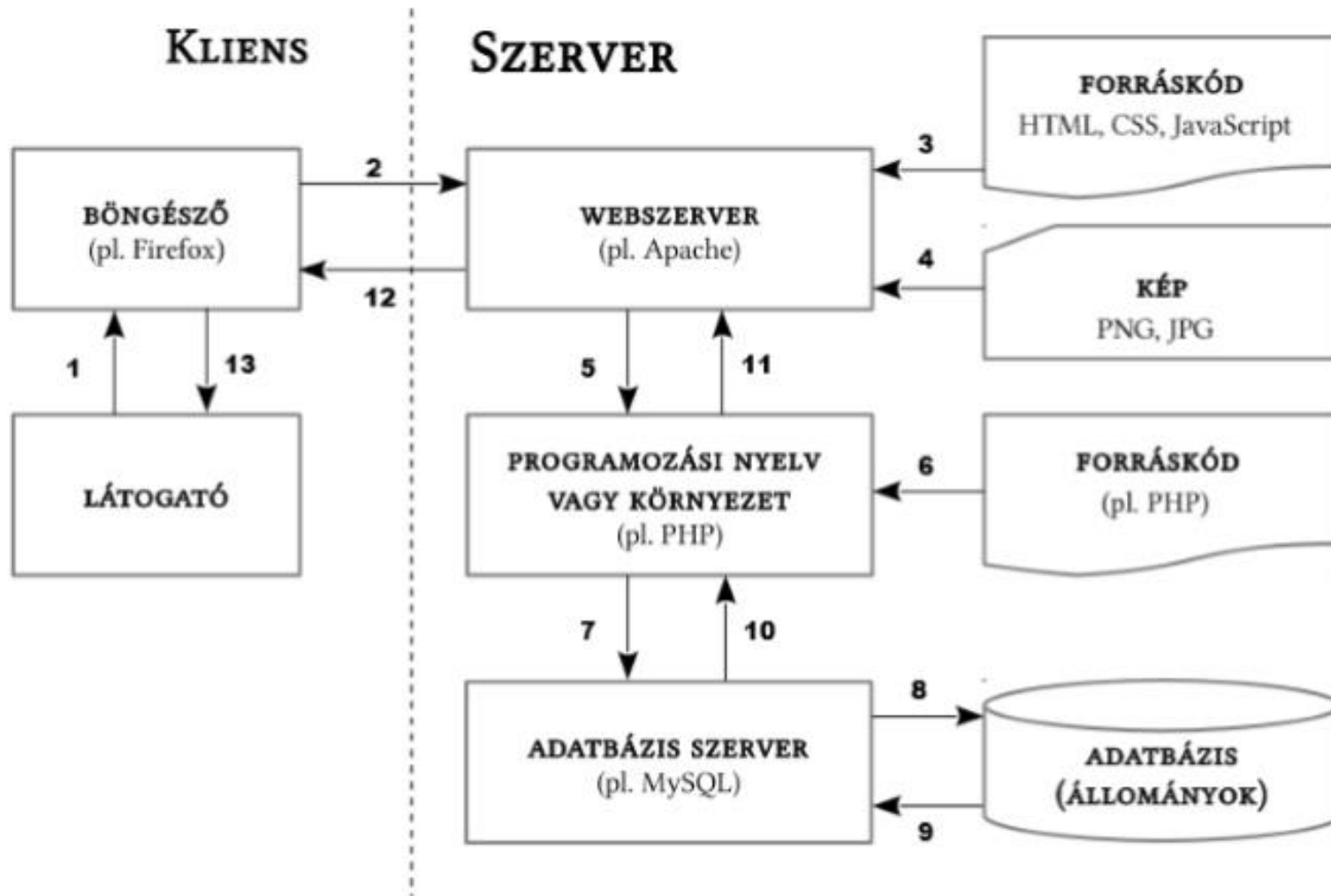


Web elmélet I.

A web működése: kliens-szerver felépítés



Statikus tartalomkiszolgálás

- ▶ A felhasználó a böngészőt használva kezdeményezheti egyes weboldalak letöltését.
- ▶ A web kezdeti időszakában a webszerver azokat az állományokat tudta kiszolgáltatni, amiket a háttértárain elhelyeztek. Ez tulajdonképpen statikus tartalmat eredményez, vagyis **az ilyen tartalom nem változik, akárhányszor is töltjük le.**
- ▶ Bizonyos esetekben ez ma is így van: például egy honlapba illesztett kép nem fog megváltozni, akárhányszor töltjük is le, ezért a webszervernek a böngésző kérésére válaszul mindössze vissza kell azt adni.

