

Webový server

Ing. Martin Dostal, Ph.D.

madostal@kiv.zcu.cz
github.com/madostal

Důležité termíny

- Zvané přednášky - viz. portál
- Zkoušky - viz. portál, už jsou vypsány na leden

Obsah

- Webový server - typy, konfigurace, zajímavosti
- Opakování - od HTML, přes CSS až po Php

Webový server

Def. Webový server je program, který reaguje na požadavky klientů na webové zdroje.

Nejznámější:

- Apache
- IIS od Microsoftu
- nginx

Web server - zpracování požadavků

- Parsování requestu klienta
- Autorizace - ověření práva přístupu
- Navázání požadovaného URL na cílový soubor
- Příprava výstupu - buď pošle přímo soubor nebo výstup vygeneruje např. z DB
- Odeslání výstupu klientovi

Apache vs Nginx

- dohromady obsluhují až 50% trafficu na internetu
- hlavní rozdíl spočívá v přístupu k obsluze požadavků
- Apache si vytvoří nové vlákno pro každé spojení zatímco nginx používá jedno vlákno (workera) pro obsluhu většího množství uživatelů

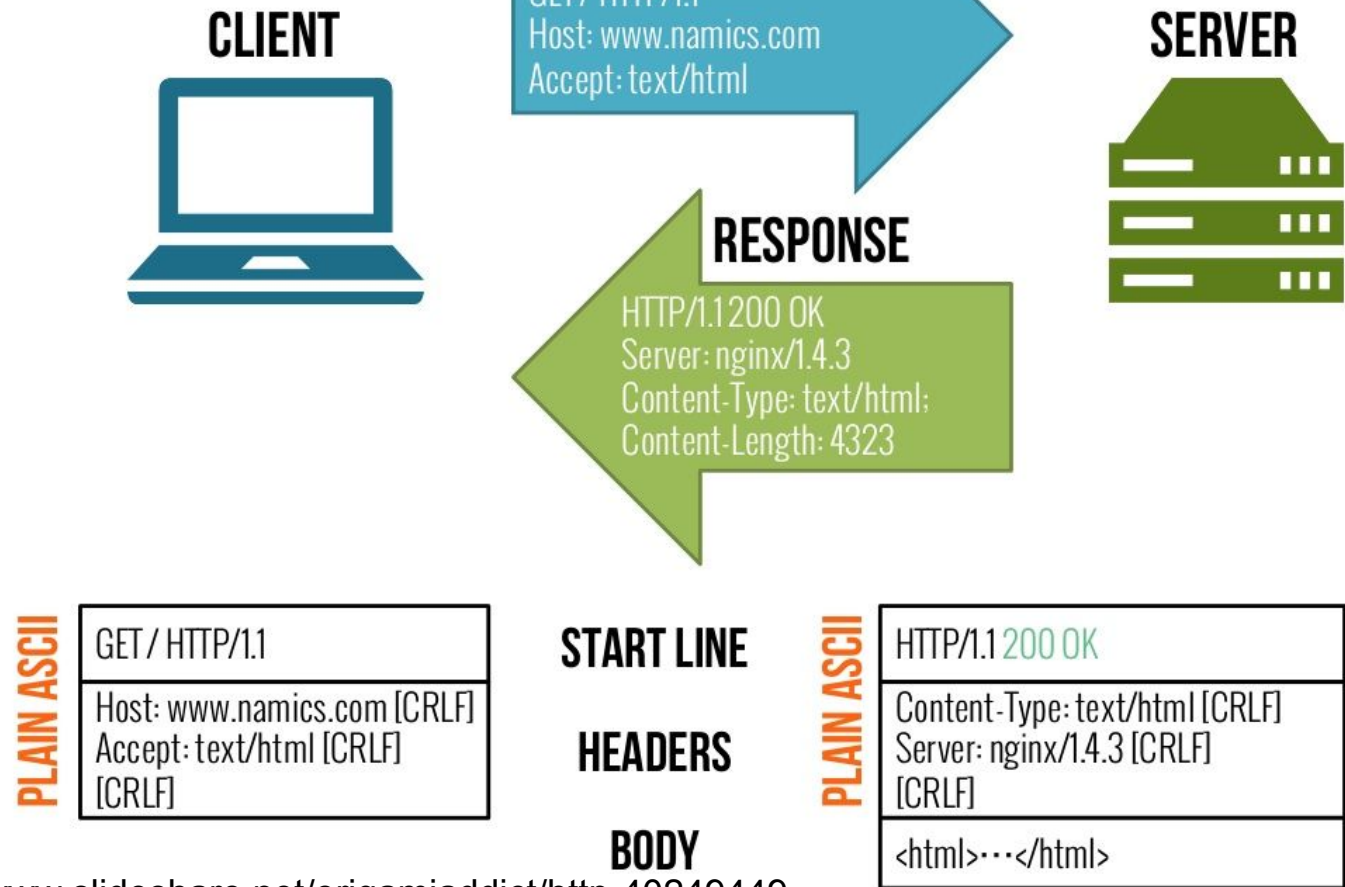
Nginx

- asynchronní událostmi řízená architektura
- asynchronous, non-blocking, event-driven connection handling algorithm
- ideální pro přístup ke statickým souborům
- přístup k dynamickým datům často přesměrovává jinam a používá se jako load balancer
- server je jedno-vláknový a jeden "worker" ve smyčce obsluhuje klienty

