

Protokol HTTP

4IZ228 – tvorba webových stránek a aplikací

Jirka Kosek

Poslední modifikace: \$Date: 2006/11/23 15:11:51 \$

Copyright © 2000–2011 Jiří Kosek

Obsah

Úvod	3
Co je to HTTP	4
Základní model protokolu	5
Struktura požadavku v HTTP 1.0 a 1.1	6
Metody požadavku	7
Struktura odpovědi v HTTP 1.0 a 1.1	8
Stavové kódy	9
Hlavičky	10
O hlavičkách obecně	11
Nejdůležitější hlavičky	12
Ovládání vyrovnávacích pamětí, proxy serverů a načítání stránek	13
Domlouvání obsahu	14
Identifikační údaje	15
Čtení hlaviček	16
Generování hlaviček	17
Praktické využití HTTP hlaviček	18
Zákaz kešování stránek	19
Automatické přesměrování klienta	20
Identifikace typu generovaných dat	21
Limity HTTP	22
Omezení HTTP	23
Řešení	24
Předávání stavových proměnných v URL a skrytých polích formulářů	25
Cookies	26
Předávání stavových informací pomocí cookies	27
Session proměnné	28
Předávání stavových informací pomocí session proměnných	29
Web Storage	30
Další zdroje informací	31
Další zdroje informací	32

Úvod

Co je to HTTP	4
Základní model protokolu	5
Struktura požadavku v HTTP 1.0 a 1.1	6
Metody požadavku	7
Struktura odpovědi v HTTP 1.0 a 1.1	8
Stavové kódy	9

Co je to HTTP

- HTTP = Hypertext Transfer Protocol
- protokol pro přenos objektů libovolného typu (stránky, obrázky, ...)
mezi webovým serverem a prohlížečem
- jednoduchý aplikační protokol vystavený nad protokolem TCP
- bezstavový protokol modelu požadavek/odpověď – přináší problémy
pro webové aplikace
- několik verzí – HTTP 0.9, HTTP 1.0, HTTP 1.1
- HTTP 1.1 nejnovější – podpora ve většině novějších serverů a prohlížečů

Základní model protokolu

- (1) navázání spojení
- (2) zaslání požadavku klientem
- (3) zaslání odpovědi serverem
- (4) uzavření spojení
- pro stránky s mnoha vloženými objekty (obrázky apod.) je tento způsob pomalý, a proto novější verze HTTP umožňují během jednoho spojení vyřídit několik požadavků/odpovědí

Struktura požadavku v HTTP 1.0 a 1.1

metoda URL_dokumentu verze_HTTP

hlavičky

prázdná_řádka

tělo_požadavku

Příklad 1. Ukázka jednoduchého požadavku

```
GET /clanky/obsah.html HTTP/1.1
```

```
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows NT)
```

```
Host: www.server.cz
```