Dinamikus tartalomkiszolgálás

- ▶ Később egyre nagyobb igény lett a dinamikus tartalmak iránt, amikor a tartalom már a látogató tevékenységei, vagy más okok miatt színesebb, változóbb lehet. Ebben az esetben a webszerver **nem önmaga válaszol** a böngésző kérésére, hanem PHP, vagy más nyelvű program állítja elő a választ, amit a webszerver csak közvetít.
- ▶ Tovább növelheti az oldal dinamizmusát, ha a tartalmak előállításához szükséges adatokat (legalább részben) **adatbázisban tároljuk**. Ekkor a PHP nyelvű forrásprogram az adatbázis-szerverrel kapcsolatot épít fel, és adatbázisból származó információkat is felhasznál a válasz elkészítéséhez, illetve a felhasználók válaszait is eltárolja az adatbázisban.

Webszerver

- ▶ A **webkiszolgáló/webszerver** egy kiszolgáló, amely elérhetővé teszi a rajta tárolt weblapokat a **HTTP protokollon** keresztül. A web szerverekhez **web böngészőkkel** lehet kapcsolódni.
- ▶ Bár a webszerverek sok mindenben különböznek, az alapvető funkcióik azonosak. Minden webszerver **HTTP kéréseket fogad** a hálózatról, és **HTTP válaszokat küld** vissza. A HTTP válasz az esetek többségében egy HTML dokumentum, de lehet még egyszerű szöveges fájl, kép, vagy más típusú fájl is.

Kérés

- ▶ A webserverek a kientől kapott kérésekben többek között URL címet kapnak, melyet aztán kétféleképpen értelmezhet a szerver a beállításaitól függően:
 1. A tartománynév után álló relatív mappa és fájl struktúrát **hozzárendelik egy gyökérmappához**. (A gyökérmappa a webservér beállításáiban van megadva, és az adatokat kérő kliens számára láthatatlan.)
 2. A tartománynév után álló relatív mappa és fájlstruktúra (vagy akár még a tartománynév is) teljesen**független** a kért címben szereplő struktúrától. Ebben az esetben szerver meghatározott szabályok szerint formázza a kért címet. Ennek segítségével egy mappára irányuló kérés teljesen más mappára vagy akár egy fájlra is mutathat és fordítva.
- ▶ A kliens például az alábbi URL-t kéri: <http://www.pelda.com/utvonal/cikk.html>
- ▶ A kliens webböngészője ezt értelmezve létrehoz egy kapcsolatot a www.pelda.com kiszolgálóval, és elküldi a következő HTTP 1.1 kérést:

```
GET /utvonal/cikk.html HTTP 1.1  
Host: www.pelda.com
```

```
...
```

Válasz

1. A www.pelda.com címet **megfelelteti** a webszerver az adott gyökérmappához (pl. /var/www/pelda), amelyhez hozzáfűzi a /utvonal/cikk.html elérést – ezzel megtörtént a megfeleltetés a fájlrendszer erőforráshoz. A kért eredmény a szerveren tehát: /var/www/pelda/utvonal/cikk.html.

Ezt követően a webszerver ellenőrzi, hogy kiszolgálható-e az adott kérés, ill. hogy létezik-e. Ha nem létezik, akkor 404-es hibakóddal tér vissza. Ha hozzáférhető, akkor beolvassa, elvégzi rajta az esetleges további műveleteket, majd elküldi a kliensnek. A válasz természetesen szintén magában foglalja a megfelelő fejléceket.

HTTP/1.1 200 OK

Date: Sun, 08 Feb xxxx 01:11:12 GMT

Server: Apache/1.3.29 (Win32)

Last-Modified: Sat, 07 Feb xxxx

ETag: "0-23-4024c3a5"

Accept-Ranges: bytes

Content-Length: 35

Connection: close

Content-Type: text/html

<h1>My Home page</h1>

Állapot

Válasz
fejléc

Üres sor

Válasz törzs

2. A második megoldás esetében, az erőforrásokhoz történő megfeleltetés előtt a címet átformázza.

```
www.pelda.com/toplista/kutyak+es+macskak
```

URL kérést a következőképpen alakíthatja át:

```
/var/www/pelda/toplista.php?cim=kutyak+es+macskak
```


