisničkih interfejsa pomoću Playwright alata za automatizaciju.

Poglavlje 15: Docker kontejnerizacija, uvodi koncept kontejnerizacije i posebno upotrebu platforme Docker za virtualizaciju domaćina za usluge u složenim arhitekturama rešenja.

Poglavlje 16: Razvoj u oblaku pomoću skupa komponenti .NET Aspire, predstavlja vam .NET Aspire, prirodno definisan način upravljanja okruženjem za razvoj u oblaku. Automatski raspoređuje lokalne implementacije tokom razvoja i uključuje integracije sa alatima o kojima ste ranije učili u knjizi, kao što su OpenTelemetry za vidljivost, Docker kontejneri za mikrousluge, i tako dalje.

Dizajn i razvoj karijere

Završni deo ove knjige bavi se više teoretskim konceptima kao što su projektni obrasci, principi i arhitektura softvera i rešenja, a završava se poglavljem koje zaokružuje sve pripremom za intervju, da biste ostvarili karijeru koju želite, u timu programera i drugih profesionalaca.

Poglavlje 17: Obrasci i principi dizajna, predstavlja vam SOLID projektne obrasce, kao i druge uobičajene obrasce i principe, kao što su **DRY (Ne ponavljaj se)**, **KISS (Zarži jednostavnost, glupane)**, **YAGNI (Neće ti trebati)** i **PoLA (Princip najmanjeg iznenađenja)**.

Poglavlje 18: Osnove arhitekture softvera i rešenja pokriva arhitekturu softvera i rešenja, uključujući upotrebu alata Mermaid za kreiranje dijagrama arhitekture.

Poglavlje 19: Vaša karijera, timski rad i intervjui, predstavlja šta vam je potrebno da biste ostvarili karijeru koju želite. Da biste to postigli, morate impresionirati svojim rezimeom i na intervjuu. Ovo poglavlje pokriva često postavljana pitanja na intervjuima i predložene odgovore koji se poklapaju sa vašim iskustvom, da bi vam pomogli da date prave odgovore.

Podešavanje vašeg razvojnog okruženja

Pre nego što počnete sa programiranjem, biće vam potreban uređivač koda za C#. Microsoft ima porodicu uređivača koda i **integriranih razvojnih okruženja (IDE)**, koja obuhvata:

- **Visual Studio**: Visual Studio 2022 za Windows. (Visual Studio 2022 za Mac prestao je da radi 31. avgusta 2024. godine i ne preporučujemo ga.)
- **Code**: Visual Studio Code za Windows, Mac, Linux, putem veb pregledača ili GitHub Codespaces.
- **Rider**: JetBrains Rider, koji je dostupan za Windows, Mac ili Linux, ali je potrebno platiti licencu. Rider je popularan među iskusnijim .NET programerima.

Izbor odgovarajućeg alata i tipa aplikacije za učenje

Koji je najbolji alat i tip aplikacije za učenje upotrebe alata i veština sa C# i .NET?

Želim da slobodno izaberete bilo koji uređivač koda za C# ili IDE za zadatke kodiranja u ovoj knjizi, uključujući Visual Studio, Code, Rider ili neki uređivač koda za koji nikad nisam čuo.

U ovoj knjizi dajem opšta uputstva koja važe za sve alate, tako da možete da koristite bilo koji alat koji vam odgovara.

Upotreba razvojnog okruženja Visual Studio za opšti razvoj

Visual Studio može da kreira većinu vrsta aplikacija, uključujući konzolne aplikacije, veb sajtove, veb usluge, aplikacije za desktop i mobilne uređaje. Visual Studio radi samo na Windows 10 verziji 1909 ili novijoj, ili Windows Server 2016 ili novijem, i to samo na 64-bitnim verzijama. Verzija 17.4 je prva verzija koja podržava izvorni ARM64.

Upozorenje!

Visual Studio za Mac zvanično ne podržava .NET 8 i biće povučen iz upotrebe u avgustu 2024. godine. Ako ste koristili Visual Studio za Mac, trebalo bi da pređete na Code za Mac ili Rider za Mac, ili da koristite Visual Studio na virtualnoj mašini na vašem lokalnom računaru ili u oblaku pomoću tehnologije kao što je Microsoft Dev Box. Najavu povlačenja možete pročitati ovde: <https://devblogs.microsoft.com/visualstudio/visual-studio-for-mac-retirement-announcement/>.

Upotreba uređivača Code za razvoj aplikacija za više platformi

Najmoderniji i najlakši uređivač koda koji možete izabrati, i jedini koji je proizvod kompanije Microsoft i koji je nezavisan od platforme, jeste Code. Može se pokretati na svim uobičajenim operativnim sistemima, uključujući Windows, macOS i mnoge varijante operativnog sistema Linux, uključujući **Red Hat Enterprise Linux (RHEL)** i Ubuntu. Code je dobar izbor za moderan razvoj na više platformi jer ima opsežan skup dodataka koji se svakim danom povećava, za podršku mnogim jezicima, pored C#.

Pošto je nezavisan od platforme i lak, može se instalirati na svim platformama na koje će vaše aplikacije biti implementirane, za brze ispravke grešaka i slično. Izborom uređivača Code, programer može da koristi uređivač koda koji radi na više platformi za razvoj aplikacija koje su takođe nezavisne od platforme.

Code je daleko najpopularniji IDE, koji je izabralo više od 74% profesionalnih programera u anketi koju je kompanija Stack Overflow sproveda 2023. godine, a koju možete naći na sledećoj adresi: <https://survey.stackoverflow.co/2023/#most-popular-technologies-new-collab-tools-prof>.

Upotreba razvojnog okruženja GitHub Codespaces za razvoj u oblaku

GitHub Codespaces je potpuno konfigurisano razvojno okruženje zasnovano na uređivaču Code koje se može pokrenuti u okruženju skladištenom u oblaku i kojem se može pristupiti preko bilo kog veb pregledača. Podržava Git spremišta, dodatke i ugrađen interfejs komandne linije tako da je moguće uređivati, pokretati i testirati sa bilo kog uređaja. Pošto se pokreće u oblaku i u interakciju sa njim stupate putem veb pregledača, nema potrebe za preuzimanjem i instalacijom, a pošto je zasnovan na uređivaču Code, na vašem lokalnom računaru radi na isti način kao Code.

Dodatne informacije

Dodatne informacije o GitHub Codespaces možete naći na sledećoj adresi: <https://github.com/features/codespaces>. Još jedan način skladištenja razvojnog okruženja je Microsoft Dev Box: <https://azure.microsoft.com/en-us/products/dev-box>.

Recenzije za GitHub Codespaces dostupne su na sledećim adresama:

- <https://dev.to/github/github-codespaces-ga-any-good-reviewed-and-tested-3e62>
- https://medium.com/@pooyan_razian/github-codespaces-a-different-way-to-code-da455777f9a

Upotreba uređivača Rider za razvoj aplikacija za više platformi

Rider je uređivač koda treće strane, kompanije JetBrains, tvorca IntelliJ IDEA, vodećeg integrisanog razvojnog okruženja za jezike Java i Kotlin i dodatka ReSharper, popularnog Visual Studio dodatka za .NET programere. Rider radi na više platformi: Windows, macOS i Linux.

Licenca se plaća, ali ga koriste uglavnom profesionalni programeri koji cene njegove dodatne funkcije, kao što su inspekcije koda u realnom vremenu, kontekstualne akcije i refaktorisanja. Korišćenje uređivača Rider ukazuje na naprednog programera.

Za očekivati je da će nove verzije uređivača Rider podržavati najnoviju verziju .NET okruženja nekoliko nedelja nakon okruženja Visual Studio. Na primer, .NET 8 je objavljen 14. novembra 2023, a Visual Studio sa zvaničnom podrškom za .NET 8 je objavljen istog dana. Međutim, Rider verzija 2023.3 sa zvaničnom podrškom za .NET 8 je objavljena 7. decembra 2023.

Iako je JetBrains fantastična kompanija sa odličnim proizvodima, i Rider i ReSharper dodatak za Visual Studio su softver, a sav softver karakterišu greške i neobično ponašanje. Na primer, moguće je da će prikazivati grešku kao što je „Cannot resolve symbol“ u vašem radnom okviru Razor Pages, Razor pogledima i Blazor komponentama. Ipak, moguće je izgraditi i pokrenuti te datoteke jer, zapravo, problem ne postoji.

Ako koristite Rider i instalirali ste dodatke Unity Support ili Heap Allocation Viewer, tada će se često žaliti na transformacije u objekte. Uobičajen scenario kada se događa transformacija u objekat je kada se tipovi vrednosti kao što su `int` i `DateTime` prosleđuju kao pozicioni argumenti u formatiranju niski. To je problem za Unity projekte jer koriste drugačiji memorijski sakupljač smeća nego normalno .NET izvršno okruženje. Za ne-Unity projekte, kao što su svi projekti u ovoj knjizi, možete zanemariti ova upozorenja o transformaciji u objekte, jer nisu relevantna. Više o ovom problemu specifičnom za Unity možete pročitati na sledećoj adresi: <https://docs.unity3d.com/Manual/performance-garbage-collection-best-practices.html#boxing>.

Šta sam ja koristio

Da napišem i testiram kod za ovu knjigu, koristio sam sledeći hardver i softver:

- Visual Studio na:
 - Windows 11 na HP Spectre (Intel) laptopu
- Code na:
 - macOS na Apple Silicon Mac mini (M1) desktop računaru
 - Windows 11 na HP Spectre (Intel) laptopu
- Rider na:
 - Windows 11 na HP Spectre (Intel) laptopu
 - macOS na Apple Silicon Mac mini (M1) desktop računaru

Nadam se da i vi imate pristup raznovrsnom hardveru i softveru, jer uviđanje razlika na različitim platformama produbljuje vaše razumevanje izazova razvoja, iako je bilo koja od prethodnih kombinacija dovoljna da naučite da gradite praktične aplikacije i veb sajtove.

Implementacija na više platformi

Vaš izbor uređivača koda i operativnog sistema za razvoj ne ograničava mogućnost implementiranja vašeg koda.

.NET 8 podržava sledeće platforme za implementaciju:

- **Windows:** Windows 10 verzija 1607, ili novija. Windows 11 verzija 22000, ili novija. Windows Server 2012 R2 SP1, ili noviji. Nano Server verzija 1809, ili novija.
- **Mac:** macOS Catalina verzija 10.15, ili novija i Rosetta 2 x64 emulator. Mac Catalyst: 11.0, ili noviji.
- **Linux:** Alpine Linux 3.17, ili noviji. Debian 11, ili noviji. Fedora 37, ili novija. openSUSE 15, ili noviji. Oracle Linux 8, ili noviji. Red Hat Enterprise Linux (RHEL) 8, ili noviji. SUSE Enterprise Linux 12 SP2, ili noviji. Ubuntu 20.04, ili noviji.
- **Android:** API 21, ili noviji.
- **iOS i tvOS:** 11.0, ili noviji.

