

# IIS (Internet Information Services)

**IIS (Internet Information Services)** je webový server vyvinutý společností Microsoft pro operační systémy Windows Server a Windows. Jedná se o robustní platformu pro hostování webových stránek, webových aplikací a služeb.

## Základní charakteristika

IIS je integrovanou součástí Windows a poskytuje kompletní prostředí pro provoz webových aplikací:

- **Webový server** - obsluha HTTP/HTTPS požadavků
- **FTP server** - přenos souborů
- **SMTP server** - odesílání emailů (starší verze)
- **Aplikační server** - spouštění [ASP.NET](#) a dalších aplikací
- **Management nástroje** - grafické rozhraní pro správu

## Historie a verze

IIS prošel vývojem od jednoduchého webového serveru k komplexní platformě:

Verze	Rok	Windows	Klíčové vlastnosti
IIS 1.0	1995	Windows NT 3.51	První verze
IIS 3.0	1996	Windows NT 4.0	ASP (Active Server Pages)
IIS 5.0	2000	Windows 2000	HTTP compression
IIS 6.0	2003	Windows Server 2003	Application pools, worker processes
IIS 7.0	2006	Windows Server 2008, Vista	Modulární architektura
IIS 7.5	2009	Windows Server 2008 R2, Windows 7	WebDAV, FTP 7.5
IIS 8.0	2012	Windows Server 2012, Windows 8	SNI, WebSockets
IIS 8.5	2013	Windows Server 2012 R2, Windows 8.1	Dynamic IP restrictions
IIS 10.0	2015	Windows Server 2016/2019/2022, Windows 10/11	HTTP/2, Nano Server

## Architektura IIS

Moderní IIS (verze 7+) používá modulární architekturu:

### Hlavní komponenty

- **HTTP.sys** - kernel-mode driver pro zpracování HTTP požadavků
- **W3SVC (WWW Service)** - Windows služba řídící IIS
- **WAS (Windows Process Activation Service)** - správa worker procesů
- **Worker Processes (w3wp.exe)** - procesy pro běh aplikací
- **Application Pools** - izolované prostředí pro aplikace

## Request Processing Pipeline

Zpracování požadavku prochází několika fázemi:

1. **HTTP.sys** přijme požadavek na úrovni kernelu
2. Požadavek je směrován do správného **Application Pool**
3. **Worker process** (w3wp.exe) zpracovává požadavek
4. **HTTP modules** zpracovávají různé aspekty (autentizace, komprese atd.)
5. **Handler** generuje odpověď (ASP.NET, PHP, statický soubor)
6. Odpověď je vrácena klientovi přes HTTP.sys

## Application Pools

**Application Pools** jsou klíčovou vlastností pro izolaci aplikací:

- Každý pool běží ve vlastním worker procesu
- Pád jedné aplikace neovlivní ostatní
- Možnost nastavit různé identity a oprávnění
- Recycling – automatický restart pro obnovu zdrojů

```
<!-- applicationHost.config -->
<applicationPools>
  <add name="DefaultAppPool"
        managedRuntimeVersion="v4.0"
        managedPipelineMode="Integrated">
    <processModel identityType="ApplicationPoolIdentity" />
    <recycling>
      <periodicRestart time="1.22:00:00" />
    </recycling>
  </add>
</applicationPools>
```

Nastavení Application Pool:

- **Pipeline Mode** – Integrated (doporučeno) nebo Classic
- **.NET CLR Version** – verze [.NET runtime](#)
- **Identity** – účet pro běh aplikace
- **Recycling** – pravidla pro restart procesu
- **Resource limits** – omezení CPU a paměti

## Websites a Virtual Directories

### Website

Základní jednotka v IIS:

- Unikátní kombinace **IP adresy, portu a hostname**

- Každý web má svůj **fyzický adresář** (Document Root)
- Může obsahovat více **virtuálních adresářů** a **aplikací**
- Přiřazen k jednomu **Application Pool**