Webtárhelyek

- ▶ A mai weboldalak kis hányada igényli, hogy egy vagy esetleg több (ún. dedikált) szerver teljes egészében a weboldal kiszolgálását végezze. Éppen ezért a legtöbb honlap más honlapokkal osztozik egy webtárhely erőforrásain. Ezt nevezzük **osztott webtárhelynek**.
- ▶ Shared hosting
 - ▶ Előny: Megfizethető, könnyű elkezdeni
 - ▶ Hátrány: Nem befolyásolható a szerver és a teljesítmény
- ▶ VPS hosting (virtuális)
 - ▶ Előny: root szerver hozzáférés, biztonságos környezetben
 - ▶ Hátrány: Sokkal drágább a share hostingtól.
- ▶ Dedicate hosting
 - ▶ Előny: Maximális kezelhetőség, jó szerver teljesítmény
 - ▶ Hátrány: drága, képzett személyzetszükséges hozzá
- ▶ Cloude hosting (felhő)
 - ▶ Előny: szerver skálázhatósága, költséghatékony
 - ▶ Hátrány: haladó informatikai ismereteket kíván, nem stabil (vitatható)

Webtárhely megoldások

- ▶ A webtárhely szolgáltatás általában tartalmaz egy adminisztrációs felületet (pl. [cPanel](#)), hogy a bérlő a tárhelyét menedzselni tudja.
- ▶ Osztott tárhelyszolgáltatók rendszerint az egyes szolgáltatásokat fizikailag elkülönített kiszolgáló rendszereken oldják meg, az ügyfélszolgáltató és adminisztrációs rendszer, a levelező kiszolgáló, az adatbázis szerver, a webszerver fizikailag elkülönített kiszolgálókon működik. A legtöbb webkiszolgáló alacsony költségű Linux vagy BSD alapú **LAMP szerver**. (A LAMP a Linux, Apache, MySQL, PHP szavak rövidítését jelenti.)
- ▶ Az egyes operációs rendszerekre épített szolgáltatások lényegében meghatározzák a felhasználó által elérhető technológiák csoportját is. Windows alapú webtárhely esetén a felhasználó választhat ASP.NET és Microsoft SQL Server, de akár PHP és MySQL Server támogatást is; míg LAMP szerver esetén csak PHP nyelvű weboldalakat készíthetünk MySQL Server támogatással.
- ▶ A **virtuális webtárhely** szolgáltatás (VPS) alatt egy olyan internetes szolgáltatást értünk, ahol egy webszerver erőforrásait több felhasználó/honlap között osztják fel. Minden felhasználó egy a rendszer által dedikált tárhelyet foglal el, aminek nyilvános tartalma egyedi domén néven érhető el. Kisebb forgalmú weboldalt költséghatékonyan lehet bérelt webtárhelyen üzemeltetni. Tárhelyet ún. tárhelyszolgáltatóktól bérelhetünk.


FUTURITY WEB
 WEB HOSTING & DESIGN
 WWW.FUTURITYWEBSERVICES.COM



CPANEL 11

Notices

News Flash!
Standard Hosting
Only R30.00

Find

Find functions quickly by typing here:

Frequently Accessed Areas

Advanced DNS Zone Editor
phpMyAdmin

Stats

Main Domain	example.co.za
Home Directory	/home/example
Last login from	74.202.255.243
Disk Space Usage	55.08 / 100 MB
Monthly Bandwidth Transfer	154.17 / 1024 MB

expand stats

Preferences

Getting Started Wizard
 Video Tutorials
 Change Password
 Update Contact Info
 Change Style
 Change Language
 Shortcuts

 RYSkin Theme Changer

Mail

Email Accounts
 Webmail
 BoxTrapper
 Spam Assassin™
 Forwarders
 Auto Responder
 Default Address

Mail Lists
 Account Level Filtering
 User Level Filtering
 Email Delivery Route
 Import Addresses/Forwarders
 Email Authentication
 MX Entry

Files

Backups
 Backup Wizard
 File Manager
 Legacy File Manager
 Web Disk
 Disk Space Usage
 FTP Accounts

FTP Session Control
 Anonymous FTP

Logs

Latest Visitors
 Bandwidth
 Webalizer
 Webalizer FTP
 Raw Access Logs
 Error Log
 Choose Log Programs