Metody požadavku

GET

nejběžnější – žádost o stránku, odeslání dat z formuláře metodou GET

POST

odeslání dat z formuláře

HEAD

zaslání samotných hlaviček odpovědi

PUT

uložení objektu (stránky, obrázku apod.) na dané URL

DELETE

smazání objektu (stránky, obrázku apod.) z daného URL

TRACE, CONNECT, OPTIONS

konfigurace a analýza způsobu připojení

Struktura odpovědi v HTTP 1.0 a 1.1

protokol stavový_kód stavové_hlášení
hlavičky
prázdná_řádka
obsah_odpovědi

Příklad 2. Ukázka odpovědi

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Wed, 06 Dec 2000 13:37:40 GMT
X-Powered-By: PHP/4.0.3pl1
Content-type: text/html
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>Dobývání znalostí z&nbsp;databází 2000</title>
<link rel="stylesheet" type="text/css" href="base.css">
...
```

Stavové kódy

1xx

informativní kód

2xx

úspěšné vyřízení požadavku

3xx

přesměrování

4xx

chyba klienta

5xx

chyba na straně serveru

Hlavičky

O hlavičkách obecně	11
Nejdůležitější hlavičky	12
Ovládání vyrovnávacích pamětí, proxy serverů a načítání stránek ...	13
Domlouvání obsahu	14
Identifikační údaje	15
Čtení hlaviček	16
Generování hlaviček	17

O hlavičkách obecně

- některé hlavičky lze použít v požadavku i v odpovědi
- některé jsou specifické pro požadavek, resp. odpověď
- ne všechny hlavičky jsou povinné, většina je volitelná

Nejdůležitější hlavičky

Date

datum a čas požadavku/odpovědi

Content-Type

druh zasílaných dat

Host

doménová adresa serveru – umožňuje správnou funkci více virtuálních serverů na jedné společné adrese

Location

přesměrování na jinou stránku

Ovládání vyrovnávacích pamětí, proxy serverů a načítání stránek

Cache-Control

řízení proxy serverů a vyrovnávacích pamětí

Pragma

vyhrazeno pro nestandardní informace (nejčastěji zákaz kešování pro starší prohlížeče)

Expires

datum, kdy vyprší platnost stránky

If-Modified-Since

podmíněné načtení stránky

Last-Modified

datum poslední modifikace souboru

Domlouvání obsahu

Accept

seznam typů dat podporovaných klientem

Accept-Charset

seznam kódování, které podporuje klient

Accept-Language

seznam podporovaných jazyků

Allow

seznam metod, kterými je dostupný určitý objekt

Identifikační údaje

User-Agent

identifikace klienta

Server

identifikace serveru

Referer

adresa stránky, kde bylo získáno URL právě kladeného požadavku (Ize použít pro analýzu typu „odkud přišli“)

From

e-mailová adresa uživatele (ještě jsem neviděl prohlížeč, který by ji posílal;)

Čtení hlaviček

- většina hlaviček je CGI rozhráním převedena na proměnné
- např. User-Agent → `$HTTP_USER_AGENT`

Generování hlaviček

- v CGI je stačí rovnou zapsat před tělo odpovědi
- v PHP je k dispozici funkce Header:

```
<?php Header("Content-Type: image/gif") ?>
```

Praktické využití HTTP hlaviček

Zákaz kešování stránek	19
Automatické přesměrování klienta	20
Identifikace typu generovaných dat	21

Zákaz kešování stránek

- informace na stránce se mění v čase
 - burzovní indexy
 - on-line přístup do IS
 - reklamní bannery

Expires: Mon, 26 Jul 1997 05:00:00 GMT *datum v minulosti*

Last-Modified: *aktuální datum a čas*

Cache-Control: no-cache

Cache-Control: post-check=0, pre-check=0

Pragma: no-cache

- používat s rozvahou, mnohdy zbytečně zatěžuje přenosové kapacity
- některé proxy servery hlavičky ignorují – do všech URL se pak musí vkládat jedinečný řetězec

Automatické přesměrování klienta

- při pohybu v historii stránek může dojít k nechtěnému opětovnému zaslání dat z formuláře
- vznikají duplicity v databázi, nebo se vypisují chybová hlášení
- stránka obsluhující formulář by měla být v optimálním případě vyřazena z historie stránek
- stránka, která posílá hlavičku `Location`, se do historie nezařadí
- pozor, adresa v hlavičce `Location` musí být absolutní

Identifikace typu generovaných dat

- pokud chceme skriptem generovat jiné druhy dat než HTML (např. obrázky, soubory ve Wordu apod.) musíme nastavit správný typ v hlavičce
- Např.: `Content-Type: image/gif`
- pokud chceme vygenerovat soubor, který bude nabídnut k uložení, lze použít následující hlavičky

`Content-type: application/octet-stream`

`Content-disposition: filename=najakysoubor.dat`

Limity HTTP

Omezení HTTP	23
Řešení	24
Předávání stavových proměnných v URL a skrytých polích formulářů	25
Cookies	26
Předávání stavových informací pomocí cookies	27
Session proměnné	28
Předávání stavových informací pomocí session proměnných	29
Web Storage	30

Omezení HTTP

- protokol HTTP je bezstavový
- server nemá stále spojení s klienty a nemůže je proto jednoznačně identifikovat
- velké komplikace pro webové aplikace, které vyžadují stavovou informaci – např. nákupní košík

Řešení

- přenášení údajů v URL a skrytých polí formuláře
- cookies
- session proměnné
- Web Storage (HTML5)

Předávání stavových proměnných v URL a skrytých polích formulářů

- nebezpečné – všechny stavové informace jsou v každém požadavku/odpovědi
- zbytečně zvyšuje přenosovou kapacitu
- velmi pracné na implementaci – za každý odkaz a do každého formuláře se musí přidat všechny stavové proměnné

Cookies

- krátká informace, kterou si server uloží v prohlížeči
- při následujících přístupech k témuž serveru je cookie zaslána zpět
- cookie je vázána na server a případně i na adresář – informace se nedostanou k tomu, komu nepatří
- časová platnost cookie
 - session cookie – platí do té doby, než se vypne prohlížeč

