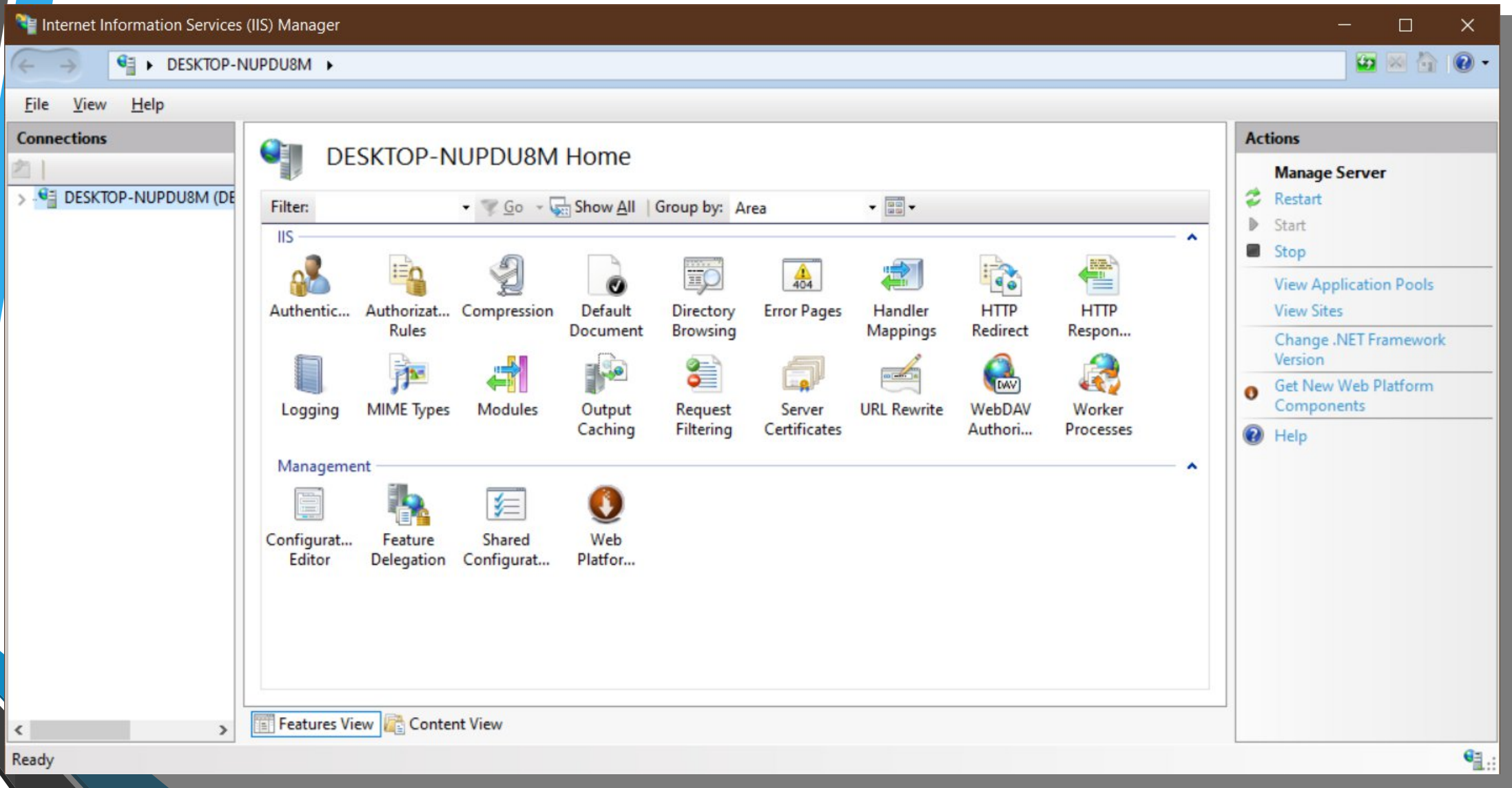




# Web programiranje

## IIS konfiguracija

# Internet Information Services (IIS) Manager





# IIS Manager

- Konfiguracija IIS Servera putem IIS Manager-a je jednostavna
- Hostovanje ASP.NET Core aplikacije je slično ASP.NET aplikaciji
- Ali pre toga, neophodno je instalirati neophodne komponente