Napomena

Najnovije podržane operativne sisteme i verzije možete pregledati na sledećoj adresi:

<https://github.com/dotnet/core/blob/main/release-notes/8.0/supported-os.md>

Preuzimanje i instaliranje razvojnog okruženja Visual Studio

Mnogi profesionalni .NET programeri koriste Visual Studio u svom svakodnevnom radu. Čak i ako izaberete da koristite Code za zadatke kodiranja u ovoj knjizi, možda bi trebalo da upoznate i Visual Studio.

Ako nemate računar sa instaliranim Windows operativnim sistemom, preskočite ovaj deo i nastavite od sledećeg, gde ćete preuzeti i instalirati Code na macOS ili Linux.

Od oktobra 2014. godine, Microsoft je profesionalno izdanje razvojnog okruženja Visual Studio učinio besplatno dostupnim za studente, programere otvorenog koda i za pojedince. Zove se Community Edition. Bilo koje od izdanja je pogodno za ovu knjigu. Ako ga još niste instalirali, hajde da to sada uradimo:

1. Preuzmite Visual Studio verziju 17.10, ili noviju, sa sledeće adrese: <https://visualstudio.microsoft.com/downloads/>.

Napomena

Morate instalirati verziju 17.10, ili noviju, za novo Copilot iskustvo.

2. Pokrenite instalacioni program.

3. Na kartici **Workloads**, izaberite sledeće:
 - **ASP.NET and Web development**
 - **.NET desktop development** (zato što obuhvata konzolne aplikacije).
 - **Desktop development with C++** sa svim podrazumevanim komponentama (zato što omogućava objavljivanje konzolnih aplikacija i veb usluga koje se brže pokreću i imaju manji uticaj na memoriju).
4. Na kartici **Individual components**, u odeljku **Code tools**, izaberite sledeće:
 - **GitHub Copilot**
 - **Git for Windows**
5. Kliknite na **Install** i sačekajte da instalacioni program preuzme i instalira odabrani softver.
6. Kada je instalacija završena, kliknite na **Launch**.
7. Prvi put kada pokrenete Visual Studio, bićete upitani da se prijavite. Koristite Microsoft nalog, ako ga imate. Ako ga nemate, registrujte se na sledećoj adresi: <https://signup.live.com/>.
8. Prvi put kada pokrenete Visual Studio, bićete upitani da konfigurišete svoje okruženje. Za **Development Settings**, izaberite **Visual C#**. Za boju teme, ja sam izabrao plavu, ali možete izabrati šta god vam se dopada.
9. Ako želite da prilagodite prečice na tastaturi, idite na **Tools | Options...**, a zatim izaberite opciju **Environment | Keyboard**.

Prečice na tastaturi za Visual Studio

U ovoj knjizi ću izbegavati prikazivanje prečica na tastaturi jer su one često prilagođene. Pokušaću da prikažem one koje su iste na svim uređivačima koda i često u upotrebi.

Ako želite da identifikujete i prilagodite prečice na tastaturi, možete to uraditi na sledećoj adresi: <https://learn.microsoft.com/en-us/visualstudio/ide/identifying-and-customizing-keyboard-shortcuts-in-visual-studio>.

Alati u Visual Studio Enterprise izdanju

Izdanje Visual Studio Enterprise je dizajnirano za veće timove i organizacije sa naprednim potrebama za razvojem, testiranjem i implementacijom. Sadrži ekskluzivne funkcionalnosti prilagođene za projekte velikog opsega i napredne razvojne scenarije. Licence za Enterprise izdanje koštaju hiljade dolara i obično su ograničene na ljude koji rade za velike organizacije. Stoga, te funkcije i alate ne pokrивam u ovoj knjizi.

Neki od ekskluzivnih alata izdanja Visual Studio Enterprise prikazani su na sledećem spisku:

- **IntelliTrace** omogućava programerima da snimaju i ponovo reprodukuju izvršavanje aplikacija, poboljšavajući proces otklanjanja grešaka pružanjem istorijskog konteksta o tome šta se desilo pre nego što se problem pojavio.
- **Code Map** vizuelizuje i istražuje složene kodne baze, olakšavajući razumevanje i navigaciju odnosa i zavisnosti unutar koda.

- **Live Unit Testing** automatski pokreće pogođene jedinice testiranja u pozadini i u realnom vremenu prikazuje rezultate i pokrivenost koda, tako da pruža trenutne povratne informacije o promenama.
- **Microsoft Fakes** podržava kreiranje objekata adaptera i umetaka, što omogućava testiranje koda u izolaciji zamenom delova aplikacije lažnim objektima.
- **Web load and performance testing** omogućavaju programerima kreiranje, upravljanje i izvršavanje testova performansi za procenu skalabilnosti i odzivnosti veb aplikacija pod velikim opterećenjem.
- **Coded UI testing** podržava kreiranje automatizovanih testova za korisnički interfejs aplikacija, osiguravajući da se korisnički interfejs ponaša prema očekivanjima.
- **CodeLens** poboljšava razumevanje koda pružanjem direktnog uvida unutar uređivača, kao što su istorija promena, reference koda i detalji radnih stavki, bez potrebe da se napušta kod.
- **Architecture and dependency validation** pomaže da se osigura da naponi razvoja prate unapred definisane zavisnosti arhitekture i standarde slojeva, što sprečava arhitektonska odstupanja.
- **Advanced static code analysis** nudi sveobuhvatnija pravila i mogućnosti za analizu kvaliteta koda i potencijalnih problema, izvan onoga što je dostupno u drugim izdanjima.
- **Release management** pruža napredne alate i funkcije upravljanja izdavanjem za automatizaciju procesa implementacije, što pomaže timovima da brže i pouzdanije isporučuju softver.
- **Snapshot debugger** omogućava programerima da prave snimke svojih proizvodnih okruženja, omogućavajući im da pregledaju stanje i otklone probleme bez uticaja na pokrenuto okruženje.
- **App Center Test** omogućava automatizovano testiranje korisničkog interfejsa na hiljadama stvarnih uređaja u oblaku.

Preuzimanje i instaliranje uređivača Code

Code je značajno napredovao tokom proteklih nekoliko godina i prijatno je iznenadio Microsoft svojom popularnošću. Ako ste hrabri i volite da budete na samoj ivici tehnologije, postoji **Insiders** izdanje, koje je dnevna izgradnja sledeće verzije.

Čak i ako planirate da koristite samo Visual Studio, preporučujem da preuzmete i instalirate Code i probate ga za zadatke kodiranja u ovom poglavlju, a zatim odlučite da li želite da koristite samo Visual Studio za ostatak knjige.

Hajde sada da preuzmemo i instaliramo Code, .NET SDK, i dodatak C# Dev Kit:

1. Preuzmite i instalirajte ili Stable Build ili Insiders izdanje uređivača Code sa sledeće adrese: <https://code.visualstudio.com/>.

Dodatne informacije

Ako vam treba više pomoći da instalirate Code na bilo koji operativni sistem, možete pročitati zvanični vodič za podešavanje na sledećoj adresi: <https://code.visualstudio.com/docs/setup/setup-overview>.

- Preuzmite i instalirajte .NET SDK za verziju 8.0 sa sledeće adrese:
<https://www.microsoft.com/net/download>.

Dobra praksa

Najnovija SDK verzija u trenutku pisanja, u junu 2024. godine, je 8.0.301. Uvek instalirajte ili ažurirajte na najnoviju verziju sa zakrpama.

- Da biste instalirali dodatak **C# Dev Kit** pomoću korisničkog interfejsa, prvo morate pokrenuti Code aplikaciju.
- U uređivaču Code, kliknite na ikonicu **Extensions**, ili idite na **View / Extensions**.
- C# Dev Kit** je jedan od najpopularnijih dostupnih dodataka, tako da bi trebalo da se nalazi na vrhu spiska, ili u okvir za pretragu unesite upit **C# Dev Kit**.

Napomena

C# Dev Kit zavisi od **C#** dodatka verzije 2.0 ili novije, tako da ne morate posebno instalirati **C#** dodatak. Imajte na umu da **C#** dodatak verzije 2.0 ili novije više ne koristi OmniSharp jer ima novu komponentu koja koristi **protokol za jezičke usluge (LSP)**. **C# Dev Kit** takođe zavisi od dodataka **.NET Install Tool for Extension Authors** i **IntelliCode for C# Dev Kit**, tako da će i oni biti instalirani.

- Kliknite na **Install** i sačekajte da se preuzmu i instaliraju potrebni paketi.

Dobra praksa

Obavezno pročitajte licencni ugovor za **C# Dev Kit**. Ima restriktivniju licencu nego **C#** dodatak: <https://aka.ms/vs/csdevkit/license>.

Instaliranje drugih dodataka

U narednim poglavljima ove knjige koristićete još više dodataka za Code. Ako želite da ih instalirate odmah, svi dodaci koje ćemo koristiti prikazani su u *tabeli 1.1*:

NAZIV DODATKA I IDENTIFIKATOR	OPIS
C# Dev Kit ms-dotnettools.csdevkit	Službeni C# dodatak kompanije Microsoft. Upravljajte svojim kodom sa pregledačem rešenja i testirajte svoj kod uz integrirano otkrivanje i izvršavanje jediničnih testova. Obuhvata C# i IntelliCode for C# Dev Kit dodatke.
C# ms-dotnettools.csharp	Pruža podršku za uređivanje C# koda, uključujući isticanje sintakse, IntelliSense, Go To Definition, Find All References, podršku za ispravljanje grešaka za .NET i podršku za csproj projekte za Windows, macOS i Linux.
IntelliCode for C# Dev Kit ms-dotnettools. vscodeintellicode-csharp	Pruža funkcije za razvoj uz pomoć veštačke inteligencije za Python, TypeScript/JavaScript, C# i Java programere.

MSBuild project tools tintoy.msbuild-project-tools	Pružajući IntelliSense za MSBuild projektne datoteke, uključujući automatsko dovršavanje za elemente kao što je <PackageReference>.
SQL Server (mssql) for Visual Studio Code ms-mssql.mssql	Namenjen je za SQL Server, Azure SQL Database i SQL Data Warehouse programiranje bilo gde, sa bogatim skupom funkcionalnosti.
REST Client humao.rest-client	Omogućava slanje HTTP zahteva i pregled odgovora direktno u uređivaču Code.
ilspy-vscode icsharpcode.ilspy-vscode	Omogućava dekompajliranje Microsoft posredničkog jezika (MSIL) , poznatog kao .NET assembler. Podržava moderan .NET, .NET Framework, .NET Core i .NET Standard.