## Virtual Directory

Mapování URL cesty na fyzický adresář:

- URL: ``http://domena.cz/obrazky/`` → Disk: ``D:\Media\Images\``
- Umožňuje organizovat obsah mimo hlavní adresář webu
- Nedefinuje novou aplikaci

## Application

Samostatná aplikace v rámci webu:

- Má vlastní konfiguraci (web.config)
- Může běžet v jiném Application Pool
- Má vlastní aplikační doménu

## Bindings

**Bindings** definují, jak je web dostupný:

```
<site name="MojeStranky" id="1">
  <bindings>
    <binding protocol="http" bindingInformation="*:80:www.domena.cz" />
    <binding protocol="https" bindingInformation="*:443:www.domena.cz" />
  />
  <binding protocol="http" bindingInformation="*:80:domena.cz" />
</bindings>
</site>
```

Typy vazeb:

- **HTTP** - standardní nešifrovaná komunikace (port 80)
- **HTTPS** - šifrovaná komunikace pomocí SSL/TLS (port 443)
- **FTP** - přenos souborů
- **Custom** - vlastní protokoly

## SSL/TLS certifikáty

IIS plně podporuje šifrované HTTPS připojení:

- **Server certificates** - certifikáty pro identifikaci serveru
- **Client certificates** - autentizace klientů pomocí certifikátů

- **SNI (Server Name Indication)** – více SSL certifikátů na jedné IP
- **Centralized SSL** – sdílené úložiště certifikátů
- **Let's Encrypt** – integrace s automatickými certifikáty

Požadavky na certifikát:

- Vydán důvěryhodnou certifikační autoritou (CA)
- Platný pro dané doménové jméno
- Nainstalován v Certificate Store ve Windows

## Autentizace

IIS podporuje různé metody autentizace:

- **Anonymous** – žádná autentizace (výchozí)
- **Basic** – username/password v plaintext (nutné HTTPS)
- **Digest** – hash-based autentizace
- **Windows Authentication** – integrace s Active Directory (NTLM, Kerberos)
- **Forms Authentication** – vlastní přihlašovací formuláře ([ASP.NET](#))
- **Client Certificate** – autentizace pomocí certifikátů

```
<!-- web.config -->
<system.webServer>
  <security>
    <authentication>
      <anonymousAuthentication enabled="false" />
      <windowsAuthentication enabled="true" />
    </authentication>
  </security>
</system.webServer>
```

## Autorizace a oprávnění

Řízení přístupu k obsahu:

- **NTFS permissions** – oprávnění na úrovni souborového systému
- **URL Authorization Rules** – pravidla na základě URL
- **IP Address Restrictions** – povolení/blokování IP adres
- **Request Filtering** – filtrování nebezpečných požadavků

## HTTP Modules

**Moduly** rozšiřují funkcionalitu IIS:

Vestavěné moduly:

- **StaticFileModule** – obsluha statických souborů

- **DirectoryListingModule** - výpis obsahu adresářů
- **DefaultDocumentModule** - výchozí dokumenty (index.html)
- **HttpLoggingModule** - logování požadavků
- **CompressionModule** - komprese odpovědí (gzip, brotli)
- **UrlRewriteModule** - přepisování URL (vyžaduje instalaci)
- **CustomErrorModule** - vlastní chybové stránky
- **AuthenticationModules** - různé typy autentizace

Moduly lze přidávat a odebírat podle potřeby:

```
<system.webServer>
  <modules>
    <remove name="WebDAVModule" />
    <add name="CustomModule" type="MyApp.CustomModule" />
  </modules>
</system.webServer>
```

## HTTP Handlers

**Handlers** generují odpovědi pro specifické typy požadavků:

- **StaticFileHandler** - obrázky, CSS, JavaScript
- **ASPClassic** - klasické ASP stránky (.asp)
- **PageHandlerFactory** - [ASP.NET Web Forms](#) (.aspx)
- **SimpleHandlerFactory** - ASP.NET HTTP Handlers (.ashx)
- **WebServiceHandlerFactory** - ASMX webové služby
- **ExtensionlessUrlHandler** - ASP.NET MVC/Web API
- **PHPHandler** - PHP skripty (vyžaduje instalaci PHP)

