

# Hypertext Transfer Protocol

---

**HTTP** (Hypertext Transfer Protocol) je internetový protokol určený pro výměnu hypertextových dokumentů ve formátu HTML. Používá obvykle port TCP/80, verze 1.1 protokolu je definována v RFC 2616. Tento protokol je spolu s elektronickou poštou tím nejvíce používaným a zasloužil se o obrovský rozmach internetu v posledních letech.

V současné době je používán i pro přenos dalších informací. Pomocí rozšíření MIME umí přenášet jakýkoli soubor (podobně jako e-mail), používá se společně s formátem XML pro tzv. webové služby (spouštění vzdálených aplikací) a pomocí aplikačních bran zpřístupňuje i další protokoly, jako je např. FTP nebo SMTP.

HTTP používá jako některé další aplikace tzv. jednotný lokátor prostředků (URL, Uniform Resource Locator), který specifikuje jednoznačné umístění nějakého zdroje v Internetu.

K protokolu HTTP existuje také jeho bezpečnější verze HTTPS, která umožňuje přenášená data šifrovat a tím chránit před odposlechem či jiným narušením.

## Činnost protokolu

---

Protokol funguje způsobem dotaz-odpověď. Uživatel (pomocí programu, obvykle internetového prohlížeče) pošle serveru dotaz ve formě čistého textu, obsahujícího označení požadovaného dokumentu, informace o schopnostech prohlížeče apod. Server poté odpoví pomocí několika řádků textu popisujících výsledek dotazu (zda se dokument podařilo najít, jakého typu dokument je atd.), za kterými následují data samotného požadovaného dokumentu.

Pokud uživatel bude mít po chvíli další dotaz na stejný server (např. proto, že uživatel v dokumentu kliknul na hypertextový odkaz), bude se jednat o další, nezávislý dotaz a odpověď. Z hlediska serveru nelze poznat, jestli tento druhý dotaz jakkoli souvisí s předchozím. Kvůli této vlastnosti se protokolu HTTP říká **bezstavový protokol** – protokol neumí uchovávat stav komunikace, dotazy spolu nemají souvislost. Tato vlastnost je nepříjemná pro implementaci složitějších procesů přes HTTP (např. internetový obchod potřebuje uchovávat informaci o identitě zákazníka, o obsahu jeho „nákupního košíku“ apod.). K tomuto účelu byl protokol HTTP rozšířen o tzv. HTTP cookies, které umožňují serveru uchovávat si informace o stavu spojení na počítači uživatele.

## Ukázka komunikace

---

Uživatelský program (klient) se připojí na server cs.wikipedia.org a zašle následující dotaz:

```
GET /wiki/Wikipedie HTTP/1.1
Host: cs.wikipedia.org
User-Agent: Opera/9.80 (Windows NT 5.1; U; cs) Presto/2.5.29 Version/10.60
Accept-Charset: UTF-8,*
```

Tímto dotazem žádá o dokument /wiki/Wikipedie na serveru cs.wikipedia.org, sděluje svou totožnost (Opera verze 10.60) a oznamuje, že podporuje kódování UTF-8 (ve skutečném dotazu je podobných informací ještě více, toto je zjednodušený příklad).

Server pak odpoví:

```
HTTP/1.0 200 OK
Date: Fri, 15 Oct 2004 08:20:25 GMT
Server: Apache/1.3.29 (Unix) PHP/4.3.8
X-Powered-By: PHP/4.3.8
Vary: Accept-Encoding, Cookie
Cache-Control: private, s-maxage=0, max-age=0, must-revalidate
Content-Language: cs
Content-Type: text/html; charset=utf-8
```

Za touto hlavičkou následuje jeden prázdný řádek (označující její konec) a pak požadovaný HTML dokument. Hlavička obsahuje informaci o tom, že dotaz se podařil (první řádek: „200 OK“), datum a čas vyřízení dotazu, popis serveru, který odpovídá, informace o typu vráceného dokumentu (MIME typ text/html v kódování UTF-8) a další informace.