# IIS Server i Manager

- Instalacija se vrši iz dijaloga Turn Windows features on or off (Control Panel -> Programs -> Programs And Features -> sa leve strane postoji opcija Turn Windows Features on or off)
- U listi je moguće naći opciju Internet Information Services
- Podešavanja su prikazana na slici, na sledećem slajdu

## Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ☒ Device Lockdown
- ☐ Guarded Host
- ☒ Hyper-V
- ☒ Internet Explorer 11
- ☒ Internet Information Services
  - ☐ FTP Server
    - ☐ FTP Extensibility
    - ☐ FTP Service
  - ☒ Web Management Tools
    - ☐ IIS 6 Management Compatibility
    - ☒ IIS Management Console
    - ☐ IIS Management Scripts and Tools
    - ☐ IIS Management Service
  - ☒ World Wide Web Services
    - ☐ Application Development Features
    - ☒ Common HTTP Features
    - ☒ Health and Diagnostics
      - ☐ Custom Logging
      - ☒ HTTP Logging
      - ☐ Logging Tools
      - ☐ ODBC Logging
      - ☐ Request Monitor
      - ☐ Tracing
    - ☒ Performance Features
      - ☐ Dynamic Content Compression
      - ☒ Static Content Compression
    - ☒ Security
      - ☐ Basic Authentication
      - ☐ Centralized SSL Certificate Support
      - ☐ Client Certificate Mapping Authentication
      - ☐ Digest Authentication
      - ☐ IIS Client Certificate Mapping Authentication
      - ☐ IP Security
      - ☒ Request Filtering
      - ☒ URL Authorization
      - ☒ Windows Authentication
- ☒ Internet Information Services Hostable Web Core
- ☐ Legacy Components
- ☒ Media Features

OK

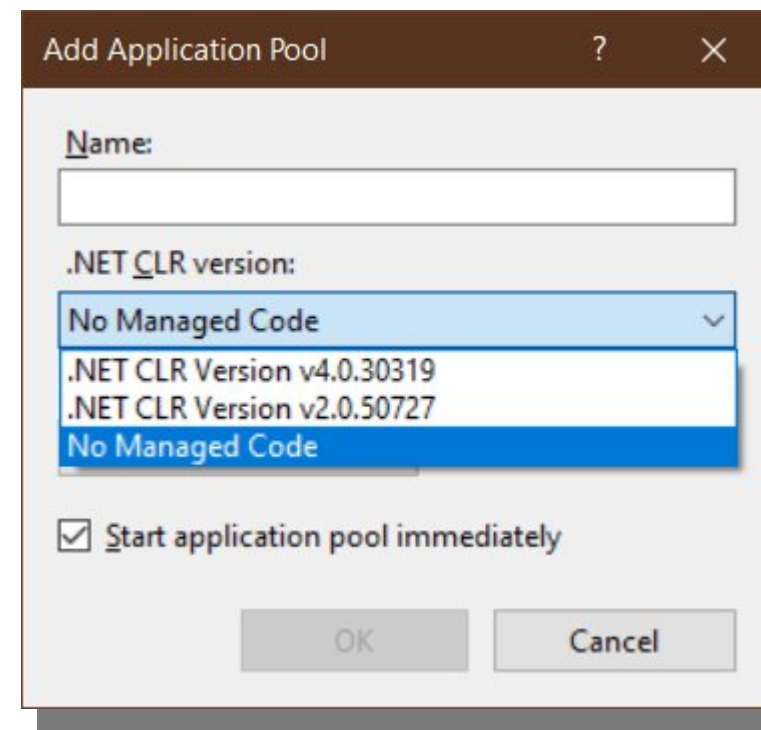
Cancel

# .NET Core Runtime & Hosting Bundle for Windows

- Sledeća komponenta koju je neophodno instalirati je .NET Core Runtime & Hosting Bundle for Windows:
  - <https://dotnet.microsoft.com/permalink/dotnetcore-current-windows-runtime-bundle-installer>
- Više informacija:
  - <https://learn.microsoft.com/en-us/aspnet/core/host-and-deploy/iis/hosting-bundle>