Tabela 1.1: Dodaci za Code korišćeni u ovoj knjizi

Upravljanje dodacima uređivača Code putem komandne linije

Možete da upravljate Code dodacima putem komandne linije ili terminala, kao što je prikazano u tabeli 1.2:

KOMANDA	OPIS
code --list-extensions	Prikazuje instalirane dodatke.
code --install-extension <extension-id>	Instalira navedeni dodatak.
code --uninstall-extension <extension-id>	Deinstalira navedeni dodatak.

Tabela 1.2: Rad sa dodacima putem komandne linije

Na primer, da biste instalirali dodatak **C# Dev Kit**, unesite sledeće u komandnu liniju:

```
code --install-extension ms-dotnettools.csdevkit
```

Napomena

Napravio sam PowerShell skriptove za instalaciju i deinstalaciju Code dodataka prikazanih u prethodnoj tabeli. Možete ih naći na sledećoj adresi: <https://github.com/markjprice/tools-skills-net8/tree/main/scripts/extension-scripts>.

Koncept verzija uređivača Code

Microsoft objavljuje novu verziju uređivača Code (skoro) svakog meseca, a verzije sa ispravkama grešaka još češće. Na primer:

- Verzija 1.90.0, izdanje sa funkcijama za maj 2024.
- Verzija 1.90.1, izdanje sa ispravkama grešaka za maj 2024.

Verzija uređivača Code je manje važna od verzije C# dodatka. Iako C# dodatak nije obavezan, on pruža IntelliSense dok kucate, navigaciju kroz kod i funkcije za ispravljanje grešaka, pa je veoma korisno instalirati ga i redovno ažurirati kao podršku za najnovije funkcije C# jezika.

Prečice na tastaturi za Code

U ovoj knjizi ću izbegavati prikazivanje prečica na tastaturi za zadatke poput kreiranja nove datoteke, jer su one uglavnom različite na različitim operativnim sistemima. Prikazivaću prečice samo u situacijama kada je potrebno više puta pritisnuti isti taster, na primer, tokom otklanjanja grešaka. Ove prečice su, takođe, doslednije na svim operativnim sistemima.

Ako želite da prilagodite prečice na tastaturi za Code, to možete učiniti kao što je prikazano na sledećoj adresi: <https://code.visualstudio.com/docs/getstarted/keybindings>.

Preporučujem da preuzmete PDF dokument sa prečicama na tastaturi za vaš operativni sistem sa sledećeg spiska:

- **Windows:** <https://code.visualstudio.com/shortcuts/keyboard-shortcuts-windows.pdf>
- **macOS:** <https://code.visualstudio.com/shortcuts/keyboard-shortcuts-macos.pdf>
- **Linux:** <https://code.visualstudio.com/shortcuts/keyboard-shortcuts-linux.pdf>

Preuzimanje i instaliranje alata Rider

Prema anketi koju je sproveo Stack Overflow 2023. godine, oko 20% profesionalnih C# i .NET programera koristi Rider u svakodnevnom radu. Rider je kompatibilan sa više platformi, tako da ga možete instalirati i pokrenuti na operativnim sistemima Windows i macOS, kao i na različitim Linux distribucijama. Možete koristiti evaluacioni licencni ključ za besplatni probni period od 30 dana.

Kompanija JetBrains takođe nudi besplatan **Toolbox App** za jednostavno upravljanje vašim integrisanim razvojnim okruženjima. Ovo je posebno korisno za Linux korisnike, jer se Rider ne ažurira putem apt (Advanced Package Tool) komande kao većina Linux programa. Takođe vam omogućava pristup programima za rani pristup softveru kao što je JetBrains Fleet, uređivač koda nove generacije.

Ako još uvek niste instalirali Rider i želite da ga koristite, hajde da to sada uradimo:

1. Preuzmite najnoviju verziju alata Rider na sledećoj adresi: <https://www.jetbrains.com/rider/download/>.

Napomena

Ako koristite Ubuntu 16.04 ili noviji, možete instalirati Rider iz komandne linije ili terminala pomoću sledeće komande: `sudo snap install rider --classic`

2. Pokrenite instalacioni program.
3. Preuzmite najnoviju JetBrains Toolbox App verziju sa sledeće adrese: <https://www.jetbrains.com/toolbox-app/>.
4. Pokrenite instalacioni program.
5. Idite na adresu <https://visualstudio.microsoft.com/downloads/#build-tools-for-visual-studio-2022> i instalirajte **Build Tools for Visual Studio**. Ovi alati su neophodni za korišćenje mašinskog AOT kompajlera.

Napomena

Ako ste instalirali JetBrains ReSharper u Visual Studio okruženju, imaćete iste alate kao u alatu Rider.

Ostali alati kompanije JetBrains

JetBrains nudi niz alata dizajniranih za poboljšanje produktivnosti, kvaliteta koda i performansi aplikacija za .NET programere. Kratak pregled alata dotPeek, dotTrace, dotMemory i dotCover, koji ističe njihove mogućnosti i prednosti, nalazi se na sledećem spisku:

- **dotPeek:** Ovaj alat može da dekompajlira kompajlirane .NET datoteke nazad u čitljiv C# kod, što olakšava istraživanje i razumevanje eksternog koda koji nema izvorni kod. Omogućava pregled sadržaja kompajliranih datoteka, pomažući programerima da istraže strukturu i zavisnosti bez potrebe za izvornim kodom. Pruža napredne mogućnosti navigacije i pretrage unutar dekompajliranog koda, kao što je brz prelazak na definicije simbola i upotrebe. Neprocenljiv je za razumevanje unutrašnjeg funkcionisanja biblioteka trećih strana, radnih okvira ili bilo kog kompajliranog .NET koda bez izvornog koda. Može da služi kao server za simbole, omogućavajući ulazak u dekompajlirani kod tokom sesija otklanjanja grešaka, pomažući u pronalaženju problema sa spoljnim bibliotekama. dotPeek je besplatan alat, što napredno dekompajliranje i istraživanje kompajliranih datoteka čini dostupnim za sve .NET programere.
- **dotTrace:** Ovaj alat pruža detaljno profilisanje performansi .NET aplikacija, identifikujući uska grla performansi merenjem vremena izvršenja i učestanosti poziva. Podržava uzorkovanje, praćenje i prikaze vremenskih linija, svaki pogodan za različite vrste analize performansi. Nudi sveobuhvatan skup pregleda i filtera za analizu podataka o profilisanju, uključujući stabla poziva, ključne tačke i vremenske prikaze. Pomaže programerima da optimizuju performanse aplikacija tako što precizno određuju sporiji kod i neefikasne algoritme.
- **dotMemory:** Ovaj alat analizira upotrebu memorije .NET aplikacija, identifikujući curenje memorije i probleme povezane sa **sakupljanjem otpada (GC)**. Pomaže u razumevanju dodele i oslobađanja memorije, ukazivanjem na prekomerne dodele koje bi mogle dovesti do degradacije performansi. Omogućava poređenje snimaka memorije uzetih u različitim trenucima, što olakšava identifikaciju curenja memorije i odgovornih objekata. Olakšava optimizaciju memorije uvidom u obrasce potrošnje memorije i otkrivanje curenja memorije.
- **dotCover:** Ovaj alat meri koliko je kodne baze pokriveno testovima, ističući pokrivene i nepokrivene blokove koda. Radi sa različitim radnim okvirima za testiranje i podržava sve .NET aplikacije, poboljšavajući efikasnost testiranja. Pruža režim neprekidnog testiranja koji automatski pokreće testove u pozadini kako se kod menja, nudeći trenutne povratne informacije o pokrivenosti i uspešnosti testa. Pomaže u poboljšanju kvaliteta koda osiguravanjem veće pokrivenosti testovima, što može dovesti do manje grešaka i stabilnijih aplikacija. Olakšava efikasan proces testiranja identifikovanjem netestiranog koda, što omogućava programerima da se fokusiraju na oblasti koje nemaju pokrivenost testovima.

Zajedno, ovi JetBrains alati čine sveobuhvatan paket alata koji može značajno da unapredi produktivnost i efikasnost .NET razvojnih timova.

Chrome alati za veštačku inteligenciju

Chrome sada ima ugrađene alate za veštačku inteligenciju o kojima možete čitati na sledećoj adresi: <https://developer.chrome.com/docs/devtools/console/understand-messages>.

Efikasna upotreba GitHub spremišta za ovu knjigu

Git je sistem za upravljanje izvornim kodom koji se često koristi. GitHub je kompanija, veb sajt i desktop aplikacija koja olakšava upravljanje Git sistemom. Microsoft je kupio GitHub 2018. godine, tako da će i dalje biti blisko integrisan sa Microsoft alatima.

Kreirao sam GitHub spremište za ovu knjigu i koristim ga za sledeće svrhe:

- Za skladištenje kodova rešenja iz knjige, koji će se održavati i nakon datuma štampanja
- Za pružanje dodatnih materijala koji proširuju knjigu, kao što su ispravke grešaka, mala poboljšanja, liste korisnih linkova i duži članci koji ne mogu da stanu u štampanu knjigu
- Za omogućavanje čitaocima da me kontaktiraju ukoliko imaju problema sa knjigom

Postavljanje pitanja u vezi sa knjigom

Ako naiđete na probleme u praćenju bilo kog uputstva u ovoj knjizi ili primetite grešku u tekstu ili kodu u rešenjima, molim vas da prijavite problem u GitHub spremištu:

1. Koristite svoj omiljeni veb pregledač da odete na sledeći link: <https://github.com/markjprice/tools-skills-net8/issues>.
2. Kliknite na **New Issue**.
3. Unesite što više detalja koji će mi pomoći da pronađem problem. Na primer:
 - Naslov specifičnog odeljka, broj stranice i broj koraka
 - Vaš uređivač koda, na primer, Visual Studio, Code, Rider ili neki drugi, uključujući broj verzije
 - Što više vašeg koda i konfiguracije koji smatrate relevantnim i potrebnim
 - Opis očekivanog ponašanja i ponašanja koje ste iskusili
 - Snimci ekrana (možete prevući i otpustiti slike u okvir za unos problema)

Sledeće informacije su manje relevantne, ali mogu biti korisne:

- Vaš operativni sistem, na primer, Windows 11 64-bit ili macOS Big Sur verzija 11.2.3
- Vaš hardver, na primer, Intel, Apple Silicon ili ARM CPU

Želim da svi moji čitaoci imaju uspeha sa mojom knjigom, tako da ako mogu da pomognem vama (i drugima) bez previše problema, rado ću to učiniti.

Pružanje povratnih informacija

Ako želite da mi date opšte povratne informacije o knjizi, možete mi poslati imejl na markjprice@gmail.com. Moj izdavač, Packt, je postavio Discord kanale za čitaoce da se povežu sa autorima i drugim čitaocima. Dobrodošli ste da nam se pridružite na sledećoj adresi: <https://packt.link/TS1e>.

