

مهمترین تفاوت‌های HTTP/2 را HTTP/1.1 را :

1. نسخه‌ی HTTP/2 حاوی داده‌های باینری (دودویی) است:

از داده‌های متنی استفاده می‌کند و این در حالی است که داده‌های متنی به طور کلی در سراسر شبکه از بازدهی

کمتری نسبت به داده‌های باینری برخوردارند.

2. هدر‌های HTTP/2 فشرده شده هستند:

به طور کلی منظور از Header اطلاعاتی است که در پاسخ به یک ریکوئست ارسال می‌شود که شامل دیتا، مبدأ، نوع،

حجم، مدت زمان گش و موارد دیگر است. برخلاف HTTP/1.1، این داده‌ها در نسخه HTTP/2 فشرده‌سازی می‌شوند

تا پرفورمنس ارتقاء یابد.

3. نسخه‌ی HTTP/2 اصطلاحاً Asynchronous است:

در HTTP/1.1، سرور باید به همان ترتیبی که ریکوئست‌ها را دریافت کرده است، ریسپانس‌ها را ارسال کند اما نسخه‌ی

Asynchronous اصطلاحاً HTTP/2 است. بنابراین پاسخ‌های سریع‌تر و در عین حال با حجم کمتری می‌تواند در زمان

کوتاه‌تری از سمت سرور ارسال شود.

4. نسخه‌ی HTTP/2 مولتی پلکس است:

در HTTP/1.1، فقط یک درخواست روی یک کانکشن اینترنتی TCP در آن واحد می‌تواند به کار گرفته شود و مرورگرها

به طور عادی قادر به ایجاد 4 تا 8 کانکشن با سرور هستند و این در حالی است که ریکوئست‌هایی با حجم زیاد می‌توانند

سرعت دانلود فایل‌های دیگر را به تاخیر بیندازند **HTTP/2** اجازه‌ی ارسال چندین ریکوئست (درخواست) و دریافت

ریسپانس (پاسخ) از سمت سرور را به طور همزمان بر روی یک کانکشن امکان‌پذیر می‌سازد.

5. نسخه‌ی **HTTP/2** امکان استفاده از **Server Push** را فراهم می‌سازد:

با استفاده از این نسخه از پروتکل **http**، سرور می‌تواند فایل‌ها و به طور کلی هر نوع داده‌ای را قبل از آنکه ریکوئستی ارسال

شود، برای مرورگر بفرستد که به این فناوری اصطلاحاً **Server Push** گفته می‌شود. برای مثال، ممکن است شما در پایین

صفحه‌ی خود به یک اسکریپت لینک دهید. در **HTTP/1.1**، مرورگر کدهای **HTML** را دانلود می‌کند، تجزیه می‌کند و

سپس فایل جاوا اسکریپت را بارگذاری می‌کند (این بارگذاری هنگامی است که با تگ **<script>** روبرو شویم). سروری که

HTTP/2 را ساپورت کند، می‌تواند چنین فایلی را قبل از اینکه نیاز آن را تشخیص دهد، برای مرورگر ارسال کند که در

نتیجه در صورت نیاز، کاربر معطل دانلود شدن فایل‌های جی‌اس نخواهد شد (که این به معنی **UX** بهتر است).

کد وضعیت **xx3** به این معنی است که شما مجدداً هدایت شده‌اید و تکمیل درخواست به اقدامات بیشتری نیاز دارد.

کدهای وضعیت 300

- **300 انتقال چندگانه (Multiple Choices)**: کد 300 به این معنی است که درخواست چندین پاسخ احتمالی دارد و کاربر باید یکی را انتخاب کند.
- **301 انتقال دائمی (Moved Permanently)**: به این معنی است که صفحه یا فایل مورد نظر ما جابجا شده است.
- **302 پیدا شده (Found)**: این کد وضعیت یعنی درخواست مورد نظر در جایی غیر از آدرس مشخص شده پیدا شده است.
- **304 اصلاح نشده (Not Modified)**: کد 304 به مشتری اطلاع می‌دهد که پاسخ اصلاح نشده است. به این معنی که مشتری می‌تواند از نسخه‌های موجود در حافظه پنهان یا کش استفاده کند تا بارگیری صفحه زودتر انجام شود.

• 305 پروکسی (Use Proxy): این کد به کاربر اطلاع می‌دهد که برای دسترسی به منبع باید از پروکسی استفاده کند.

• 306 تغییر پروکسی (Switch Proxy): مانند کد قبلی عمل می‌کند، با این تفاوت که به کاربر اطلاع می‌دهد برای ادامه پروکسی را تغییر دهد.

در اینجا توضیحی در مورد هر هدر، اعم از استفاده از آن در یک درخواست یا پاسخ، و هدف آن آمده است:

: این هدر نام دامنه و در صورت تمایل، شماره پورت سرور میزبان منبع درخواستی را مشخص می‌کند. در درخواست های HTTP/1.1 مورد نیاز است.

: این هدر نشان دهنده URL صفحه ای است که عامل کاربر (معمولًاً یک مرورگر وب) را به منبع درخواستی ارجاع داده است. برای تجزیه و تحلیل، ورود به سیستم و برای کنترل دسترسی به برخی منابع بر اساس صفحه ارجاع استفاده می‌شود.

: این سربرگ کد(های) محتوا را مشخص می‌کند که عامل کاربر قادر به درک آن است. سرور می‌تواند از این اطلاعات برای ارسال پاسخ با استفاده از یک رمز گذاری مناسب برای کاهش اندازه داده های منتقل شده استفاده کند. رمز گذاری های رایج عبارتند از deflate و gzip

: این هدر نوع رسانه (نوع MIME) منبع را در درخواست یا بدن پاسخ نشان می‌دهد. برای مثال، text/html یا application/json برای یک شی JSON

: این هدر اندازه درخواست یا بدن پاسخ را بر حسب بایت ارائه می‌کند. این به گیرنده اجازه می‌دهد تا بداند چه زمانی بدن به طور کامل دریافت شده است.

: این هدر همراه با درخواست ها و پاسخ های محتوای جزئی استفاده می‌شود که محدوده بایت های ارسالی و اندازه کل منبع را نشان می‌دهد. معمولًاً برای از سرگیری بارگیری های قطع شده یا پخش جریانی فایل های بزرگ استفاده می‌شود.

URL Location (Response): این هدر در پاسخ های تغییر مسیر (به عنوان مثال، کد های وضعیت 3xx) برای نشان دادن مکان جدیدی که عامل کاربر باید درخواست کند، استفاده می شود.

Last-Modified (Response): این هدر تاریخ و زمانی را که منبع درخواستی آخرین بار اصلاح شده را نشان می دهد. این توسط عامل کاربر برای اهداف کش استفاده می شود تا مشخص کند آیا نسخه ذخیره شده محلی هنوز به روز است یا خیر.

Cache-Control (Request/Response): این هدر دستورالعمل های کش را برای عامل کاربر و هر حافظه پنهان میانی (مانند CDN) در مورد نحوه کش کردن منبع، مدت زمان ذخیره آن و اینکه آیا می توان از حافظه نهان بدون استفاده از حافظه پنهان ارائه کرد، ارائه می کند. اعتبار سنجی مجدد نمونه هایی از دستورالعمل ها عبارتند از حداکثر سن، بدون حافظه پنهان و خصوصی.

این هدر ها نقش اساسی در مدیریت ارتباط بین مشتریان و سرور ها، ارائه اطلاعات در مورد منابع درخواستی و بهینه سازی انتقال داده ها دارند.