# .NET Core Runtime & Hosting Bundle

- Instalacijom .NET Core Runtime & Hosting Bundle-a, IIS server treba da se podesi da runtime .NET Core-a koristi direktno preko njega, a ne preko .NET Framework-a
  - Za računar, otvoriti Application Pools i kreirati novi (Add Application Pool) sa odgovarajućim nazivom i bez .NET CLR verzije (No Managed Code)



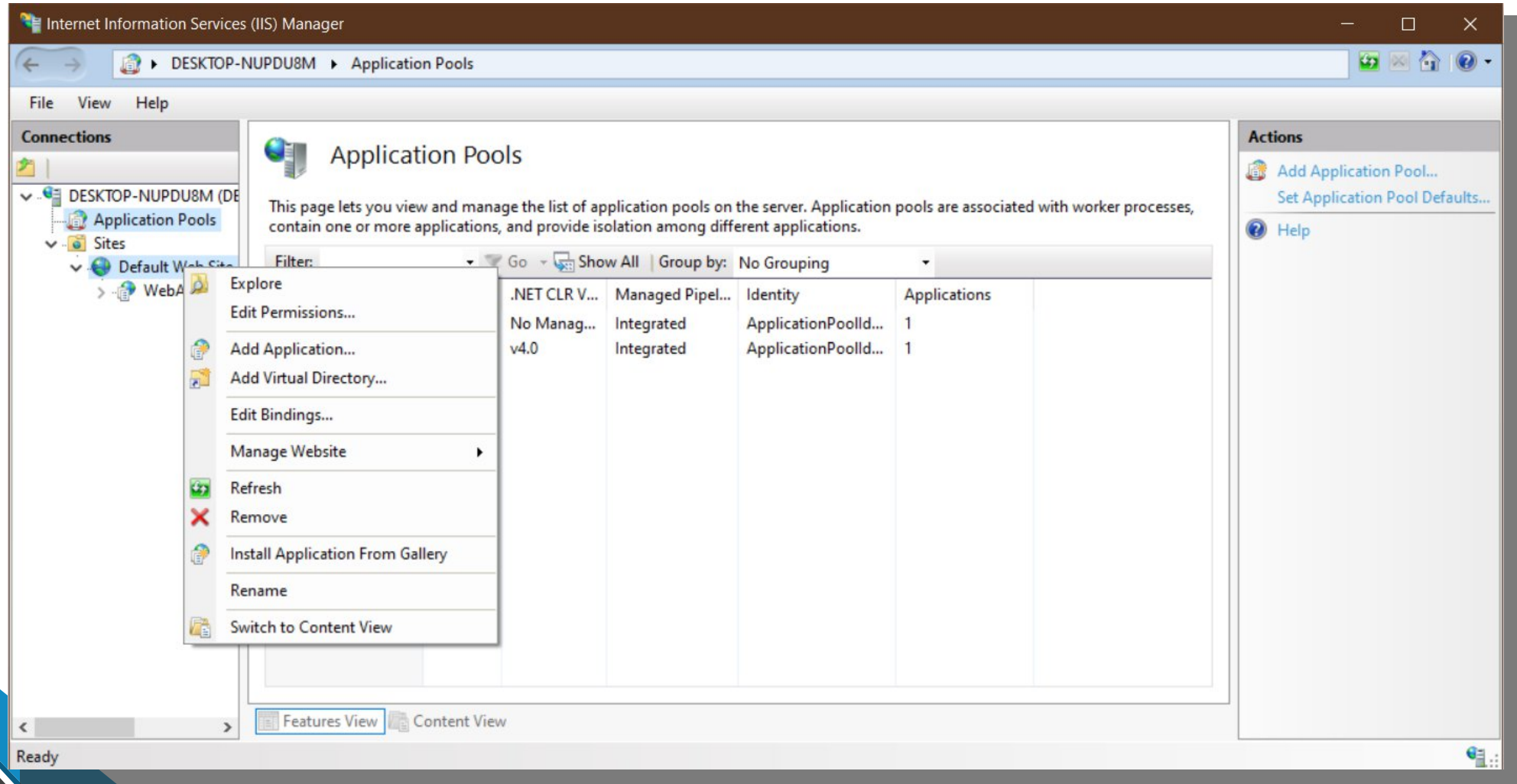
# URL Rewrite

- Pre samog kreiranja aplikacije, ukoliko ona zahteva URL Rewrite modul (naše dosadašnje aplikacije ne zahtevaju, ali čest je slučaj da je potreban), treba ga instalirati iz Web Platform Installer-a ili direktno sa:
  - <https://www.iis.net/downloads/microsoft/url-rewrite>