Volim da čujem od svojih čitalaca šta im se dopada u mojim knjigama, kao i o predlozima za poboljšanja i kako rade sa C# i .NET okruženjem, zato nemojte biti stidljivi. Molim vas, javite mi se! Unapred vam zahvaljujem na vašim pažljivim i konstruktivnim povratnim informacijama.

Preuzimanje koda rešenja iz GitHub spremišta

Koristim GitHub za skladištenje rešenja za sve praktične, postupne zadatke kodiranja kroz poglavlja i praktične vežbe koje se nalaze na kraju svakog poglavlja. Spremište možete pronaći na sledećoj adresi: <https://github.com/markjprice/tools-skills-net8>.

Ako samo želite da preuzmete sve datoteke sa rešenjima bez korišćenja Git sistema, kliknite na zeleno dugme **Code**, a zatim izaberite **Download ZIP**.

Preporučujem da prethodnu adresu dodate u svoje omiljene ili oznake.

Dobra praksa

Najbolje je kopirati ili preuzeti kod rešenja na kratkoj udaljenosti do direktorijuma, kao što je C:\tools-skills-net8\ ili C:\book\, da biste izbegli da datoteke generisane prilikom izgradnje premaše maksimalnu dužinu putanje. Takođe bi trebalo da izbegavate specijalne karaktere, poput #. Na primer, nemojte koristiti ime direktorijuma kao što je C:\C#projects\. To ime direktorijuma može da funkcioniše za jednostavan konzolni projekat, ali kada počnete da dodajete funkcije koje automatski generišu kod, verovatno ćete imati čudne probleme. Neka imena direktorijuma budu kratka i jednostavna.

Gde potražiti pomoć

Ovaj deo se odnosi na pronalaženje kvalitetnih informacija o programiranju, na internetu.

Moje knjige su specifično napisane u stilu materijala sa uputstvima, tako da možete početi brzo i sa samopouzdanjem, pre nego što pređete na zvaničnu dokumentaciju za više detalja ili za specijalizovane situacije.

Ovu tehniku uglavnom koriste programeri da nauče da kodiraju. Više od 50% koristi knjige ili druge fizičke medije, a više od 80% koristi druge onlajn resurse, kao što je prikazano u anketi Stack Overflow platforme za 2023. godinu na sledećoj adresi: <https://survey.stackoverflow.co/2023/#section-learning-to-code-learning-how-to-code>.

Čitanje dokumentacije na platformi Microsoft Learn

Neprikosnoven resurs za pomoć sa Microsoft alatima i platformama za programere je tehnička dokumentacija na veb sajtu Microsoft Learn, a možete je pronaći na sledećoj adresi: <https://learn.microsoft.com/en-us/docs>.

Pomoć za alat dotnet

U komandnoj liniji, možete tražiti pomoć od alata dotnet za njegove komande. Sintaksa je sledeća:

```
dotnet help <komanda>
```

Ovako ćete u vašem veb pregledaču otvoriti stranicu sa dokumentacijom o određenoj komandi. Uobičajene `dotnet` komande su `new`, `build` i `run`.

Upozorenje

Komanda `dotnet help new` je radila na verzijama .NET Core 3.1 do verzije .NET 6, ali vraća grešku na .NET 7 i novijim verzijama: `Specified command 'new' is not a valid SDK command. Specify a valid SDK command. For more information, run dotnet help.` Nadamo se da će .NET tim uskoro popraviti tu grešku!

Drugi tip pomoći je dokumentacija komandne linije. Ona prati ovu sintaksu:

```
dotnet <komanda> -?|-h|--help
```

Na primer, `dotnet new -?`, `dotnet new -h` ili `dotnet new --help` prikazuju dokumentaciju o komandi `new` u komandnoj liniji.

Veliki jezički modeli kao što je ChatGPT

Jedna od najvećih promena u kodiranju i razvoju u poslednjih godinu dana je pojava alata generativne veštačke inteligencije i velikih jezičkih modela koji mogu da pruže pomoć za zadatke kodiranja, kao što su završavanje koda, implementacija cele funkcije, pisanje jediničnih testova i predlaganje popravki za otklanjanje grešaka postojećeg koda.

ChatGPT trenutno ima nekoliko modela: 3.5 i 4o (oba besplatna) i individualnu pretplatu Plus (20 dolara mesečno). Postoje i pretplate Team i Enterprise. Možete uporediti opcije na sledećoj adresi: <https://openai.com/chatgpt/pricing/>.

Recimo da želite da saznate više o alatima i veštinama za .NET. Možete uneti sledeći upit: „Koji su najvažniji alati i veštine koje profesionalni .NET programer treba da nauči?“

Odgovor bi mogao biti nešto poput sledećeg:

„Zdravo! Ako želite da postanete profesionalni .NET programer, postoji nekoliko ključnih alata i veština kojima bi trebalo da ovladate. Evo detaljnog pregleda:

Alati:

- **Visual Studio:** Ovaj IDE je standard za .NET razvoj. Uverite se da ste upoznati sa njegovim funkcijama i dodacima.
- **Visual Studio Code:** Dok je Visual Studio moćan, Visual Studio Code je lagan, svestran uređivač koda koji postaje sve popularniji među .NET programerima.
- **.NET Core/.NET 5 i noviji:** Ostanite u toku sa najnovijim verzijama .NET radnih okvira. .NET 5 i njegovi naslednici nude podršku za više platformi i poboljšanja performansi.
- **Git i GitHub:** Kontrola verzija je ključ. Git je popularan distribuirani sistem za kontrolu verzija, a GitHub pruža platformu za zajednički razvoj i deljenje koda.
- **Azure DevOps:** Upoznajte Azure DevOps za upravljanje projektima i neprekidnu integraciju i neprekidnu isporuku (CI/CD) protočnih struktura.
- **NuGet:** Koncept upravljanja paketima alatom NuGet je ključ za dodavanje biblioteka i paketa vašim .NET projektima.

- **Docker:** Kontejnerizacija postaje sve važnija za implementaciju .NET aplikacija. Docker vam pomaže da kreirate i efikasno upravljate kontejnerima.

Veštine:

- **C#:** Kad ste .NET programer, C# je vaš primarni jezik. Savladajte koncepte objektno-orijentisanog programiranja, asinhronog programiranja i LINQ jezika.
- **ASP.NET Core:** Za razvoj veb aplikacija, ASP.NET Core je prvi izbor radnog okvira. Učite o MVC arhitekturi, Razor Pages i razvoju API interfejsa.
- **Entity Framework Core:** Naučite da radite sa bazama podataka pomoću okvira Entity Framework Core za efikasan pristup podacima.
- **Ubrizgavanje zavisnosti:** Upoznajte principe ubrizgavanja zavisnosti i kako se u .NET okruženju koriste za upravljanje zavisnostima objekata.
- **Jedinično testiranje:** Pisanje jediničnih testova pomoću alata kao što su xUnit ili NUnit je ključno za obezbeđivanje pouzdanosti vašeg koda.
- **Dizajn veb API interfejsa:** Ako gradite API interfejse, naučite RESTful principe dizajna, autentifikaciju i verziranje.
- **Tehnologije na strani korisnika:** Upoznajte HTML, CSS, JavaScript i klijentske radne okvire kao što su React ili Angular ako želite da se bavite veb programiranjem.
- **Azure usluge:** Ako radite u oblaku, istražite Azure usluge, kao što su Azure Functions, Azure SQL Database i Azure App Service.
- **Bezbednost:** Naučite najbolje prakse za bezbednost, uključujući autentifikaciju, autorizaciju i zaštitu podataka.
- **Optimizacija performansi:** Učite o profilisanju, keširanju i drugim tehnikama za optimizaciju performansi vaših aplikacija.
- **Agilne metodologije:** Agilne prakse kao što su Scrum i Kanban su često korišćene u razvoju softvera. Poznavanje ovih metoda može da poboljša timsku saradnju.
- **Socijalne veštine:** Komunikacija, rešavanje problema i timski rad su ključne socijalne veštine svakog profesionalnog programera.

Pomenuo sam dosta toga, ali zapamtite da se tehnološki pejzaž stalno razvija. Radoznalost i neprekidno učenje su same po sebi važne veštine. Ako imate neka specifična pitanja ili vam treba-ju dodatni detalji o bilo kojoj od ovih tema, slobodno pitajte!

Postoji nekoliko tema koje ChatGPT pominje, a moje knjige ne. To su:

- **Azure DevOps i Azure usluge:** Za ovu knjigu sam odlučio da uklonim zahteve za Azure i druge usluge u oblaku. Želim da svi čitaoci moje knjige budu u mogućnosti da završe sve zadatke na svom lokalnom računaru bez plaćanja za nalog u oblaku. Želim da izbegnem da čitaoci budu usmereni na bilo kog pružaoca usluga u oblaku, tako da će moje knjige od sada biti neutralne prema oblaku.
- **Tehnologije na strani korisnika:** HTML, CSS i JavaScript, kao i mnogi klijentski radni okviri za razvoj aplikacija kao što su React i Angular.
- **Agilne metodologije:** Ova i druge metodologije zaista zahtevaju celu knjigu da bi se temeljno obradile. Ako uspem da nađem način da ih korisno uvedem na samo nekoliko stranica, to bi se moglo naći u sledećem izdanju.

Dobijanje bolje pomoći od velikih jezičkih modela pomoću inženjeringa upita

Možda ste čuli za pojam **inženjering upita**. Ovo je važna veština za svakog profesionalca, jer će moderni zadaci često biti efikasnije obavljani u saradnji sa veštačkom inteligencijom.

Napomena

Alati veštačke inteligencije koji se blisko integrišu sa vašim uređivačem koda, kao što je GitHub Copilot, su obrađeni u poglavlju 2, *Najbolja upotreba alata u uređivaču koda*.

Da biste razjasnili ili detaljnije razumeli neku temu ili bilo koji deo informacije, koristite sledeće šablone upita:

- Objasnite [navedite određenu temu] jednostavnim rečima.
- Objasnite mi kao da sam početnik u [polju].
- Objasnite mi na prirodan, ljudski način.
- Objasnite mi kao da imam 10 godina.
- Napišite mi detaljan [esej/tekst/pasus] o [temi] dodavanjem svih potrebnih informacija.

Koristite afirmativna uputstva i izbegavajte negativne formulacije kao što je „nemoj“.

Za složene odgovore ili u situacijama kada veštačka inteligencija daje površan odgovor, dodajte frazu „razmišljaj korak po korak“.