Nginx (2)

- zátěž serveru = spotřeba CPU i RAM je relativně konstatní i při velké zátěži
- zpracování např. Php není zabudováno do vlastního workera jako u Apache, ale je využíván externí interpret
- Nginx standardně nepodporuje .htaccess a konfiguraci přístupu dle adresářů, ale využívá jeden konfigurační soubor

HTTP protokol



HTTP požadavky

- **get** - požadavek na konkrétní stránku
 - přístup přes `$_GET`
- **post** - odeslání dat a získání výstupu ze serveru
 - přístup přes `$_POST`
- získání dat z GET i POST: `$_REQUEST`
- http protokol je bezstavový - je třeba používat např. Cookies (+Session) k udržení stavu aplikace

HTTP stavové kódy

Základní:

- 200 (OK),
- 404 (Not Found)
- 503 (Service Unavailable)

Kategorie:

- 2xx – úspěšné vyřízení požadavku
- 3xx – přesměrování
- 4xx – chyby související s vyřízením požadavku (stránka není dostupná, apod.)
- 5xx – interní chyby serveru

Access Control

- **Authentikace** - identifikace uživatele - např. formou přihlášení
 - webový formulář
 - http autentikace - řeší webový prohlížeč, nutné použití https, neboť jinak je to nešifrované (login:heslo a zakódováno s Base64 kuli přenosu přes Internet)
- **Autorizace** - určení, kteří uživatelé mají právo přístupu ke zdroji. Často používán ACL = access control list

Konfigurace Apache

- Soubor httpd.conf
- Některá defaultní nastavení nemusí být optimální
- Pro zviditelnění změn se musí server restartovat
- # pro zakomentování řádky
- ServerRoot "C:\Program Files\Apache Group\Apache"
- Cesta ke konfiguraci, logům atd.
- Místo, kde je Apache nainstalován

Konfigurace Apache

- Listen 80
 - Port, na kterém webserver poslouchá
- ServerName localhost
 - Adresa serveru (www.kiv.zcu.cz)
 - Lokálně při ladění localhost
- DocumentRoot
 - Cesta ke kořenovému adresáři webu
 - Standardně ukazuje do htdocs, lze změnit

Directory 1

- Nastavení oprávnění k jednotlivým adresářům
- Nastavení bezpečnosti webu
- Pravidla mezi tagy Directory
- Na začátku cesta a jméno adresáře
- Pokud na jeden adresář sedí více pravidel, uplatní se od nejméně specifických k nejvíce
- Pro nastavení všech adresářů použít jako jméno /
- Pravidla se uplatní i na podadresáře

Directory 2

- Lze nastavovat oprávnění i pro soubory (files):

`<FilesMatch "^\.ht">`

Order allow,deny

Deny from all

`</FilesMatch>`

Hlavní nastavení:

- Options – které vlastnosti budou pro daný adresář zapnuty
- Přidání vlastnosti + (defaultní), odebrání –
- Hodnoty: None, All, ExecCGI (povolení spuštění CGI skriptů), Indexes (pokud v adrese není indexový soubor, tak server vrátí výpis adresáře)

Directory 3

- Allow Override
- Která nastavení mohou být změněna souborem .htaccess
 - None
 - Order – allow, deny (deny, allow) – pořadí zda se nejprve povoluje a pak zakazuje nebo naopak
 - Deny/Allow form all – zakazuje/povoluje přístup odkudkoliv

Alias

- Odkazy, pomocí kterých můžeme do kořenového adresáře webu přidat adresář, který se nachází někde jinde
- Př.: Alias /mysql/ “c:/phpmyadmin”
- Musí být zapnutý modul `mod_alias`
- Měly by se nastavit práva pro tento adresář

Directory index, ErrorDocument

- DirectoryIndex
- Jména souborů, která jsou brána jako indexová
- Soubor se použije po zadání adresy bez určení souboru
- Hodnota např. index.html index.php
- ErrorDocument 404 /404.html
- Lze nastavit různý vzhled chybových hlášení

.htaccess

- Zvláštní textový soubor určený k tomu, aby si autor stránek mohl sám upravit některé vlastnosti serveru Apache
- Umístěn v adresáři, pro který chceme měnit vlastnosti (instrukce jako v httpd.conf)
- Příklad – vlastní chybová stránka
- Soubor je skrytý (. na začátku)

Zapnutí podpory .htaccess

Zapnutí podpory .htaccess v konfiguraci Apache a jeho souboru: httpd.conf

```
<Directory /var/www/>
```

```
Options Indexes FollowSymLinks
```

```
AllowOverride All
```

```
Require all granted
```

```
</Directory>
```

Podpora pro Php v Apachi

V konfiguraci Apache:

- `PHPIniDir "C:\\Program Files\\PHP\\"`
- `LoadModule php5_module "C:\\..\\php5apache2_2.dll"`

Konfigurace php – php.ini (1)