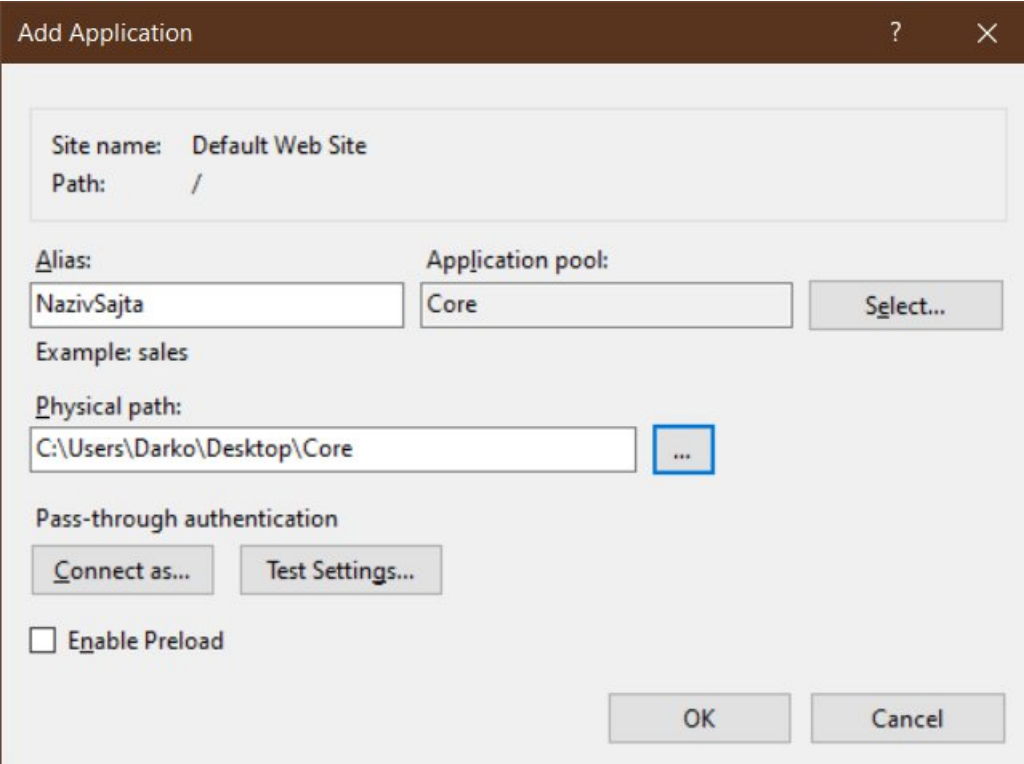
# Kreiranje aplikacije unutar IIS Servera

- Aplikacija se kreira za zadati sajt (sajt može da bude bilo koji i na bilo kojem portu, za zadati server) -> Add Application...



# Kreiranje aplikacije

- Podešavanja koja treba koristiti se mogu videti na slici:
  - Alias je naziv poddomena na kome će sajt da se nalazi
  - Application pool treba da bude malopre kreirani
  - Physical path je putanja do publish direktorijuma aplikacije