Dodelite ulogu velikim jezičkim modelima. Na primer: „Vi ste stručnjak za arhitekturu rešenja sa decenijama iskustva u implementaciji .NET okruženja.“

Ako niste sigurni kako da formulišete upit, zatražite pomoć od velikog jezičkog modela! Na primer:

„Želim da budete moj inženjer upita. Vaš cilj je da mi pomognete da kreiram najbolji upit za moje potrebe. Upit ćete koristiti vi, ChatGPT. Sledite ove korake: Vaš prvi odgovor će biti da me pitate o čemu bi upit trebalo da bude. Ja ću dati svoj odgovor, ali moramo zajedno raditi na njegovom poboljšanju kroz stalne iteracije sledećih koraka. Na osnovu mog unosa, generisaćete tri odeljka:

1. Ispravljen upit. Molim vas da obezbedite vaš preformulisani upit. Treba da bude jasan, sažet i lako razumljiv.
2. Predlozi. Molim vas da date ideje o tome koje detalje uključiti u upit da biste ga poboljšali.
3. Pitanja. Molim vas postavite sva relevantna pitanja da biste prikupili dodatne informacije od mene i poboljšali upit.

Nastavićemo ovaj iterativni proces tako što ću vam ja pružati više informacija, a vi ažurirati upit u odeljku Ispravljen upit dok ne bude dovršen. Hvala!

Napomena

Ako imate računar sa Nvidia GTX karticom, možete preuzeti besplatnu aplikaciju za konverzaciju za prilagođavanje i razgovor sa lokalnim velikim jezičkim modelima. Ova aplikacija čak može da obrađuje datoteke kao što je PDF. Ako neko od vas pokuša da doda PDF bilo koje od mojih knjiga da biste mogli da postavljate pitanja o njima, javite mi kako vam ide. Više informacija možete pronaći na sledećoj adresi:

<https://blogs.nvidia.com/blog/chat-with-rtx-available-now/>

Upotreba veštačke inteligencije od strane programera

Prema anketi Stack Overflow platforme iz 2023. godine (<https://survey.stackoverflow.co/2023/#section-sentiment-and-usage-ai-tools-in-the-development-process>), „70% svih ispitanika koristi ili planira da koristi alate veštačke inteligencije u svom procesu razvoja, ove godine. Veća je verovatnoća da koriste ili će koristiti alate veštačke inteligencije oni koji uče da kodiraju, u poređenju sa profesionalnim programerima (82% naspram 70%).“

Pomoć na Discord platformi i drugim forumima za konverzaciju

Postavljanje pitanja na programerskim forumima i Discord kanalima je umetnost, jednako kao i nauka. Da biste povećali svoje šanse da dobijete koristan odgovor, treba da težite kombinaciji jasnoće, specifičnosti i svesti o zajednici.

Evo nekoliko saveta za postavljanje pitanja:

- **Postavljajte pitanja na javnim kanalima, a ne privatno:** Molim vas, nemojte slati direktne poruke autoru sa pitanjem ili zahtevom za prijateljstvo. Zapamtite, svako pitanje i odgovor grade kolektivno znanje i snalažljivost cele zajednice. Postavljanje pitanja javno, takođe, omogućava drugim čitaocima da vam pomognu, a ne samo autor. Zajednica koju smo Pakt i ja izgradili oko mojih knjiga je prijateljska i pametna. Dozvolite da vam svi pomognemo.
- **Istražujte pre nego što pitate:** Važno je da sami potražite odgovore pre nego što se obratite zajednici. Koristite pretraživače, zvaničnu dokumentaciju i funkciju pretrage unutar foruma ili Discord servera. Ovo ne samo da poštuje vreme zajednice, već vam pomaže da efikasnije učite. Još jedno mesto na kom prvo možete da potražite odgovore je odeljak za greške u knjizi na sledećoj adresi: <https://github.com/markjprice/tools-skills-net8/blob/main/docs/errata/README.md>.
- **Budite specifični i sažeti:** Jasno navedite šta pokušavate da postignete, šta ste do sada pokušali i gde ste zapeli. Na sažeto pitanje je verovatnije da ćete dobiti brz odgovor.
- **Navedite mesto u knjizi:** Ako ste zapeli na određenom mestu u knjizi, navedite broj stranice i naslov odeljka da bi drugi mogli da pronađu kontekst vašeg pitanja.
- **Pokažite svoj rad:** Dokaz da ste se i sami potrudili da rešite problem pruža kontekst i pomaže drugima da razumeju vaš način razmišljanja i gde ste možda pogrešili.
- **Pripremite svoje pitanje:** Izbegavajte preširoka ili previše nejasna pitanja. Snimci ekrana grešaka ili delova koda (sa pravilnim formatiranjem) mogu biti veoma korisni.

Napomena:

Čudno, vidim da sve više čitalaca slika svoj ekran i objavljuje te slike. One su teže za čitanje i ograničene mogućnosti prikaza. Bolje je kopirati i umetnuti tekst vašeg koda ili poruke o grešci da bi i drugi mogli da ga kopiraju i umetnu. Ili, bar, napravite snimak ekrana visoke rezolucije, umesto fotografije!

- **Pravilno formatirajte svoj kod:** Većina foruma i Discord servera podržava formatiranje koda pomoću Markdown sintakse. Koristite formatiranje da bi vaš kod bio čitljiviji. Na primer, okružite ključne reči koda jednostrukim apostrofima, kao: ``public void``, i okružite blokove koda sa tri apostrofa sa opcionim kodom jezika, kao što je prikazano u sledećem kodu:

```
```cs
using static System.Console;
WriteLine("Ove je formatiran C# kod.");
```
```

Dobra praksa

Nakon tri apostrofa koji započinju blok koda u Markdown sintaksi, navedite kod jezika, kao što su `cs, csharp, js, javascript, json, html, css, cpp, xml, mermaid, python, java, ruby, go, sql, bash, ili shell`.

Dodatne informacije

Da biste naučili da formatirate tekst u porukama Discord kanala, posetite sledeću adresu: <https://support.discord.com/hc/en-us/articles/210298617-Markdown-Text-101-Chat-Formatting-Bold-Italic-Underline>.

- **Budite pristojni i strpljivi:** Zapamtite, tražite pomoć od ljudi koji svoje vreme daju dobrovoljno. Pristojan ton i strpljenje dok čekate odgovor mnogo znače. Učesnici kanala su često u drugoj vremenskoj zoni, pa možda neće videti vaše pitanje do sledećeg dana.
- **Budite spremni da aktivno učestvujete:** Nakon što postavite pitanje, ostanite angažovani. Možda ćete dobiti dodatna pitanja za pojašnjenje. Brzo i jasno odgovaranje može značajno povećati vaše šanse za dobijanje korisnog odgovora. Kada postavim pitanje, podesim alarm na tri sata kasnije da se vratim i vidim da li je neko odgovorio. Ako nije bilo odgovora, onda postavim još jedan alarm za 24 sata.

Primena ovih pristupa prilikom postavljanja pitanja ne samo da povećava verovatnoću dobijanja korisnog odgovora, već i pozitivno doprinosi zajednici ukazivanjem poštovanja za vreme i trud drugih učesnika.

Dobra praksa

Nikada nemojte reći samo „Zdravo“ na bilo kom sistemu za konverzaciju. Razlog zašto tako možete pročitati na sledećoj adresi: <https://nohello.net/>. Takođe, nemojte pitati da pitate: <https://dontasktoask.com/>.

Podešavanje baze podataka i projekata za ovu knjigu

Potrebna nam je baza podataka i neki projekti koje ćemo koristiti tokom ove knjige. Da bi sve bilo koliko-toliko realistično, potrebni su nam različiti projekti koji koriste zajedničke funkcionalnosti kao što su SQL Server baza podataka, biblioteke klasa, jedinično testiranje i slično.

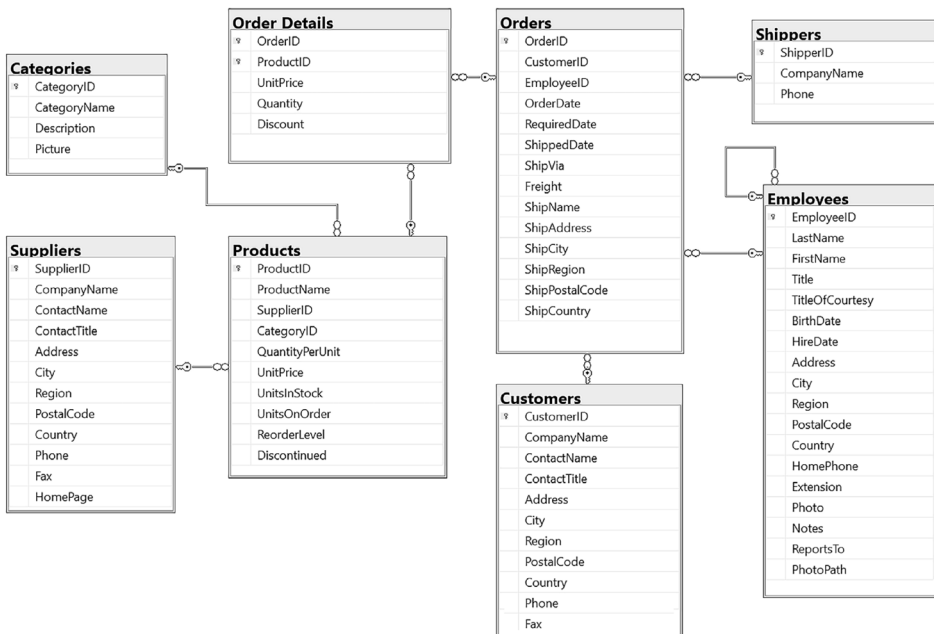
Definisaćemo model entiteta podataka kao par klasa koje se mogu ponovo koristiti. Jedan deo para će definisati entitete kao što su `Product` i `Customer`. Drugi deo para će definisati tabele u bazi podataka, podrazumevanu konfiguraciju za povezivanje sa bazom podataka i koristiće Fluent API interfejs za dodatne opcije konfiguracije modela.

Napravićemo tri projekta:

- Biblioteku klasa za modele entiteta kao što su `Category` i `Product` pod nazivom `Northwind.EntityModels`.
- Biblioteku klasa za EF Core kontekst podataka pod nazivom `Northwind.DataContext`.
- xUnit projekat za jedinične i integracione testove pod nazivom `Northwind.Tests`.