Google Analytics Integration
 Awstats

Security

Security

Password Protect Directories
 IP Deny Manager
 HotLink Protection
 GnuPG Keys

Domains

Subdomains
 Addon Domains
 Parked Domains
 Redirects
 Simple DNS Zone Editor
 Advanced DNS Zone Editor

Databases

MySQL MySQL® Databases
 MySQL MySQL® Database Wizard
 phpMyAdmin
 Remote MySQL

Software/Services

QuickInstall
 CGI Center
 BaseKit SiteBuilder
 Site Software
 Perl Modules
 PHP PEAR Packages
 PHP Configuration

RubyGems
 Ruby on Rails
 Optimize Website
 php.ini E2Config
 Fantastico De Luxe

Advanced

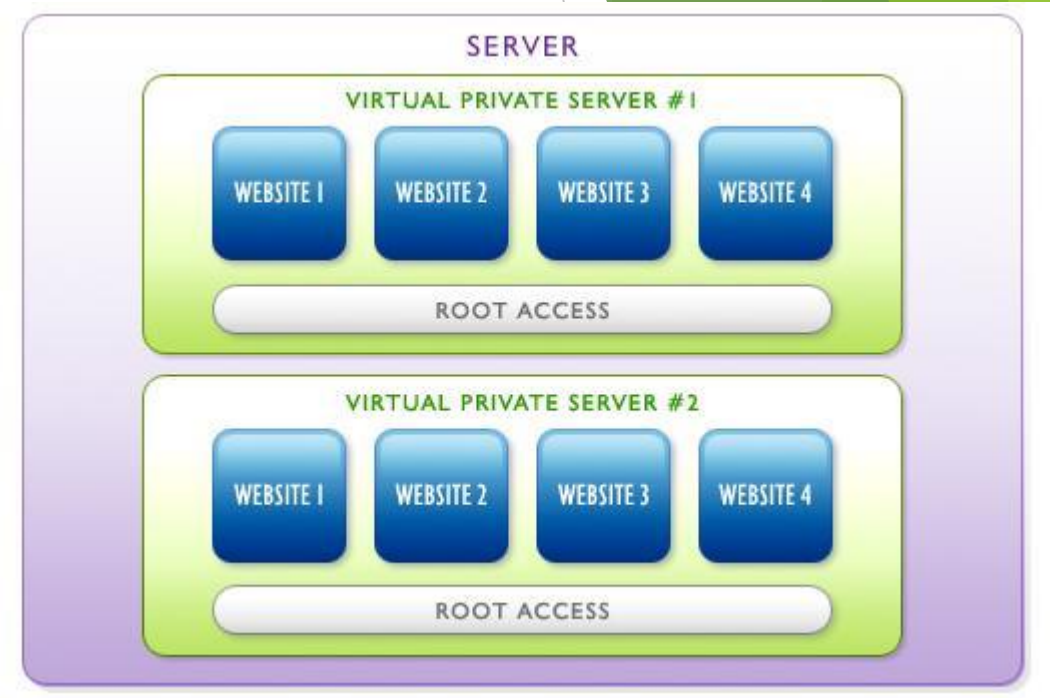
Apache Handlers
 Image Manager
 Index Manager
 Error pages
 Cron jobs
 Network Tools
 FrontPage® Extensions

Cloudflare
 Website Templates
 MIME Types

lite graphics | hide icons | reset all interface settings | reveal all boxes | reset box order

[Home](#) | [Trademarks](#) | [Help](#) | [Documentation](#) | [Contact](#) | [Logout](#)