```
SetCookie('název', hodnota)
```

- nastavena na konkrétní délku

```
SetCookie('název', hodnota, platnost)
```

- cookie třetích stran, rizika, P3P

Předávání stavových informací pomocí cookies

- nebezpečné – všechny stavové informace jsou v každém požadavku/odpovědi
- implementace je velice snadná
- podporu cookies lze v prohlížeči vypnout, proto by dobře napsaná aplikace měla fungovat i bez nich

Session proměnné

- každému novému uživateli se přiřadí unikátní identifikátor (tzv. session-id)
 - předává se s každým požadavkem pomocí cookie nebo parametrů v URL, resp. skrytých polí ve formuláři
 - session-id je konstruováno tak, aby bylo těžko odhadnutelné (většinou náhodné číslo + hashovací funkce MD5 nebo SHA)
- pro každé session-id má webový server vyhrazen prostor pro ukládání dat (proměnných)
 - sdílená paměť
 - soubory
 - databáze

Předávání stavových informací pomocí session proměnných

- poměrně bezpečné – s každým požadavkem se přenáší jen malá část dat a session-id
- šetří kapacitu sítě – data jsou ukládána přímo na web-serveru
- velice snadná implementace – většina prostředí pracuje se session proměnnými téměř stejně jako s běžnými proměnnými
- podpora session proměnných ve skriptových prostředích:
 - ASP – zabudovaná podpora, pracuje pouze s cookies
 - PHP3 – lze použít PHPLIB, obsahuje nástroje, které usnadňují použití i při vypnutí cookies
 - PHP4, PHP5 – zabudovaná podpora, podporuje cookies i automatické přepisování URL adres
 - JSP – zabudovaná podpora, podporuje cookies, velice snadno může podporovat i přepisování URL adres
 - ASP.NET – zabudovaná podpora, podporuje cookies i automatické přepisování URL adres

Web Storage

- úložiště dat na klientovi
- součást HTML5, podporováno všemi moderními prohlížeči
- pojme více dat než cookies a nepřenáší se na server, data zůstávají u klienta
- localStorage – je persistentní i přes uzavření prohlížeče
- sessionStorage – platné jen po dobu jedné relace

Další zdroje informací

Další zdroje informací	32
------------------------------	----

Další zdroje informací

HTTP a cookies

- HTTP 1.0¹
- HTTP 1.1²
- Původní specifikace cookies³
Aktuální specifikace cookies⁴

Session proměnné

- Použití session proměnných v PHP⁵

Web Storage

- Specifikace Web Storage⁶

¹ <ftp://ftp.vse.cz/pub/docs/rfc/rfc1945.txt>

² <ftp://ftp.vse.cz/pub/docs/rfc/rfc2616.txt>

³ http://home.netscape.com/newsref/std/cookie_spec.html

⁴ <http://www.faqs.org/rfcs/rfc2965.html>

⁵ <http://cz2.php.net/manual/en/session.examples.basic.php>

⁶ <http://dev.w3.org/html5/webstorage/>