Upotreba uzorka relacione baze podataka

Bilo bi korisno imati uzorak baze podataka koja ima srednju složenost i pristojan broj uzoraka zapisa. Microsoft nudi nekoliko uzoraka baza podataka, od kojih je većina previše složena za naše potrebe, pa ćemo umesto toga koristiti bazu podataka koja je prvi put kreirana ranih 1990-ih godina, poznatu kao **Northwind**. Hajde da odvojimo minut da pogledamo dijagram Northwind baze podataka i njenih osam najvažnijih tabela. Možete koristiti dijagram na *slici 1.2* kao referencu dok budemo pisali kod i upite tokom ove knjige:



Slika 1.2: Tabele i veze baze podataka Northwind

Obratite pažnju na sledeće:

- Svaka kategorija ima jedinstven identifikator, naziv, opis i sliku. Slika je sačuvana kao niz bajtova u JPEG formatu.
- Svaki proizvod ima jedinstveni identifikator, naziv, jediničnu cenu, broj jedinica na lageru i druge kolone.
- Svaki proizvod je povezan sa kategorijom tako što čuva jedinstveni identifikator kategorije.
- Veza između tabela Categories i Products je jedan-prema-više, što znači da svaka kategorija može imati nula, jedan ili više proizvoda.
- Svaki proizvod dostavlja dobavljačka kompanija označena jedinstvenim identifikatorom dobavljača.
- Količina i jedinična cena proizvoda su sačuvani za svaki detalj narudžbine.
- Svaka narudžbina je napravljena od strane kupca, preuzeta od strane zaposlenog i isporučena od strane kompanije za dostavu.
- Svaki zaposleni ima ime, adresu, kontakt podatke, datume rođenja i zapošljavanja, referencu na svog menadžera (osim šefa, za koga u polju `ReportsTo` stoji `null`) i fotografiju koja je sačuvana kao niz bajtova u JPEG formatu. Tabela ima vezu jedan-prema-više sa samom sobom, jer jedan zaposleni može da upravlja mnogim drugim zaposlenima.

Podešavanje sistema SQL Server i Northwind baze podataka

Microsoft nudi razne verzije svog popularnog i moćnog SQL Server proizvoda za Windows, Linux i Docker kontejnere.

Napomena

Ako imate Windows, možete koristiti besplatnu verziju koja radi samostalno, poznatu kao SQL Server Developer Edition. Takođe, možete koristiti Express izdanje ili besplatno izdanje SQL Server LocalDB koje se može instalirati sa Visual Studio okruženjem. Da instalirate SQL Server lokalno na Windows operativnom sistemu, pogledajte uputstva na sledećoj adresi: <https://github.com/markjprice/tools-skills-net8/blob/main/docs/sql-server/README.md>. Ako više volite da instalirate SQL Server lokalno na Linux sistemu, naći ćete uputstva na sledećoj adresi: <https://learn.microsoft.com/en-us/sql/linux/sql-server-linux-setup>.

Ako nemate Windows računar ili ako želite da koristite sistem baze podataka koji je prilagođen za različite platforme, pogledajte onlajn prilog, **Instaliranje Azure SQL Edge na Docker**, koji se nalazi na sledećoj adresi: <https://github.com/markjprice/tools-skills-net8/blob/main/docs/sql-server/edge.md>.

Potrebno je da postavite SQL Server, pokrenete SQL skript za kreiranje baze podataka Northwind i potvrdite da možete da se povežete sa bazom podataka i pregledate redove u njenim tabelama, kao što su `Products` i `Categories` pre nego što nastavite sa projektom. Sledeća dva pododeljka sadrže detaljna uputstva koja vam pomažu da to uradite koristeći lokalni SQL Server ili SQL Edge na Docker platformi. Preskočite naredne odeljke ako ste već sve postavili.

Kreiranje Northwind baze podataka za lokalni SQL Server

Da biste pokrenuli SQL skript za kreiranje uzorka baze podataka Northwind za lokalni SQL Server:

1. Ako prethodno niste preuzeli ili klonirali GitHub spremište za ovu knjigu, uradite to sada: <https://github.com/markjprice/tools-skills-net8/>.
2. Kopirajte skript za kreiranje baze podataka Northwind za SQL Server sa sledeće putanje u vašem lokalnom Git spremištu: `/scripts/sql-scripts/Northwind4SQLServer.sql` u radni direktorijum.
3. Pokrenite **SQL Server Management Studio**.
4. U dijalogu **Connect to Server**, za ime servera, unesite `.` (tačku), što znači ime lokalnog računara, a zatim kliknite na dugme **Connect**. Ako ste morali da kreirate imenovanu instancu, kao što je `tools-skills-net8`, unesite `.\tools-skills-net8`.
5. Idite na **File | Open | File...**
6. Pronađite i izaberite datoteku `Northwind4SQLServer.sql` i kliknite na dugme **Open**.
7. U traci sa alatima kliknite na **Execute** i obratite pažnju na poruku **Command(s) completed successfully**.
8. U **Object Explorer**, proširite bazu podataka **Northwind**, a zatim proširite **Tables**.
9. Desnim klikom na **Products**, kliknite na **Select Top 1000 Rows** i obratite pažnju na vraćene rezultate.
10. Napustite **SQL Server Management Studio**.

Kreiranje Northwind baze podataka za SQL Edge na Docker platformi

Da biste pokrenuli skript baze podataka za kreiranje uzorka baze podataka Northwind za SQL Edge na Docker platformi:

1. U svom omiljenom uređivaču koda otvorite datoteku `Northwind4AzureSQLEdge.sql`.
2. Povežite se sa SQL Edge na Docker platformi pomoću sledećih informacija o povezivanju:
 - **Data Source** ili **server**: `tcp:127.0.0.1,1433`
 - Morate koristiti **SQL Server Authentication**, odnosno **SQL prijavu**, što znači da morate uneti korisničko ime i lozinku. Azure SQL Edge slika već ima kreiranog sa korisnika i morali ste mu dati jaku lozinku kada ste pokrenuli kontejner. Izabrali smo lozinku `s3cret-Ninja`.
 - **Database**: `master` ili ostavite prazno. Kreiraćemo bazu podataka Northwind pomoću SQL skripta.
3. Izvršite SQL skript:
 - Ako koristite Visual Studio, kliknite desnim tasterom miša na skript, izaberite **Execute**, i zatim sačekajte poruku **Command completed successfully**.
 - Ako koristite Code, kliknite desnim tasterom miša na skript, izaberite **Execute Query**, izaberite profil veze za **Azure SQL Edge in Docker**, a zatim sačekajte poruku **Commands completed successfully**.
4. Osvežite vezu sa podacima:

- Ako koristite Visual Studio, u **Server Explorer**, kliknite desnim tasterom miša na **Tables** i izaberite **Refresh**.
 - Ako koristite Code, kliknite desnim tasterom miša na profil veze za **Azure SQL Edge in Docker** i izaberite **Refresh**.
5. Proširite **Databases**, proširite **Northwind**, a zatim proširite **Tables**.
 6. Primitićete da je kreirano 13 tabela, na primer, **Categories**, **Customers** i **Products**. Takođe ćete primiti da je kreirano na desetine pogleda i ugrađenih procedura.
 7. Sledeće, definišaćemo model entiteta podataka za bazu podataka Northwind kao par biblioteka klasa koje se mogu ponovo koristiti.

Dobra praksa

Trebalo bi da kreirate poseban projekat biblioteke klasa za svoje modele entiteta podataka. Ovo omogućava lakše deljenje između serverskih i korisničkih desktop, mobilnih i Blazor aplikacija.

Kreiranje biblioteke klasa za modele entiteta pomoću sistema SQL Server

Sada ćete kreirati modele entiteta pomoću alata `dotnet-ef`:

1. U svom omiljenom uređivaču koda, kreirajte novi projekat prema sledećoj listi:
 - Šablon projekta: **Class Library** / `classlib`
 - Datoteka i direktorijum projekta: `Northwind.EntityModels`
 - Datoteka i direktorijum rešenja: `Chapter01`
2. U projektu `Northwind.EntityModels`, prikažite upozorenja kao greške i dodajte reference na pakete za SQL Server dobavljač baze podataka i EF Core podršku u fazi dizajniranja, kao što je istaknuto u sledećem kodu:

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFramework>net8.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
    <TreatWarningsAsErrors>true</TreatWarningsAsErrors>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference
      Include="Microsoft.EntityFrameworkCore.SqlServer"
      Version="8.0.6"
    />
    <PackageReference
      Include="Microsoft.EntityFrameworkCore.Design"
      Version="8.0.6">
```

```
<PrivateAssets>all</PrivateAssets>
<IncludeAssets>runtime; build; native; contentfiles;
analyzers;
buildtransitive</IncludeAssets>
</PackageReference>
</ItemGroup>
</Project>
```

Napomena

Možete proveriti najnovije verzije paketa na sledećim adresama: <https://www.nuget.org/packages/Microsoft.EntityFrameworkCore.SqlServer> i <https://www.nuget.org/packages/Microsoft.EntityFrameworkCore.Design>.

Upozorenje

Ako niste upoznati sa načinom na koji paketi kao što je `Microsoft.EntityFrameworkCore.Design` mogu upravljati svojim resursima, možete saznati više na sledećoj adresi: <https://learn.microsoft.com/en-us/nuget/consume-packages/package-references-in-project-files#controlling-dependency-assets>.

Dobra praksa

Podrazumevano, kompajler može prikazati upozorenja ako postoje potencijalni problemi sa vašim kodom kada prvi put kompajlirate projekat, ali ona ne sprečavaju kompajliranje i sakrivaju se ako ponovite kompajliranje. Upozorenja se daju s razlogom, tako da ignorisanje upozorenja podstiče loše prakse razvoja. Preporučujem da naterate sebe da ispravite upozorenja omogućavanjem opcije da prikazujete upozorenja kao greške.

3. Obrišite datoteku `Class1.cs`.
4. Kompajlirajte projekat `Northwind.EntityModels`.
5. Otvorite komandnu liniju ili terminal za direktorijum `Northwind.EntityModels`.
6. Ako još uvek nemate alat `dotnet-ef`, instalirajte najnoviju verziju pomoću sledeće komande:

```
dotnet tool install --global dotnet-ef
```

7. Umesto instaliranja, možete izvršiti ažuriranje pomoću sledeće komande:

```
dotnet tool update --global dotnet-ef
```

Napomena

Sledeći korak pretpostavlja nisku veze sa bazom podataka za lokalni SQL Server sa Windows integrisanom autentifikacijom. Prilagodite ga za Azure SQL Edge sa korisničkim identifikatorom i lozinkom ako je potrebno.

8. U komandnoj liniji generišite modele klasa entiteta za sve tabele, pomoću sledećih komandi:

```
dotnet ef dbcontext scaffold "Data Source=.;Initial
```

```
Catalog=Northwind;Integrated Security=true;TrustServerCertificate=True;"
Microsoft.EntityFrameworkCore.SqlServer --namespace Northwind.
EntityModels --data-annotations
```

Obratite pažnju na sledeće:

- Komanda koja se izvršava: `dbcontext scaffold`.
 - Niska veze: `"Data Source=.;Initial Catalog=Northwind;Integrated Security=true;TrustServerCertificate=True;"`.
 - Dobavljač baze podataka: `Microsoft.EntityFrameworkCore.SqlServer`.
 - Prostor imena za generisane klase: `--namespace Northwind.EntityModels`
 - Za korišćenje anotacija podataka kao i Fluent API interfejsa: `--data-annotations`.
9. Primitićete da je generisano 28 klasa, od `AlphabeticalListOfProduct.cs` do `Territory.cs`.
10. Na vrhu datoteke `NorthwindContext.cs`, uvezite prostor imena za rad sa ADO.NET tipovima, kao što je prikazano u sledećem kodu:

```
using Microsoft.Data.SqlClient; // Za korišćenje SqlConnectionStringBuilder.
```

11. Izmenite metod `OnConfiguring` da dinamički postavi nisku veze i postavi sve osetljive parametre pomoću promenljive okruženja, kao što je prikazano u sledećem kodu.