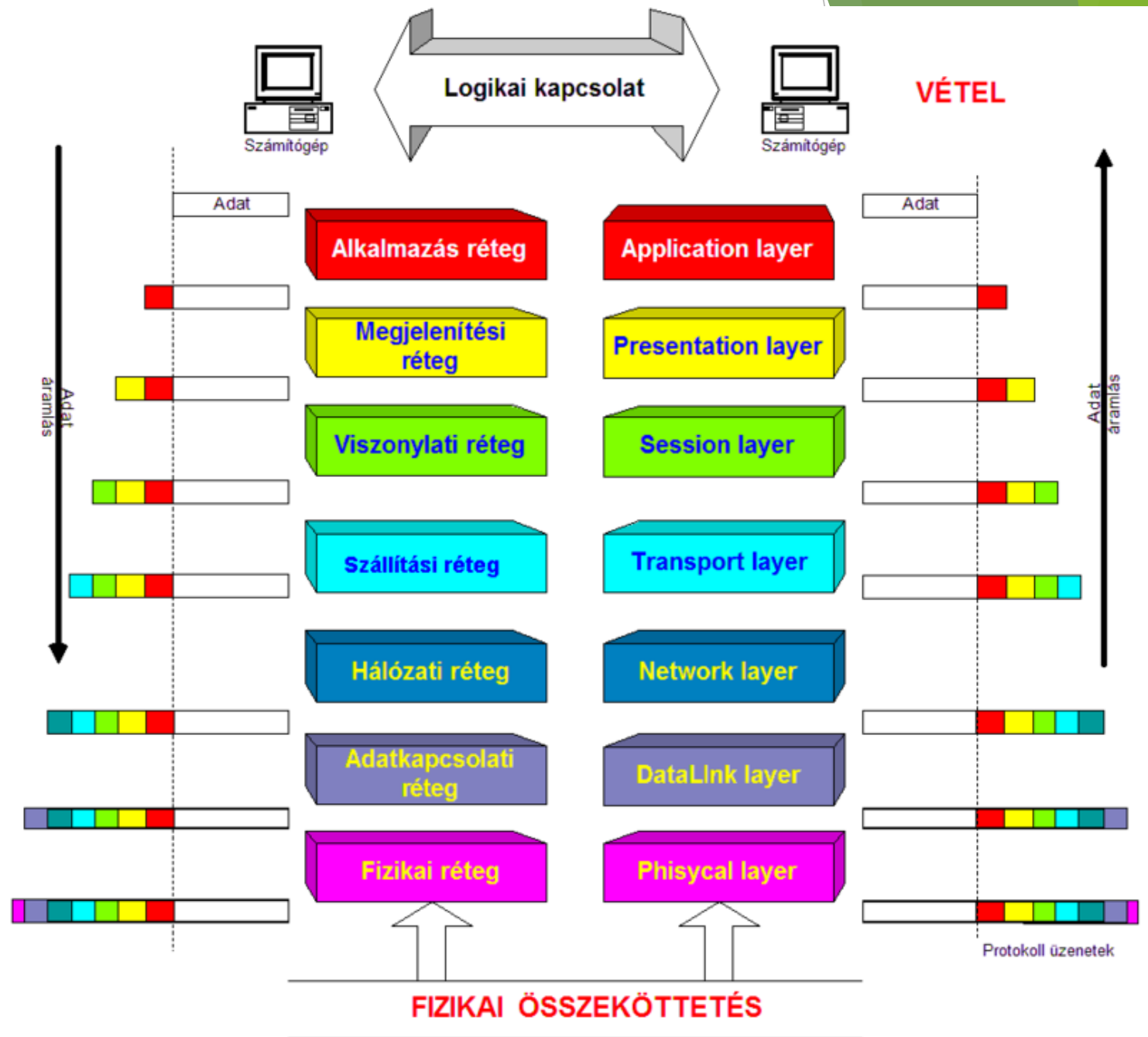
- ▶ A virtuális szerver, más néven VPS (Virtual Private Server) a valós szerver erőforrásainak egy bizonyos részét használhatja. Ezek az erőforrások lehetnek dedikáltak, azaz kizárólagosan az adott virtuális szerver veheti igénybe a neki kiosztott erőforrásokat, fizikai hardver elemeket, vagy megosztottak: pl. két virtuális szerver használhatja az egyik processzort, azaz osztoznak a teljesítményén, de lehetőségük van nagyobb kapacitás kihasználására is, amennyiben a másik fél kevésbé terheli.



HTTP protokoll

- ▶ A HTTP (HyperText Transfer Protocol) egy információátviteli protokoll a világhálón. Az eredeti célja a **HTML lapok publikálása és fogadása** volt.
- ▶ A HTTP egy kérés-válasz alapú protokoll kliensek és szerverek között. A kommunikációt mindig a kliens kezdeményezi. A HTTP klienseket gyűjtőnéven user agentnek is nevezik. A user agent jellemzően, de nem feltétlenül webböngésző.
- ▶ A HTTP általában a TCP/IP réteg felett helyezkedik el, de nem függ tőle. (OSI modell, Alkalmazói rétegben)

OSI modell



A következő ábra XHTML állomány esetén, az oldal teljes elérési folyamatát ábrázolja:



User clicks



Browser

```
GET http://weather.example.org/oaxaca HTTP/1.1
Host: weather.example.org
Accept: application/xhtml+xml
```

sends request

Server handles request



Web server for weather.example.org

```
HTTP/1.1 200 OK
Content-Length: 45178
Content-Type: application/xhtml+xml; charset=utf-8

<!DOCTYPE html PUBLIC "...
<html xmlns="http://www...
<head>
<title>5 day forecast ...
```

sends response

Server



Browser interprets representation,
displays presentation

Munkamenet (session)

- ▶ A HTTP egy állapot nélküli protokoll. Az állapot nélküli protokollok előnye, hogy a szervernek nem kell nyilvántartania felhasználói információkat az egyes kérések kiszolgálása között. A HTTP eredetileg nem arra készült, hogy felhasználók jelentkezzenek be rajta keresztül szerverekre és ott munkamenetet (session-t) indítsanak. Történetileg azonban úgy alakult, hogy a HTTP terjedt el széles körben más, felhasználói bejelentkezést támogató protokollok helyett, ami arra kényszerítette a webfejlesztőket, hogy kerülőutakon járva tárolják a felhasználók munkamenet-állapotait, ha arra szükség van.

URL

- ▶ A webcím, más néven URL (amely a Uniform Resource Locator, vagyis egységes erőforrás-azonosító rövidítése), az Interneten megtalálható bizonyos erőforrások (például szövegek, képek) szabványosított címe.
- ▶ Először Tim Berners-Lee alkotta meg a World Wide Weben való használatra. A jelenleg használt formátumot részletesen leírja az [IETF RFC 17381](#) szabványa.

Egyetlen címben összefoglalja a dokumentum megtalálásához szükséges négy alapvető információt:

1. a protokollt, amit a célgéppel való kommunikációhoz használunk;
2. a szóban forgó gép vagy tartomány nevét;
3. a hálózati port számát, amin az igényelt szolgáltatás elérhető a célgépen;
4. a fájlhoz vezető elérési utat a célgépen belül.

Egy tipikus, egyszerű webcím így néz ki:

```
http://hu.wikipedia.org:80/wiki
```

Ennek részei:

1. A http határozza meg a használandó protokollt. A protokoll neve után kettőspont (:) írandó.
2. A hu.wikipedia.org adja meg a célgép tartománynevét. Ez elé két perjel (//) írandó.
3. A 80 adja meg a célgép azon hálózati portszámát, amin kérésünket várja; ez elé kettőspont (:) írandó. Ezt a részt gyakran teljesen elhagyhatjuk, például esetünkben a HTTP protokoll alapértelmezett portszáma a 80.
4. A /wiki a kért elérési út a célgépen. Ez a rész mindig a per-jellel (/) kezdődik.

- ▶ A legtöbb böngésző nem is igényli, hogy a `http://` részt begépeljük egy weblap eléréséhez, hiszen az esetek döntő többségében úgyis ezt használjuk. Egyszerűen begépelhetjük a lap címét,
 - ▶ például: `hu.wikipedia.org/wiki/Bit`.
- ▶ A címlap megtekintéséhez általában elég a tartomány nevét beírni, például `hu.wikipedia.org`.

- ▶ A **webcímek egyéb részeket** is tartalmazhatnak, http esetében
 - ▶ például az elérési út után, egy kérdőjel (?) mögé helyezve keresési kérdés szerepelhet, ami egy get metódusú HTML űrlapból vagy speciális módon összeállított hiperhivatkozásból származik.
 - ▶ Az elérési út után, attól egy kettős kereszttel (#) elválasztva szerepelhet a hiperszöveg egy részére hivatkozó azonosító. Ez az azonosító nem része a webcímnek, de gyakran szerepel vele kapcsolatban.

Példák:

```
http://hu.wikipedia.org/w/wiki.phtml?title=Bit&action=history
```

```
http://hu.wikipedia.org/wiki/1999#Események
```