## URL Rewrite

**URL Rewrite Module** umožňuje přepisování a přesměrování URL:

```
<system.webServer>
  <rewrite>
    <rules>
      <!-- Přesměrování na HTTPS -->
      <rule name="HTTP to HTTPS redirect" stopProcessing="true">
        <match url="(.*)" />
        <conditions>
          <add input="{HTTPS}" pattern="off" />
        </conditions>
        <action type="Redirect" url="https://{HTTP_HOST}/{R:1}"
          redirectType="Permanent" />
      </rule>

      <!-- Odstranění www -->
      <rule name="Remove WWW" stopProcessing="true">
```

```
<match url="(.*)" />
<conditions>
  <add input="{HTTP_HOST}" pattern="^www\.(.+)$" />
</conditions>
<action type="Redirect" url="https://{C:1}/{R:0}"
  redirectType="Permanent" />
</rule>
</rules>
</rewrite>
</system.webServer>
```

Možnosti URL Rewrite:

- **Redirect** - přesměrování na jinou URL (301, 302)
- **Rewrite** - změna URL bez přesměrování
- **Custom Response** - vlastní HTTP odpověď
- **Abort Request** - zamítnutí požadavku

## Request Filtering

Ochrana před útoky a nebezpečnými požadavky:

```
<system.webServer>
  <security>
    <requestFiltering>
      <requestLimits maxAllowedContentLength="52428800"
        maxUrl="4096"
        maxQueryString="2048" />
      <fileExtensions>
        <add fileExtension=".config" allowed="false" />
        <add fileExtension=".cs" allowed="false" />
      </fileExtensions>
      <verbs>
        <add verb="TRACE" allowed="false" />
      </verbs>
      <hiddenSegments>
        <add segment="App_Code" />
        <add segment="bin" />
      </hiddenSegments>
    </requestFiltering>
  </security>
</system.webServer>
```

## Kompresie

IIS podporuje kompresi pro snížení přenesených dat:

- **Static compression** - komprese statických souborů (uloženo v cache)

- **Dynamic compression** - komprese dynamického obsahu (ASP.NET)
- **Gzip** - široce podporovaný formát
- **Brotli** - modernější, lepší komprese (IIS 10+)

```
<system.webServer>
  <httpCompression directory="%SystemDrive%\inetpub\temp\IIS Temporary
Compressed Files">
    <scheme name="gzip" dll="%Windir%\system32\inetsrv\gzip.dll" />
    <staticTypes>
      <add mimeType="text/*" enabled="true" />
      <add mimeType="application/javascript" enabled="true" />
      <add mimeType="application/json" enabled="true" />
    </staticTypes>
    <dynamicTypes>
      <add mimeType="text/*" enabled="true" />
      <add mimeType="application/json" enabled="true" />
    </dynamicTypes>
  </httpCompression>
  <urlCompression doStaticCompression="true" doDynamicCompression="true"
/>
</system.webServer>
```

## Caching

IIS nabízí několik úrovní cachování:

- **Kernel-mode cache** - cache v HTTP.sys (nejrychlejší)
- **User-mode cache** - cache ve worker procesu
- **Output cache** - cache pro ASP.NET (web.config nebo kód)

```
<system.webServer>
  <caching>
    <profiles>
      <add extension=".jpg" policy="CacheUntilChange"
kernelCachePolicy="CacheUntilChange" />
      <add extension=".css" policy="CacheUntilChange"
kernelCachePolicy="CacheUntilChange" />
    </profiles>
  </caching>
  <staticContent>
    <clientCache cacheControlMode="UseMaxAge"
cacheControlMaxAge="7.00:00:00" />
  </staticContent>
</system.webServer>
```

## Logging

IIS zaznamenává požadavky do log souborů:

Formáty logů:

- **W3C Extended Log** – nejpoužívanější, konfigurovatelný
- **IIS Log** – Microsoft formát
- **NCSA Common Log** – kompatibilní s Apache
- **Custom** – vlastní formát

Umístění logů (výchozí):

- `C:\inetpub\logs\LogFiles\W3SVC1\` (číslo = ID webu)

Log obsahuje:

- Datum a čas požadavku
- IP adresu klienta
- HTTP metodu (GET, POST)
- URL a query string
- HTTP status kód
- User agent
- Referrer
- Bytes přeneseny

## Failed Request Tracing

**FREB (Failed Request Event Buffering)** pro diagnostiku problémů:

- Detailní záznamy o požadavcích, které selžou nebo trvají dlouho
- Zachycení všech modulů a handlerů v pipeline
- XML soubory s detailními informacemi
- XSL pro vizualizaci v prohlížeči

Použití:

1. Povolit Failed Request Tracing pro web
2. Definovat pravidla (status kód, časový limit)
3. Analyzovat vygenerované XML logy

## Application Initialization

Předehrání aplikací pro rychlejší první požadavek:

```
<system.webServer>
  <applicationInitialization
    doAppInitAfterRestart="true"
    skipManagedModules="false">
    <add initializationPage="/" />
    <add initializationPage="/api/warmup" />
  </applicationInitialization>
```

```
</system.webServer>
```

V Application Pool:

- Start Mode: **AlwaysRunning**
- Preload Enabled: **True**

## WebSockets

IIS 8.0+ podporuje WebSocket protokol:

- Obousměrná komunikace přes jedno TCP spojení
- Nízká latence pro real-time aplikace
- Integrace s [ASP.NET SignalR](#)

```
<system.webServer>  
  <websocket enabled="true" />  
</system.webServer>
```

## HTTP/2

IIS 10.0+ podporuje HTTP/2 protokol:

- Vyžaduje Windows Server 2016+ nebo Windows 10+
- Automaticky povoleno pro HTTPS připojení
- Multiplexing více požadavků přes jedno spojení
- Server push (experimentální)
- Header compression

## Správa IIS

### IIS Manager

Grafické rozhraní pro správu:

- **GUI aplikace** - spustitelná z nabídky Start nebo `inetmgr`
- **Hierarchická struktura** - server → websites → applications
- **Feature View** - konfigurace jednotlivých funkcí
- **Content View** - procházení souborů webu
- **Remote Management** - vzdálená správa serverů

### PowerShell

Správa pomocí PowerShellu:

```
# Import IIS modulu
Import-Module WebAdministration

# Vytvoření nového webu
New-Website -Name "MujWeb" `
    -PhysicalPath "C:\inetpub\wwwroot\mujweb" `
    -Port 80 `
    -HostHeader "www.mujweb.cz"

# Spuštění/zastavení webu
Start-Website -Name "MujWeb"
Stop-Website -Name "MujWeb"

# Vytvoření Application Pool
New-WebAppPool -Name "MujAppPool"
Set-ItemProperty IIS:\AppPools\MujAppPool `
    -Name managedRuntimeVersion `
    -Value "v4.0"

# Restart Application Pool
Restart-WebAppPool -Name "MujAppPool"

# Získání seznamu webů
Get-Website

# Nastavení binding
New-WebBinding -Name "MujWeb" `
    -Protocol "https" `
    -Port 443 `
    -HostHeader "www.mujweb.cz" `
    -SslFlags 1
```

## AppCmd.exe

Příkazový řádek pro správu IIS:

```
REM Výpis všech webů
appcmd list site

REM Vytvoření webu
appcmd add site /name:MujWeb /physicalPath:C:\inetpub\wwwroot\mujweb
/bindings:http/*:80:www.mujweb.cz

REM Zastavení webu
appcmd stop site "MujWeb"

REM Výpis Application Pools
appcmd list apppool
```

```
REM Recyklace Application Pool  
appcmd recycle apppool "DefaultAppPool"
```

```
REM Export konfigurace  
appcmd add backup "ZalohaKonfigurace"
```

```
REM Obnovení konfigurace  
appcmd restore backup "ZalohaKonfigurace"
```