```
protected override void OnConfiguring(
    DbContextOptionsBuilder optionsBuilder)
{
    // Ako već nije konfigurisan od strane klijentskog projekta. Na primer,
    // klijentski projekat može koristiti AddNorthwindContext da bi
    // predefinisao nisku veze baze podataka.

    if (!optionsBuilder.IsConfigured)
    {
        SqlConnectionStringBuilder builder = new();

        builder.DataSource = ".";
        builder.InitialCatalog = "Northwind";
        builder.TrustServerCertificate = true;
        builder.MultipleActiveResultSets = true;

        // Ako koristi Azure SQL Edge.
        // builder.DataSource = "tcp:127.0.0.1,1433";

        // Zato što želimo brže da otkrijemo greške. Podrazumevano 15 sekundi.
        builder.ConnectTimeout = 3;
```



```

// Ako koristi Windows integrisanu autentifikaciju.
builder.IntegratedSecurity = true;

// Ako koristi SQL Server autentifikaciju.
// builder.UserID = Environment.GetEnvironmentVariable("MY_SQL_USR");
// builder.Password = Environment.GetEnvironmentVariable("MY_SQL_
PWD");

optionsBuilder.UseSqlServer(builder.ConnectionString);
}
}

```

12. U `Customer.cs`, alat `dotnet-ef` je ispravno identifikovao da je kolona `CustomerId` primarni ključ i da je ograničena na najviše pet karaktera, ali mi takođe želimo da vrednosti uvek budu velika slova. Dakle, dodajte regularni izraz da proverava vrednost primarnog ključa kako bi dozvolio samo velika slova Western karaktera, kao što je istaknuto u sledećem kodu:

```

[Key]
[StringLength(5)]
[RegularExpression("[A-Z]{5}")]
public string CustomerId { get; set; } = null!;

```

Kreiranje biblioteke klasa za kontekst podataka pomoću sistema SQL Server

Sledeće, premestićete model konteksta koji predstavlja bazu podataka u posebnu biblioteku klasa:

1. Dodajte novi projekat prema sledećoj listi:
 - Šablon projekta: **Class Library** / `classlib`
 - Datoteka i direktorijum projekta: `Northwind.DataContext`
 - Datoteka i direktorijum rešenja: `Chapter01`
2. U projektu `DataContext`, dodajte referencu na projekat `EntityModels` i dodajte referencu na paket EF Core dobavljača podataka za SQL Server, kao što je prikazano u sledećem kodu:

```

<ItemGroup>
  <PackageReference
    Include="Microsoft.EntityFrameworkCore.SqlServer" Version="8.0.6" />
</ItemGroup>

<ItemGroup>
  <ProjectReference Include=
    "..\Northwind.EntityModels\Northwind.EntityModels.csproj" />
</ItemGroup>

```

Napomena

Možete isprobati probne verzije alata EF Core 9 tako što ćete navesti verziju 9.0-*. Ciljni radni okvir za vaš projekat treba i dalje da koristi net8.0. Korišćenjem džokera, automatski ćete preuzeti najnoviju mesečnu probnu verziju kada obnovite pakete za projekat. Kada EF Core 9 GA verzija bude objavljena u novembru 2024. godine, promenite verziju paketa na 9.0.0, ili noviju. Nakon februara 2025. godine, moći ćete da uradite slično sa EF Core 10 (koristite verziju paketa 10.0-*), ali to će verovatno zahtevati da projekat cilja net10.0, pa ćete morati da instalirate probnu verziju .NET 10 SDK platforme.

3. U projektu Northwind.DataContext, obrišite datoteku Class1.cs.
4. Kompajlirajte projekat Northwind.DataContext.
5. Premestite datoteku NorthwindContext.cs iz projekta/direktorijuma Northwind.EntityModels u projekat/direktorijum Northwind.DataContext.
6. U projektu Northwind.DataContext, dodajte klasu pod nazivom NorthwindContextExtensions.cs i izmenite njen sadržaj da definiše dodatni metod koji dodaje Northwind kontekst baze podataka u kolekciju usluga zavisnosti, kao što je prikazano u sledećem kodu:

```
using Microsoft.Data.SqlClient; // SqlConnectionStringBuilder
using Microsoft.EntityFrameworkCore; // UseSqlServer
using Microsoft.Extensions.DependencyInjection; // IServiceCollection

namespace Northwind.EntityModels;

public static class NorthwindContextExtensions
{
    /// <summary>
    /// Dodajte NorthwindContext u navedenu kolekciju IServiceCollection.
    /// Koristi SQL Server dobavljač baze podataka.
    /// </summary>
    /// <param name="services">Kolekcija usluga.</param>
    /// <param name="connectionString">Postavite da predefinišete
    /// podrazumevanu vrednost.</param>
    /// <returns>IServiceCollection može da se koristi za dodavanje više
    /// usluga.</returns>
    public static IServiceCollection AddNorthwindContext(
        this IServiceCollection services,
        string? connectionString = null)
    {
        if (connectionString == null)
        {
            SqlConnectionStringBuilder builder = new();

            builder.DataSource = ".";
        }
    }
}
```

```
builder.InitialCatalog = "Northwind";
builder.TrustServerCertificate = true;
builder.MultipleActiveResultSets = true;

// Ako koristi Azure SQL Edge.
// builder.DataSource = "tcp:127.0.0.1,1433";

// Zato što želimo brže da otkrijemo greške. Podrazumevano 15
// sekundi.
builder.ConnectTimeout = 3;

// Ako koristi Windows integrisanu autentifikaciju.
builder.IntegratedSecurity = true;

// Ako koristi SQL Server autentifikaciju.
// builder.UserID = Environment.GetEnvironmentVariable("MY_SQL_
USR");
// builder.Password = Environment.GetEnvironmentVariable("MY_SQL_
PWD");

connectionString = builder.ConnectionString;
}

services.AddDbContext<NorthwindContext>(options =>
{
options.UseSqlServer(connectionString);

// Kreiranje zapisa u konzoli prilikom izvršavanja EF Core kom-
andi.
options.LogTo(Console.WriteLine,
new[] { Microsoft.EntityFrameworkCore
.Diagnostics.RelationalEventId.CommandExecuting });
},
// Registrujte sa prolaznim vekom trajanja da izbegnete probleme sa
// konkurentnošću u Blazor Server projektima.
contextLifetime: ServiceLifetime.Transient,
optionsLifetime: ServiceLifetime.Transient);

return services;
}
}
```

7. Kompajlirajte obe biblioteke klasa i ispravite sve greške prilikom kompajliranja.

Dobra praksa

Obezbedili smo opcioni argument za metod `AddNorthwindContext` tako da možemo da predefinišemo SQL Server nisku veze baze podataka. To će nam omogućiti veću fleksibilnost da, na primer, učitam ove vrednosti iz konfiguracione datoteke.

Kreiranje testnog projekta za proveru integracije biblioteka klasa

Pošto u ovom poglavlju nećemo kreirati klijentski projekat koji koristi EF Core model, trebalo bi da napravimo testni projekat da bismo bili sigurni da se kontekst baze podataka i modeli entiteta ispravno integrišu:

1. Koristite svoj omiljeni uređivač koda da dodate novi **xUnit Test Project [C#]** / xunit projekat pod nazivom `Northwind.Tests` u rešenje `Chapter01`.
2. U datoteci `Northwind.Tests.csproj`, izmenite konfiguraciju tako da se upozorenja prikazuju kao greške i dodajte grupu stavki sa referencom na projekat `Northwind.DataContext`, kao što je prikazano u sledećem kodu:

```
<ItemGroup>
  <ProjectReference Include=
    "..\Northwind.DataContext\Northwind.DataContext.csproj" />
</ItemGroup>
```

Upozorenje!

Putanja do reference na projekat u vašoj projektnoj datoteci ne bi trebalo da sadrži prelome linija.

3. Kompajlirajte projekat `Northwind.Tests` da biste izgradili i obnovili zavisnosti projekta.
4. Preimenujte datoteku `UnitTest1.cs` u `NorthwindEntityModelsTests.cs` (Visual Studio će vas pitati da preimenujete klasu kada preimenujete datoteku).
5. U datoteci `NorthwindEntityModelsTests.cs`, ako koristite Code, ručno preimenujte klasu u `NorthwindEntityModelsTests`.
6. U datoteci `NorthwindEntityModelsTests.cs`, izmenite klasu da uvozi `Northwind.EntityModels` prostor imena i dodajte nekoliko test metoda koji osiguravaju da klasa konteksta može da se poveže, da je dobavljač SQL Server i da je prvi proizvod nazvan `Chai`, kao što je prikazano u sledećem kodu:

```
using Northwind.EntityModels; // Da koristimo NorthwindContext i Product.
```

```
namespace Northwind.Tests;
```

```
public class NorthwindEntityModelsTests
{
```

```
[Fact]
public void CanConnectIsTrue()
{
    using (NorthwindContext db = new()) // pripremi
    {
        bool canConnect = db.Database.CanConnect(); // testiraj
        Assert.True(canConnect); // proveriti
    }
}

[Fact]
public void ProviderIsSqlServer()
{
    using (NorthwindContext db = new())
    {
        string? provider = db.Database.ProviderName;
        Assert.Equal("Microsoft.EntityFrameworkCore.SqlServer", provider);
    }
}

[Fact]
public void ProductId1IsChai()
{
    using (NorthwindContext db = new())
    {
        Product? product1 = db?.Products?.Single(p => p.ProductId == 1);
        Assert.Equal("Chai", product1?.ProductName);
    }
}
}
```

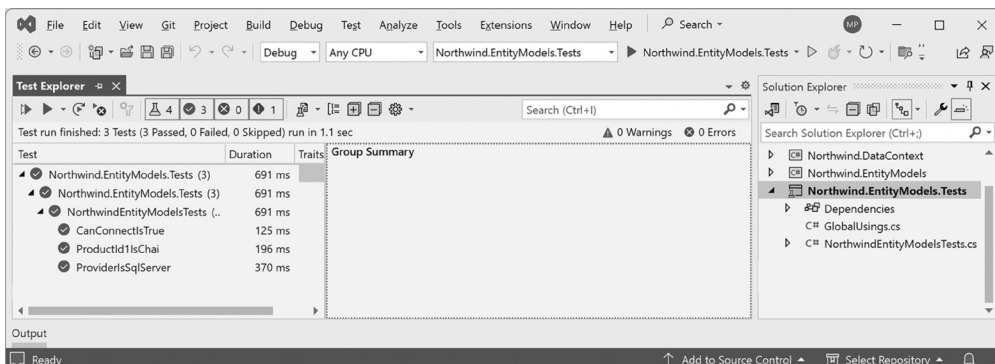
Pokretanja testova

Sada smo spremni da pokrenemo testove i vidimo rezultate pomoću okruženja Visual Studio ili uređivača Code.

Upotreba okruženja Visual Studio:

1. U okruženju **Visual Studio**, na kartici **Solution Explorer**, desnim klikom na projekat `Northwind.Tests` izaberite **Run Tests**.

2. Na kartici **Test Explorer**, primetićete da rezultati pokazuju da su tri testa izvršena i da su svi uspeli, kao što je prikazano na *slici 1.3*:



Slika 1.3: Svi testovi su uspeli

Upotreba uređivača Code:

1. U uređivaču **Code**, u **TERMINAL** prozoru projekta `Northwind.Tests`, pokrenite testove pomoću sledeće komande:

```
dotnet test
```

Napomena

Ako koristite **C# Dev Kit**, takođe možete da izgradite testni projekat, a zatim da pokrenete testove iz odeljka **Testing** u **Primary Side Bar**.

2. Na izlazu, primetićete da rezultati pokazuju da su tri testa izvršena i da su svi uspeli.

Upotreba .NET 9 uz ovu knjigu

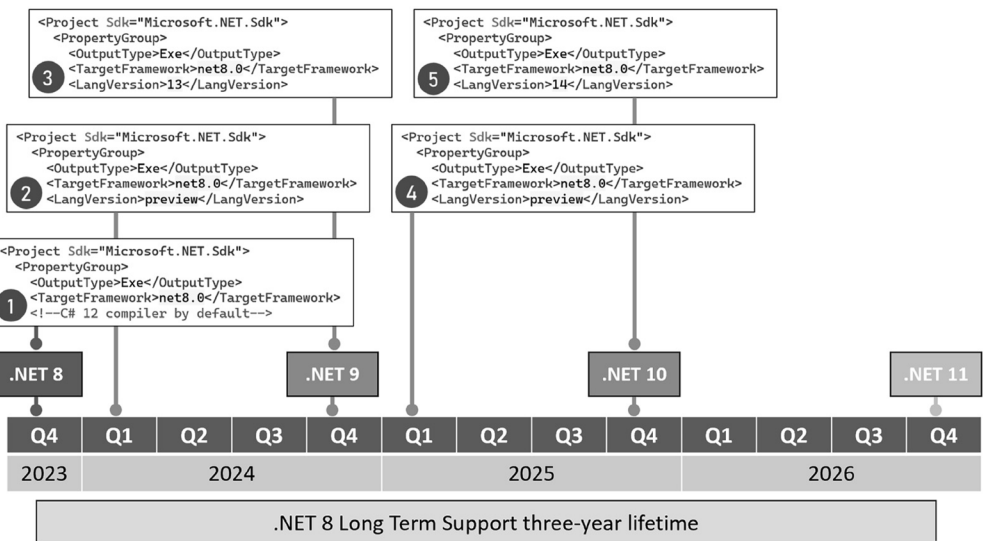
Microsoft će objaviti **.NET 9** na konferenciji **.NET Conf 2024** u utorak, 12. novembra 2024, nedelju dana nakon predsedničkih izbora u Sjedinjenim Američkim Državama 5. novembra. Mnogi čitaoci će želeti da koriste ovu knjigu sa **.NET 9**, pa ovaj deo objašnjava kako da to urade.

U vreme izdavanja, u julu 2024. godine, **.NET 9** je već dostupan kao probna verzija, ili možete da sačekate konačnu verziju, najavljenju za novembar 2024. godine. Ali budite oprezni, jer jednom kada instalirate **.NET 9 SDK**, on će se podrazumevano koristiti za sve **.NET** projekte, osim ako ga ne promenite pomoću datoteke `global.json`. Više o tome možete saznati na sledećoj adresi: <https://learn.microsoft.com/en-us/dotnet/core/tools/global-json>.

Možete lako nastaviti da ciljate na **.NET 8** izvršno okruženje dok instalirate i koristite buduće **C#** kompajlere, kao što je prikazano na *slici 1.4* i ilustrovano na sledećem spisku:

1. **Od novembra 2023. nadalje:** Instalirajte **.NET SDK 8.0.100** ili noviji i koristite ga za izgradnju projekata koji ciljaju na **.NET 8** i podrazumevano koriste **C# 12** kompajler. Svakog meseca, ažurirajte **.NET 8 SDK** zakrpe na razvojnom računaru i ažurirajte zakrpe **.NET 8** izvršnog okruženja na bilo kom računaru za implementaciju.

2. **Od februara do oktobra 2024:** Opciono, instalirajte .NET SDK 9 probne verzije svakog meseca da biste istražili nove funkcije C# jezika i .NET biblioteke. Imajte na umu da nećete moći da koristite nove funkcije biblioteke dok ciljate na .NET 8. Da biste saznali o novim funkcijama u toj probnoj verziji, pročitajte mesečne objave dostupne na sledećoj adresi: <https://github.com/dotnet/core/discussions/9234>.
3. **Od novembra 2024. nadalje:** Instalirajte .NET SDK 9.0.100 ili noviji i koristite ga za izgradnju projekata koji nastavljaju da ciljaju na .NET 8 i koriste C# 13 kompajler za nove funkcije. Koristićete potpuno podržan SDK i potpuno podržano izvršno okruženje. Takođe, možete da koristite nove funkcije u EF Core 9 jer će nastaviti da cilja na .NET 8.
4. **Od februara do oktobra 2025:** Kao opciju, instalirajte .NET 10 probne verzije da biste istražili nove funkcije C# jezika i .NET biblioteke. Počnite da planirate primenu bilo koje nove funkcije biblioteke i ASP.NET Core funkcije u .NET 9 i .NET 10 na vaše .NET 8 projekte, kada budete spremni za migraciju.
5. **Od novembra 2025. nadalje:** Instalirajte .NET 10.0.100 SDK ili noviji i koristite ga za izgradnju projekata koji ciljaju na .NET 8 i koriste C# 14 kompajler. Možete migrirati svoje .NET 8 projekte na .NET 10 jer je to verzija sa **dugoročnom podrškom (LTS)**. Rok da završite migraciju je novembar 2026. kada .NET 8 dostiže kraj radnog ciklusa.



Slika 1.4: Ciljate na .NET 8 za dugoročnu podršku dok koristite najnovije C# kompajlere

Kada odlučujete da instalirate .NET SDK, zapamtite da se najnoviji podrazumevano koristi za izgradnju svih .NET projekata. Jednom kada instalirate .NET 9 SDK probnu verziju, ona će se koristiti podrazumevano za sve projekte, osim ako ne forsirate upotrebu starije, potpuno podržane SDK verzije, kao što je 8.0.100 ili kasnija zakrpa.

Da biste iskoristili prednosti bilo koje nove funkcije koja je dostupna u C# 13, dok i dalje ciljate na .NET 8 za dugoročnu podršku, izmenite svoju datoteku projekta, kao što je istaknuto u sledećem kodu:

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net8.0</TargetFramework>
    <LangVersion>13</LangVersion> <!--Zahteva .NET 9 SDK GA-->
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
  </PropertyGroup>
</Project>
```

Dobra praksa

Koristite SDK verziju koja je **generalno dostupna (GA)** kao što je .NET 9, da biste koristili nove funkcije kompajlera dok i dalje ciljate na starije verzije .NET okruženja sa dužom podrškom, kao što je .NET 8.

Vežbanje i istraživanje

Proverite svoje znanje i razumevanje odgovaranjem na neka pitanja, praktičnim vežbama i dubljim istraživanjem tema iz ovog poglavlja.

Vežba 1.1 – Materijal dostupan isključivo na internetu

Uverite se da ste spremni da koristite više uređivača koda. Prvo poglavlje knjige *C# 12 i .NET 8 – Osnovi modernog razvoja na više platformi* ima onlajn delove koji prikazuju kako da započnete više projekata pomoću različitih uređivača koda, uključujući Visual Studio, Code ili Rider.

Možete pročitati te delove na sledećoj adresi: <https://github.com/markjprice/cs12dot-net8/blob/main/docs/code-editors/README.md>.

Vežba 1.2 – Praktične vežbe

Ako još uvek nemate OpenAI ChatGPT nalog, registrujte se i pokušajte da mu postavite neka pitanja: <https://chat.openai.com/>.

Vežba 1.3 – Testirajte svoje znanje

Da biste dobili najbolji odgovor na neka od ovih pitanja, biće potrebno da sami istražujete. Želim da „razmišljate van okvira knjige“, pa namerno nisam obezbedio sve odgovore u knjizi.

Želim da kod vas podstaknem dobru naviku da tražite pomoć i na drugim mestima, po principu „nauci ga da peca“.

1. Kako se gleda na .NET programere koji odluče da koriste Rider, u odnosu na programere koji koriste samo Visual Studio?
2. Koja je ključna fraza prilikom inženjeringa upita sa ChatGPT jezičkim modelom?
3. Kada postavljate pitanje na forumu ili Discord kanalu, šta treba imati na umu?

4. Nakon objavljivanja u novembru 2024. godine, preuzimate i instalirate .NET 9 SDK. Kako ćete konfigurisati svoje projekte da i dalje ciljaju na .NET 8 za dugoročnu podršku, a istovremeno uživati u prednostima C# 13 kompajlera?
5. Za šta služi GitHub spremište za ovu knjigu?

Napomena

Dodatak sa odgovorima na pitanja za proveru znanja možete preuzeti na sledećoj adresi:
<https://packt.link/isUsj>.

Vežba 1.4 – Istražite teme

Koristite linkove na sledećoj stranici da biste saznali više o temama obrađenim u ovom poglavlju:
<https://github.com/markjprice/tools-skills-net8/blob/main/docs/book-links.md#chapter-1---introducing-tools-and-skills-for-net>.

Rezime

U ovom poglavlju:

- Upoznali ste alate i veštine o kojima ćete učiti u ovoj knjizi.
- Podesili ste svoje razvojno okruženje.
- Naučili ste gde da potražite pomoć i kako da dobijete najbolju pomoć od alata veštačke inteligencije, kao što je ChatGPT.
- Podesili ste uzorak baze podataka pod imenom Northwind za SQL Server, kao i neke .NET projekte za definisanje EF Core modela sa testovima za upotrebu u narednim poglavljima.

U sledećem poglavlju ćete naučiti da maksimalno iskoristite alate u svom uređivaču koda ili integrisanom razvojnom okruženju.

Pridružite se Discord prostoru naše knjige

Čitajte ovu knjigu zajedno sa drugim korisnicima, i sa autorom.

Postavljajte pitanja, rešavajte probleme drugih čitalaca, razgovarajte sa autorom tokom *Ask Me Anything* sesija, i još mnogo toga.

<https://packt.link/TS1e>