- Jméno vlastnosti = hodnota
- ; pro komentář
- Register-globals = off
 - Registrace globálních proměnných pro GET, POST, COOKIE atd.
 - Místo `$_POST['name']` lze použít `$name`
 - Vypínáno z bezpečnostních důvodů
- error_reporting = E_ALL
 - Typy zobrazovaných chybových zpráv
 - E_ALL – zobrazí vše, vhodné pro vývoj a ladění
 - E_ALL & ~E_NOTICE – vše mimo notice

php.ini (2)

- display_errors = On
 - zapnutí/vypnutí výstupu chyb
 - Na ostrém serveru jen logovat
- Session.gc_maxlifetime = 1440
 - Minimální délka session při neaktivitě

php.ini (3)

vypnutí nebezpečných php funkcí, které nepotřebujeme:

- `disable_functions = exec,passthru, shell_exec,system,proc_open,popen,curl_exec,curl_multi_exec,parse_ini_file, show_source`

php.ini (4)

Spousta dalších voleb

- include cesta (viz Smarty)
- session
- errors (viz zpracování chyb)
- extension
- Safemod
- mysql

Hezké URL adresy

- vždy je třeba, aby web nejdříve běžel s ošklivými adresami
 - `index.php?id=3` nebo `index.php?url=kontakt`
- následně je možné aktivovat hezké URL adresy:
 - je potřeba zapnout podporu `.htaccess`
 - je třeba vytvořit `.htaccess` a zapsat vhodná pravidla
 - je třeba jednotné generování URL v rámci aplikace

.htaccess - jak na hezké URL

- Přepíše ahoj.htm na index.php?q=ahoj.htm

RewriteEngine On

RewriteBase /

The Friendly URLs part - tohle je nejdůležitější

pokud daný adresář nebo soubor neexistuje, tak se provede přesměrování na index.php a do parametru q se vloží požadavek

RewriteCond %{REQUEST_FILENAME} !-f

RewriteCond %{REQUEST_FILENAME} !-d

RewriteRule ^(.*)\$ index.php?q=\$1 [L,QSA]

.htaccess - triky - odstranit www

Rewrite `www.domain.com` -> `domain.com`

RewriteCond `%{HTTP_HOST}` .

RewriteCond `%{HTTP_HOST}` `!^example-domain-please-change\.com` [NC]

RewriteRule `(.*)` `http://example-domain-please-change.com/$1` [R=301,L]

.htaccess - nastavení 404

- do .htaccess souboru v rootu webu vložíme následující příkaz:
 - **ErrorDocument 404 /404.htm**
- V případě nenalezené stránky pak zobrazí soubor 404.htm, který následně můžete přesměrovat do `index.php?q=404.htm`

Generování URL

- Potřebujeme 2 základní věci:
 - metodu např. `makeURL($id_clanku)`, která Vám např. dle kontanty `FRIENDLY_URL` vytvoří nové nebo staré URL
 - podporu pro šablony s možností generovat URL např. `[[~5]]` - odkaz na dokument s ID=5

Řešení [[~5]]

```
$result = preg_replace_callback('!\[[\~([0-9]+)\]!',  
    array($this, 'CallbackProcessLinks'), $template);
```

```
public function CallbackProcessLinks($matches) {  
    $resource_id = @$matches[1] + 0;  
    if ($resource_id > 0) {  
        if ($minicms->content->ExistsContentByID($resource_id)) {  
            // existuje, tak mohu pozadat o jeho url  
            $url = $minicms->makeUrl($resource_id);  
            return $url;  
        } ...  
    }
```


Co si pamatovat?

- prohlédnout si php.ini a pamatovat si základní bezpečnostní nastavení:
 - display_errors=Off
 - log_errors=On
 - error_log=/var/log/httpd/php_scripts_error.log
 - file_uploads=On
 - # user can only upload upto 1MB via php
 - upload_max_filesize=1M

php.ini - pamatovat:

- aby nešlo spustit vzdálený kód:
 - allow_url_fopen=Off
 - allow_url_include=Off
 - magic_quotes_gpc=Off - automatické escapování uvozovek a apostrofů, od verze Php 5.4 odstraněna
- kontrola zdrojů:
 - max_execution_time = 30
 - max_input_time = 30
 - memory_limit = 40M

Opakování

- Jaký je rozdíl mezi připojením k serveru přes localhost a 127.0.0.1? Chyták ... má to něco společného se socketama a ...řešení na další stránce
- Co to je PDO? Když nefunguje, tak co zkontrolovat?
- Co to je MVC?
- Jak prakticky udělat mini-aplikaci v MVC?

Localhost vs 127.0.0.1

1. Differs between Windows and Linux. If you use a unix domain socket it'll be slightly faster than using TCP/IP (because of the less overhead you have).
2. Windows is using TCP/IP as a default, whereas Linux tries to use a Unix
 - a. Domain Socket if you choose localhost
 - b. and TCP/IP if you take 127.0.0.1.

česky:

- IP adresa - použije se TCP/IP
- localhost - použijí se sokety, často přes UDP, nemusí fungovat na některých konfiguracích virtuálních serverů