## Konfigurace

IIS používá hierarchický konfigurační systém:

### applicationHost.config

Hlavní konfigurační soubor pro celý server:

- Umístění: `%SystemRoot%\System32\inetsrv\config\applicationHost.config`
- Definiuje globální nastavení IIS
- Application Pools
- Websites a jejich bindings
- Moduly a handlers

### web.config

Konfigurační soubor pro konkrétní aplikaci:

- Umístěn v kořenovém adresáři webu/aplikace
- Dědí nastavení z applicationHost.config
- Může být v každém podadresáři (kaskádová konfigurace)
- Může být upravován bez administrátorských práv

```
<?xml version="1.0" encoding="UTF-8"?>  
<configuration>  
  <system.webServer>  
    <defaultDocument>  
      <files>  
        <clear />  
        <add value="index.html" />  
        <add value="index.htm" />  
        <add value="default.aspx" />  
      </files>  
    </defaultDocument>  
  
    <httpErrors errorMode="Custom">  
      <remove statusCode="404" />  
      <error statusCode="404" path="/404.html" responseMode="File" />  
    </httpErrors>  
  </system.webServer>  
</configuration>
```

```
</httpErrors>

<httpProtocol>
  <customHeaders>
    <add name="X-Frame-Options" value="SAMEORIGIN" />
    <add name="X-Content-Type-Options" value="nosniff" />
  </customHeaders>
</httpProtocol>
</system.webServer>
</configuration>
```

## Výkon a optimalizace

Tipy pro optimalizaci výkonu IIS:

- **Application Pool recycling** - nastavit rozumné intervaly
- **Output caching** - cache statického i dynamického obsahu
- **Compression** - povolit gzip/brotli kompresi
- **Static content cache** - nastavit cache headers
- **Connection limits** - omezit současná spojení
- **Request queue** - nastavit vhodnou velikost fronty
- **Disable unused modules** - odebrat nepotřebné moduly
- **Kernel-mode cache** - využít cache na úrovni kernelu
- **CDN** - použít pro statický obsah

## Bezpečnost

Doporučení pro zabezpečení IIS:

- **Aktualizace** - pravidelně instalovat Windows Update
- **Least privilege** - minimální oprávnění pro Application Pool identity
- **HTTPS** - vždy používat pro citlivá data
- **Request filtering** - blokovat nebezpečné požadavky
- **Custom errors** - nezobrazovat detailní chyby v produkci
- **Directory browsing** - zakázat výpis adresářů
- **Remove headers** - skrýt verzi serveru
- **IP restrictions** - omezit přístup k admin rozhraním
- **Strong passwords** - silná hesla pro účty
- **Firewall** - povolit jen potřebné porty

```
<!-- Odstranění server headers -->
<system.webServer>
  <httpProtocol>
    <customHeaders>
      <remove name="X-Powered-By" />
    </customHeaders>
  </httpProtocol>
<security>
```

```
<requestFiltering removeServerHeader="true" />
</security>
</system.webServer>
```

## Load Balancing

IIS podporuje několik způsobů distribuce zátěže:

- **ARR (Application Request Routing)** – reverzní proxy a load balancer od Microsoftu
- **NLB (Network Load Balancing)** – Windows funkce pro distribuci síťového provozu
- **Hardware load balancers** – dedikované zařízení (F5, Citrix NetScaler)
- **Cloud load balancers** – Azure Load Balancer, AWS ELB

## Web Farm

**Web Farm** – skupina IIS serverů:

- Sdílená konfigurace – synchronizace web.config a applicationHost.config
- Sdílený obsah – UNC cesta nebo DFS
- Centralizovaný SSL – sdílené certifikáty
- Shared Configuration – exportovaná konfigurace na síťové cestě

