

Hálózatkezelés

Tóth Zsolt

Miskolci Egyetem

2013

Tartalomjegyzék

- 1 Hálózati Alapismeretek
- 2 System.Net Namespace
- 3 Socket Kezelés
- 4 Példa

ISO–OSI Modell

- 1 Fizikai réteg
- 2 Adatkapcsolati réteg
- 3 Hálózati réteg
- 4 Szállítási réteg
- 5 Viszonylati réteg
- 6 Megjelenítési réteg
- 7 Alkalmazási réteg

Internet Protokoll

- 1 Ethernet
- 2 MAC címek
- 3 Internet (IP címek)
- 4 TCP, UDP
- 5 Alkalmazási réteg
 - ▶ DNS, HTTP, FTP, SSH, SMTP, IMAP, POP3

1 Fizikai réteg

- ▶ Átviteli közeg (Ethernet kábel, Rádió hullám, stb.)

2 Adatkapcsolati réteg

- ▶ Media Access Control címek
- ▶ Egyediség
- ▶ Gyári beállítás

5 Alkalmazás réteg

3 Internet réteg

- ▶ IP Címek
- ▶ 4 byte
- ▶ 193.6.5.42 ✓
- ▶ 127.0.0.1 ✓
- ▶ 243.256.42.314 X

4 Szállítási réteg

- ▶ TCP kapcsolat alapú
- ▶ UDP datagram

https://193.6.5.42/svn/CSharp
[protokoll://]host[:port]/[path]

- Hálózati erőforrás azonosítója
- Általános használat
- port elhagyható → default port
- path a host-on belüli elérési út

- Néhány protokoll:

Protokoll	Port
http	80
ftp	21
jdbc	1521
file	N/A

- Multipurpose Internet Mail Extensions
- Formátum jelzés
- SMTP, HTTP
- Típusok
 - ▶ text/plain
 - ▶ text/html
 - ▶ text/css
 - ▶ audio/mp3
 - ▶ video/mpeg

Tartalomjegyzék

1 Hálózati Alapismeretek

2 System.Net Namespace

3 Socket Kezelés

4 Példa

- Egyszerű programozói felület
 - További névterek
 - ▶ Configuration
 - ▶ Http
 - ▶ Mail
 - ▶ Mime
 - ▶ PeerToPeer
 - ▶ Security
 - ▶ **Socket**
 - ▶ WebSocket
- ## Osztályok
- Cookie
 - Dns
 - *EndPoint*
 - IPAddress
 - IPEndPoint
 - WebClient
 - ...

- Domain Name System
- IP cím \iff domain név
- static osztály
- static metódusok gyűjteménye

Metódusok

- `GetHostAddresses(string)` : `IPAddress[]`
- `GetHostName()` : `string`

- IP címet reprezentál
- Properties
 - ▶ AddressFamily
 - ▶ IsIPv4MappedToIPv6
 - ▶ ...
- Adattagok
 - ▶ Any
 - ▶ Broadcast
 - ▶ Loopback

Metódusok

- Equals
- MapToIPv4
- MapToIPv6
- Parse
- ToString

System.Net.IPEndPoint

- IPAddress + port
- Properties
 - ▶ Address
 - ▶ AddressFamily
 - ▶ Port
- IPEndPoint(IPAddress,Int32)

- Egyszerű küldés/fogadás
- URI alapján
- WebClient()
- Properties
 - ▶ BaseAddress
 - ▶ Encoding
 - ▶ Headers

Metódusok

- DownloadData
- DownloadFile
- DownloadString
- OpenRead
- OpenWrite
- UploadData

Tartalomjegyzék

1 Hálózati Alapismertetek

2 System.Net Namespace

3 Socket Kezelés

4 Példa

- Végpont
 - Hálózaton keresztül
 - IPC mechanizmus
 - Elterjedt
 - Socket API
 - Berkeley Socket
 - POSIX Socket
- Kliens–Szerver alapú
 - Szerver
 - ▶ Host-on fut
 - ▶ Porton figyel
 - ▶ Klientet vár
 - ▶ Üzenetet küld/fogad
 - Kliens
 - ▶ Szerverhez kapcsolódik
 - ▶ Üzenetet küld/fogad

TcpListener

- Socket
- TCP porton figyel
- TcpListener(port : IPEndPoint)
- Szinkron/Aszinkron működés
- AcceptTcpClien() : TcpClient
- Start()
- Stop()

```
IPAddress host =  
    IPAddress.Parse(  
        "127.0.0.1");  
Int32 port = 9876;  
TcpListener server = new  
    TcpListener(host,  
        port);
```

TcpClient

- Kliens
- TCP kapcsolat
- Szerverhez kapcsolódik
- Szinkron/Aszinkron működés
- Connect()
- Close()
- GetStream()

```
TcpClient client = new  
    TcpClient("127.0.0.1",  
            9876);
```


NetworkStream

- public class NetworkStream : Stream
 - Read()
 - Write()
 - Flush()
 - Close()
- ```
NetworkStream ns =
 client.GetStream();
byte[] buffer = new
 byte[256];
while ((ns.Read(buffer,
 0, buffer.Length))
 != 0)
{
 ...
}
ns.Close();
```

# Tartalomjegyzék

1 Hálózati Alapismertetek

2 System.Net Namespace

3 Socket Kezelés

4 Példa

# WebClient

```
WebClient client = new WebClient();
Stream data =
client.OpenRead("http://www.iit.uni-miskolc.hu");
StreamReader sr = new StreamReader(data);
string content = sr.ReadToEnd();
Console.WriteLine(content);
data.Close();
sr.Close();
```

- Az oldal HTML kódja