- ▶ A webcím az URI(Uniform Resource Identifier: Egy az Interneten, a helyi hálózaton vagy a helyi gépen elhelyezkedő erőforrást azonosító hivatkozás.) egy olyan fajtája, ahol az azonosítás a dokumentum helye alapján történik.
- ▶ Ha egy weblapot egyértelműen meghatároz egy webcím, akkor rá mutató hivatkozást hozhatunk létre.
- ▶ Ez a helyzet nem mindig áll fenn, pl. egy menüpont megváltoztathatja a lapon belül az egyik keret tartalmát anélkül, hogy ennek az új kombinációnak külön webcíme lenne. Ezen kívül egy weblap függhet ideiglenesen tárolt információtól is. Még ha van is egy weblapnak vagy keretnek önálló webcíme, ez nem mindig nyilvánvaló annak a számára, aki rá mutató hivatkozást kíván létrehozni.
- ▶ Egy keret webcíme nem látszik a címsávban, és létrehozható címsáv nélküli ablak is. A webcím ilyenkor a lap forrása, illetve egyes részeinek „tulajdonsága” alapján meghatározható.
- ▶ Azon kívül, hogy rá mutató hivatkozást hozzunk létre, egyéb okokból is kíváncsiak lehetünk egy lap webcímére: ha tartalmát önállóan akarjuk megjeleníteni, illetve, ha bizonyos megszorításokat (eszköztár nélküli, vagy kicsi, méretezhetetlen ablak) meg akarunk kerülni.

HTTPS

- ▶ A https egy URI-séma, amely biztonságos http kapcsolatot jelöl. Szintaktikailag megegyezik a http sémával, amelyet a HTTP protokollnál használnak, de **a https nem önálló protokoll, hanem csak egy URI séma**, mely azt jelzi, hogy a HTTP protokollt kell használni a szerver 443-as TCP portján a HTTP és a TCP szintek közé titkosító/autentikáló SSL vagy TLS réteg beiktatásával.
- ▶ (A titkosítatlan HTTP rendszerint a 80-as TCP portot használja.)

- ▶ A https-t a Netscape fejlesztette ki, hogy a webes kommunikáció titkosítható és autentikálható(azonosítható) legyen.
- ▶ Ma széles körben használják ezt a rendszert a weben biztonságilag kritikus kommunikációknál, mint amilyenek például a fizetési tranzakciók és a felhasználói jelszavas bejelentkezések.

Példa: otp direkt

The image shows a web browser window displaying the OTP Bank OTPdirekt login page. The address bar shows the URL <https://www.otpbank.hu/portal/hu/OTPdirekt/Belepes>, which is highlighted with a red rectangle. The page content includes a cookie notice, a login form with fields for 'Azonosító' (ID) and 'Jelszó' (Password), and a 'Karbantartás' (Maintenance) section. A modal window titled 'Tanúsítvány' (Certificate) is overlaid on the page, displaying information about the SSL certificate used by the website. The browser's security panel is also visible on the right side of the screen.

Cookie / süti kezelés
Ahogy majdnem minden honlappal rendelkező cég, az OTP Bank is használ sütiket a weboldalain.
→ [Tudjon meg többet...](#)

Elfogadom

OTPdirekt intern
Belépés az internetbankba
Számkezelés a nap 24 órában

Karbantartás
Tájékoztatjuk, hogy a banki számlavezetés 04:00 között az OTPdirekt internetes szolgáltatásunk lesznek elérhetők.
Megértését köszönjük!

OTPdirekt belépés

Azonosító:

Számla 117: 73377-60179232-

Jelszó: Jelszó

☐ Tranzakciónkénti azonosítás

Tanúsítvány
Általános Részletek Tanúsítványlánc

Információ a tanúsítványról

A tanúsítvány a következő célokra használható:

- Távoli számítógép identitásának biztosítása
- Az Ön identitásának igazolása távoli számítógépeken
- 2.16.840.1.113733.1.7.23.6

* Részleteket a hitelesítésszolgáltató közleményében találhat.

Tulajdonos: www.otpbank.hu

Kiállító: Symantec Class 3 EV SSL CA - G3

Érvényesség: 2015. 11. 17. vége: 2016. 11. 18.

[Kiállító nyilatkozata](#)

Security Overview

Overview

Main Origin

Reload to view details

This page is secure (valid HTTPS).

- Valid Certificate
The connection to this site is using a valid, trusted server certificate.
[View certificate](#)
- Secure Resources
All resources on this page are served securely.

Console

chrome-extension://gi...obklfep Preserve log

Felhasznált források

- ▶ <http://web.progtanulo.hu/web-programozas-alapismeretek/1-az-alapok/12-a-web-mukodese/124-http-protokoll>
- ▶ <https://support.google.com/adwords/answer/2580401?hl=hu>
- ▶ <https://hu.wikipedia.org/wiki/OSI-modell>
- ▶ <http://www.zsuuu.hu/alapism/html.html>