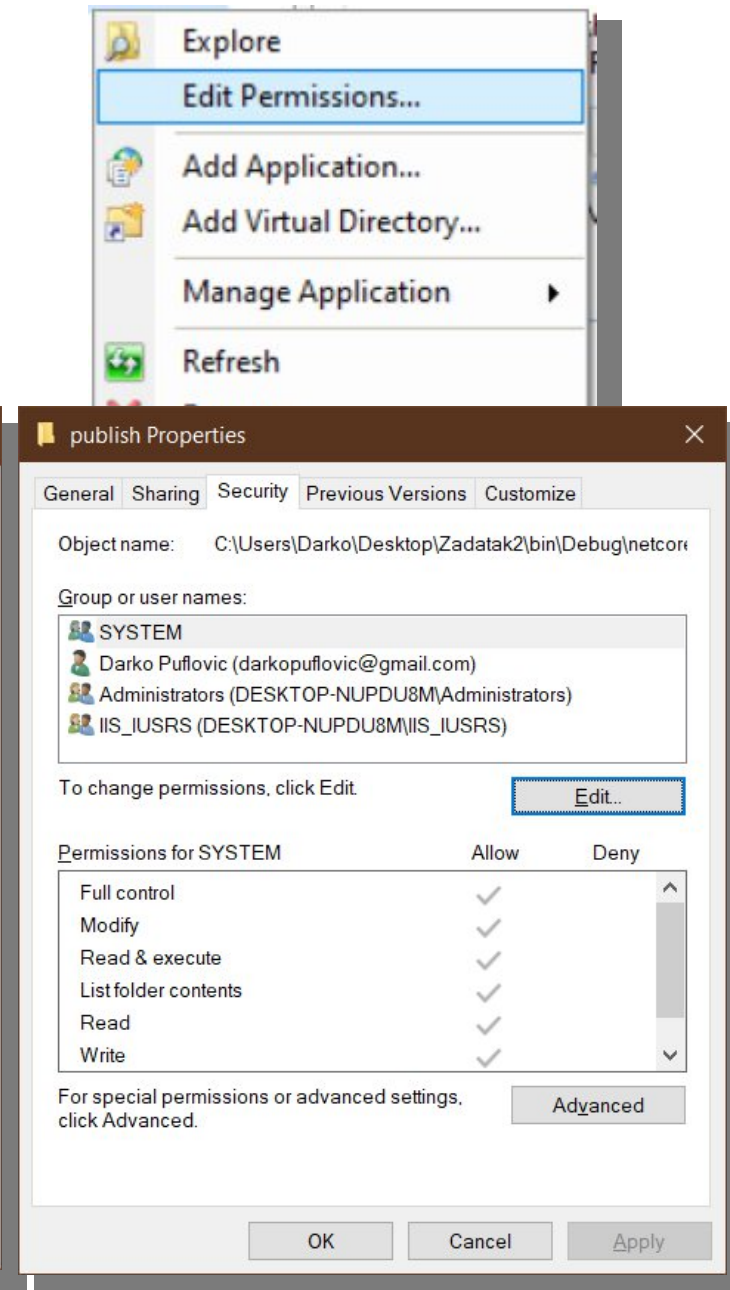
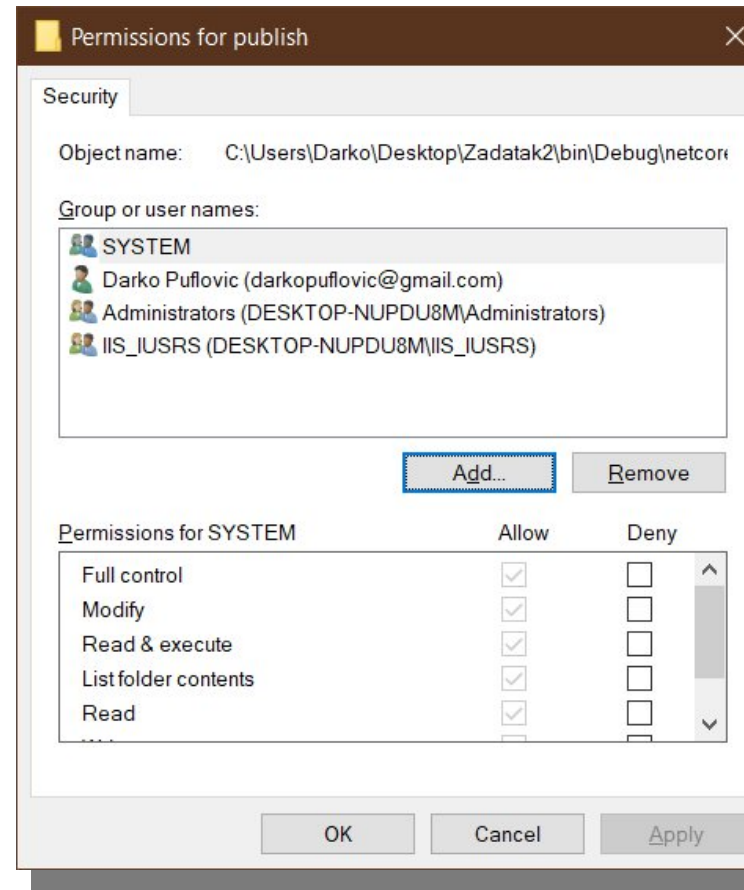