## Verze HTTP

---

První byla publikována verze 0.9 v roce 1991 [1]. Existovala tehdy pouze metoda `GET` s jediným parametrem a to názvem požadovaného dokumentu. Server jako odpověď poslal přímo požadovaný dokument bez hlavičky (`HTTP/... 200 OK...`) uvedené v předchozí kapitole. Jednodušší už to být nemohlo. Případná chybová hlášení vracel server také ve formě HTML dokumentu.

HTTP/1.0 je popsáno v RFC 1945 z května 1996 a přináší HTTP hlavičky, nové metody `HEAD` a `POST`, pro identifikaci typu dokumentů používá MIME.

HTTP/1.1 bylo původně popsáno v RFC 2068 (leden 1997), v červnu 1999 nahrazeno RFC 2616. Tato verze umožňuje udržovat TCP spojení (tzv. `keep-alive connection`), tím pádem umožňuje i zpracování více požadavků po sobě v jednom spojení. Dále přidává další metody `OPTIONS`, `PUT`, `DELETE`, `TRACE` a `CONNECT`.

Existuje ještě experimentální rozšíření PEP (Protocol Extension Protocol, viz [2] nebo RFC 2774), které bylo jeden čas označováno jako HTTP/1.2, ale v praxi se zřejmě nepoužívá.

## Dotazovací metody

---

HTTP definuje několik metod, které se mají provést nad uvedeným objektem (dokumentem).

<metoda> <objekt> HTTP/<verze>

### GET

Požadavek na uvedený objekt se zasláním případných dat (proměnné prohlížeče, session id, ...). Výchozí metoda při požadavku na zobrazení hypertextových stránek, RSS feedů aj.

Celkově nejpoužívanější.

### HEAD

To samé jako metoda `GET`, ale už nepředává data. Poskytne pouze metadata o požadovaném cíli (velikost, typ, datum změny, ...).

### POST

Odesílá uživatelská data na server. Používá se například při odesílání formuláře na webu. S předaným objektem se pak zachází podobně jako při metodě `GET`. Data může odesílat i metoda `GET`, metoda `POST` se ale používá pro příliš velká data (více než 512 bajtů, což je velikost požadavku `GET`) nebo pokud není vhodné přenášet data zobrazením jako součást URL (data předávaná metodou `POST` jsou obsažena v HTTP požadavku).

### PUT

Nahrává data na server. Objekt je jméno vytvářeného souboru. Používá se velmi zřídka, pro nahrávání dat na server se běžně používá FTP nebo SCP/SSH.

### DELETE

Smaže uvedený objekt ze serveru. Jsou na to potřeba jistá oprávnění stejně jako u metody `PUT`.

## **TRACE**

Odešle kopii obdrženého požadavku zpět odesílateli, takže klient může zjistit, co na požadavku mění nebo přidávají servery, kterými požadavek prochází.

## **OPTIONS**

Dotaz na server, jaké podporuje metody.

## **CONNECT**

Spojí se s uvedeným objektem před uvedený port. Používá se při průchodu skrze proxy pro ustanovení kanálu SSL.

## **Zabezpečené HTTP**

---

Existují dvě metody zabezpečeného http připojení: HTTPS URI a nadstavba HTTP 1.1 představená v RFC 2817. Druhou metodu ovšem zatím prohlížeče moc nepodporují, takže HTTPS se k vytvoření zabezpečené komunikace používá nejčastěji.

### **HTTPS URI**

Je syntakticky identické jako http, pouze přidává signalizaci prohlížeči, aby použil šifrovací metodu SSL/TLS k přenosu dat. SSL je vhodné pro HTTP, protože dokáže poskytnout ochranu přenosu, i když je pouze jedna strana komunikace ověřená. Typicky je ověřen pouze server (např. uživatel potvrdí certifikát). Aby pomocí HTTPS bylo možné rozlišovat virtuální servery, existuje rozšíření SNI.

---