## Integrace s ASP.NET

IIS a [ASP.NET](#) úzce spolupracují:

- **Integrated Pipeline Mode** – ASP.NET moduly integrované do IIS pipeline (doporučeno)
- **Classic Pipeline Mode** – starý model, ASP.NET oddělen od IIS (kompatibilita)

V Integrated módu:

- ASP.NET moduly vidí všechny požadavky (i statické soubory)
- Lepší výkon a flexibilita
- Jednotný konfigurační model

## Monitoring a diagnostika

Nástroje pro monitorování IIS:

- **Performance Monitor** – čítače výkonu (% CPU, Requests/sec, Connection Count)
- **IIS Logs** – analýza požadavků
- **Failed Request Tracing** – diagnostika problémů
- **Event Viewer** – systémové události a chyby
- **Process Explorer** – detailní informace o procesech
- **PerfView** – profilování výkonu

- **Application Insights** – monitoring v Azure

Důležité čítače:

- `Web Service\Current Connections`
- `ASP.NET\Requests Queued`
- `Process(w3wp)\% Processor Time`
- `Process(w3wp)\Private Bytes`

## Porovnání s jinými webovými servery

Vlastnost	IIS	Apache	Nginx
Platforma	Windows	Multi-platform	Multi-platform
Licence	Proprietární (součást Windows)	Open-source	Open-source
Konfigurace	GUI + XML	Textové soubory	Textové soubory
.NET podpora	Nativní	Pomocí Mono	Reverzní proxy
PHP podpora	FastCGI	Mod_php, FastCGI	FastCGI
Výkon	Velmi dobrý	Dobrý	Vynikající
Správa paměti	Application Pools	MPM moduly	Asynchronní

## Instalace IIS

### Windows Server

```
# PowerShell - instalace role Web Server
Install-WindowsFeature -Name Web-Server -IncludeManagementTools

# Instalace ASP.NET 4.x
Install-WindowsFeature -Name Web-Asp-Net45

# Instalace URL Rewrite
# Stáhnout z: https://www.iis.net/downloads/microsoft/url-rewrite
```

### Windows 10/11

1. Ovládací panely → Programy → Zapnout nebo vypnout funkce systému Windows
2. Zaškrtnout **Internetová informační služba**
3. Vybrat potřebné komponenty (World Wide Web Services, ASP.NET atd.)
4. OK a počkat na instalaci

## Výhody a nevýhody

### Výhody

- **Integrace s Windows** – hluboká integrace s OS a Active Directory

- **Bezpečnost** – pravidelné bezpečnostní aktualizace
- **ASP.NET optimalizace** – nejlepší platforma pro ASP.NET aplikace
- **GUI správa** – přehledné grafické rozhraní
- **Application Pools** – izolace aplikací
- **Enterprise podpora** – podpora od Microsoftu
- **Stabilita** – osvědčený v enterprise prostředí

## Nevýhody

- **Pouze Windows** – nelze použít na Linuxu
- **Licence** – vyžaduje Windows Server licenci
- **Uzavřený** – není open-source
- **Komplexita** – může být složitý pro začátečníky
- **Závislost** – úzká vazba na Windows ekosystém

## Související pojmy

- [ASP.NET](#) – webový framework pro IIS
- [.NET Framework](#) – běhové prostředí
- [Windows Server](#) – serverový operační systém
- [Active Directory](#) – adresářová služba
- [Kestrel](#) – webový server pro ASP.NET Core
- [Nginx](#) – alternativní webový server
- [Apache](#) – alternativní webový server

## Externí odkazy

- [Oficiální stránky IIS](#)
- [Microsoft dokumentace k IIS](#)
- [IIS fórum](#)
- [Ke stažení pro IIS](#) (moduly, extensions)

From:

<http://serviceit.cz/> - IT ENCYKLOPEDIE

Permanent link:

<http://serviceit.cz/doku.php?id=iis>

Last update: **2026/01/05 11:16**