The screenshot shows the 'Add Application' dialog box in IIS Manager. The dialog has a title bar with a question mark and a close button. The main content area contains the following fields and controls:

- Site name:** Default Web Site
- Path:** /
- Alias:** NazivSajta
- Application pool:** Core
- Example:** sales
- Physical path:** C:\Users\Darko\Desktop\Core
- Pass-through authentication:** Connect as... Test Settings...
- Enable Preload:** ☐
- Buttons:** Select..., OK, Cancel

# Kreiranje aplikacije

- IIS server mora da ima privilegije pristupa određenom folderu, gde se fajlovi nalaze, pa je neophodno iz menija selektovati Edit Permissions... otići u Security tab i tamo dodati (Edit -> Add...) IIS\_IUSRS korisnika sa privilegijama za čitanje



# Kreiranje aplikacije

- Aplikacija bi nakon ovog koraka trebalo da bude funkcionalna
- Postoje neki problemi koji mogu da se jave:
  - Za pristup bazi, takođe su potrebne privilegije:
  - To može da se reši na sledeći način

# Podešavanje privilegija u bazi

- `CREATE LOGIN [userName] WITH PASSWORD = '[password]';`
- `CREATE USER [userName] FOR LOGIN [userName];`
- `EXEC sp_addrolemember 'db_owner', '[password]';`
- Gde [userName] i [password] treba zameniti odgovarajućim korisničkim imenom i šifrom

# Podešavanje instance LocalDB baze podataka

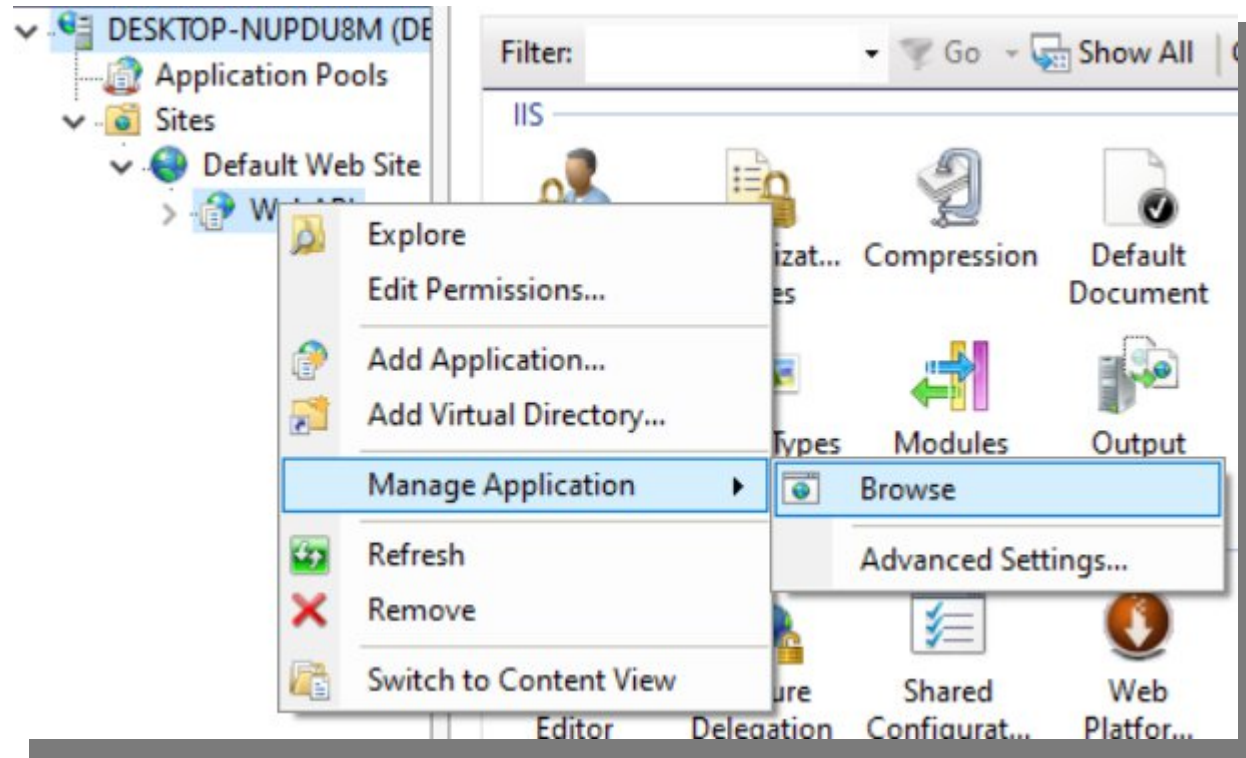
- LocalDB baza mora da bude vidljiva od strane IIS Servera, pa zato mora da se izvrši sledeća komanda iz Command Prompt-a (administrator):
  - sqllocaldb share [Instanca LocalDB baze] IIS\_DB
- Na ovaj način se kreira nova instanca baze (iste baze) koja se zove IIS\_DB
- U Connection String-u je takođe potrebno izmeniti ova podešavanja, tako da Connection String postaje:
  - Server=(localdb)\\.\IIS\_DB;Database=model;User Id=[userName];Password=[password];

# Kreiranje aplikacije

- Još jedan problem koji može da se javi je pozivanje odgovarajućeg Controller handler-a iz JavaScript-a
- Putanja do aplikacije više nije localhost:port/[Controller]/[Metoda]...
- Sada postoji i poddomen na kome se nalazi aplikacija, tako da, ukoliko ne znamo njegov naziv, nemoguće je pozvati ga direktno iz JavaScript-a kompletnim URL-om, već na sledeći način:
  - **fetch(„Faculty/GetFacutly/3“)**, a ne:
  - *fetch(„/Faculty/GetFacutly/3“)*, zato što bi takav pristup pokušao da pristupi adresi: localhost/Faculty..., što ne mora da bude realna putanja.

# Pozivanje aplikacije

- Nakon što je aplikacija kreirana, moguće joj je putem browser-a pristupiti isto kao da se radi i o pokretanju iz Visual Studio-a (IIS Express) ili CMD-a (Kastrel) serveri ili kao na slici.







# Nginx konfiguracija na Linux-u

(Ubuntu, Debian...)

# Instalacija nginx servera na Ubuntu Linux-u

- Instalacija se vrši na sledeći način:
  - `sudo apt-get update`
  - `sudo apt-get install nginx`
- Da bi se pokretao automatski, potrebno je ukucati sledeću komandu:
  - `sudo /etc/init.d/nginx start`

# Nginx

- Ovako instaliran nginx server je dovoljan za izvršavanje statičkih web stranica
- Da bi ste izvršavali .NET Core aplikaciju, neophodno je da se koristi kombinacija nginx-a i Kastrel servera i njegove konfiguracije

# .NET Core & nginx

- Pre svega, neophodno je da je instaliran .NET Core na Ubuntu Linux-u
- Instrukcije su dostupne na:
  - <https://dotnet.microsoft.com/en-us/download?initial-os=linux>
- Tu se nalaze instrukcije za aktuelnu verziju, na Linux OS, ostale Linux distribucije su takođe podržane, moguće je na stranici selektovati odgovarajuću i izabrati verziju .NET Core Framework-a

# .NET Core & nginx

- Nakon kreiranja aplikacije (dotnet new web...), dopisivanja odgovarajuće funkcionalnosti i testiranja, potrebno je kreirati njenu publish verziju
  - dotnet publish --configuration Release
- Ova komanda kreira u bin\Debug direktorijumu i publish direktorijum, sa svim datotekama i direktorijumima potrebnim za izvršenje aplikacije

# .NET Core & nginx

- Proxy server:

- Podešavanje se vrši u Program.cs na sledeći način:

```
app.UseForwardedHeaders(new ForwardedHeadersOptions  
{  
    ForwardedHeaders = ForwardedHeaders.XForwardedFor |  
    ForwardedHeaders.XForwardedProto  
});
```

```
app.UseAuthentication();
```

# .NET Core & nginx

- Takođe je potrebno dodati i sledeći kod:

```
services.Configure<ForwardedHeadersOptions>(options =>
{
    options.KnownProxies.Add(IPAddress.Parse(„[adresa]")); // npr: 127.0.0.1
});
```

# Pokretanje nginx servera

- Pokretanje servera se vrši sledećom komandom:
  - `sudo service nginx start`
- Nakon toga je neophodno izvršiti konfiguraciju tako da se .NET Core aplikacija izvršava preko nginx servera:
  - `/etc/nginx/sites-available/default` - fajl za konfiguraciju



# Konfiguracija nginx

- server {
- **listen** **80;** // port na kome server radi (ovaj deo ne treba pisati, samo komentar)
- server\_name example.com \*.example.com;
- location / {
- proxy\_pass <http://localhost:5000>; // putanja na kojoj se izvršava Kastrel app
- proxy\_http\_version 1.1;
- proxy\_set\_header Upgrade \$http\_upgrade;
- proxy\_set\_header Connection keep-alive;
- proxy\_set\_header Host \$host;
- proxy\_cache\_bypass \$http\_upgrade;
- proxy\_set\_header X-Forwarded-For \$proxy\_add\_x\_forwarded\_for;
- proxy\_set\_header X-Forwarded-Proto \$scheme;
- }
- }

# Nginx

- Nakon što je konfiguracioni fajl izmenjen, potrebno je proveriti da li je konfiguracioni fajl validan i učitati promene:
  - `sudo nginx -t` - ako je sve u redu
  - `sudo nginx -s reload` - onda reload

# Kreiranje servisa za aplikaciju

- `sudo mcedit /etc/systemd/system/kestrel-helloapp.service`
  - Kreira fajl u `/etc/systemd/system/` direktorijumu sa nazivom `kestrel-helloapp.service`
- Sadržaj fajla treba da bude nalik na:

- [Unit]
- Description=.NET Web App working on Ubuntu!!!
- [Service]
- WorkingDirectory=/var/www/helloapp // putanja do aplikacije (direktorijum)
- ExecStart=/usr/bin/dotnet /var/www/helloapp/helloapp.dll
- // /usr/bin/dotnet je putanja do dotnet aplikacije, dok je druga (.dll) do aplikacije
- Restart=always // Restartuje se u slučaju greške i to na 10 sekundi (ispod)
- # Restart service after 10 seconds if the dotnet service crashes:
- RestartSec=10
- KillSignal=SIGINT
- SyslogIdentifier=dotnet-example
- User=www-data // Korisnik koji se ovde koristi, mora da bude kreiran i da mu se dodele privilegije
- Environment=ASPNETCORE\_ENVIRONMENT=Production // Production aplikacija
- Environment=DOTNET\_PRINT\_TELEMETRY\_MESSAGE=false
- [Install]
- WantedBy=multi-user.target

# Pokretanje servisa

- Kreiranje servisa se vrši na sledeći način:
  - `sudo systemctl enable kestrel-helloapp.service`
- Dok se startuje sledećom komandom:
  - `sudo systemctl start kestrel-helloapp.service`
- A provera statusa sledećom:
  - `sudo systemctl status kestrel-helloapp.service`

# Korisni linkovi

- <https://docs.microsoft.com/en-us/aspnet/core/host-and-deploy/linux-nginx>
- <https://docs.microsoft.com/en-us/aspnet/core/host-and-deploy/proxy-load-balancer>
- <https://medium.com/@benmorel/creating-a-linux-service-with-systemd-611b5c8b91d6>
- <https://linuxize.com/post/how-to-create-a-sudo-user-on-ubuntu/>
- <https://help.ubuntu.com/stable/ubuntu-help/user-add.html.en>
- <https://help.ubuntu.com/lts/serverguide/user-management.html#user-profile-security>