. 1مهم ترین تفاوت های HTTP 2/را با HTTP 1.1/توضیح دهید.

.1نسخهٔ HTTP/2 حاوی داده‌های باینری (دودویی) است :از داده‌های متنی استفاده می‌کند و این در حالی HTTP/1.1 است که داده‌های متنی به طور کلی در سراسر شبکه از بازدهی کمتری نسبت به داده‌های باینری برخوردارند.

.2هِدِرهای 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 گَش و موارد دیگر است. برخلافHTTP/1.1 ، این داده‌ها در نسخهٔ HTTP/2 فشرده‌سازی می‌شوند تا پرفورمنس ارتقاء یابد.

.3نسخهٔ HTTP/2 اصطلاحاً Asynchronous است :درHTTP/1.1 ، سرور باید به همان ترتیبی که ریکوئست‌ها را دریافت کرده است، ریسپانس‌ها را ارسال کند اما نسخهٔ HTTP/2 اصطلاحاً Asynchronous است؛ بنابراین پاسخ‌های سریع‌تر و در عین حال با حجم کمتری می‌تواند در زمان کوتاه‌تری از سمت سرور ارسال شود.

.4نسخهٔ HTTP/2 مولتی‌پلکس است :درHTTP/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 تأخیر بیندازند HTTP/2 !اجازهٔ ارسال چندین ریکوئست (درخواست) و دریافت ریسپانس (پاسخ) از سمت سرور را به طور هم‌زمان بر روی یک کانکشن امکان‌پذیر می‌سازد.

.5نسخهٔ HTTP/2 امکان استفاده از Server Push را فراهم می‌سازد :با استفاده از این نسخه از پروتکل اچ‌تی‌تی‌پی، 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 صفحهٔ خود به یک اسکریپت لینک دهید. در HTTP/1.1، مرورگر کدهای HTML را دانلود می‌کند، تجزیه می‌کند و سپس فایل جاوااسکریپت را بارگذاری می‌کند (این بارگذاری هنگامی است که با تگ <script>روبه‌رو 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.

. 2مهم ترین تفاوت های HTTP 3/را با HTTP 2/توضیح دهید.

HTTP3 بر روی UDP ساخته شده است اما HTTP2 بر روی TPC ساخته شده است .

HTTP3 ار طریق رمزگزاری یکپارچه TLS 1.3 استفاده می کند و برابراین از سوالات امنیتی غیر ضروری جلوگیری میکند .

HTTP3 به دلیل رمزگزاری TLS 1.3 فقط از اتصالات رمزگذاری شده پشتیبانی میکند.

. 3توضیح دهید که کدهای وضعیت 3 xxارسال شده از سمت سرور چه عملکردی روی کلاینت دارند

300: پاسخ 300 گزینه چند گزینه ای نشان می دهد که گزینه های متعددی برای منبعی که مشتری می تواند از آن انتخاب کند وجود دارد. این گزینه ها می توانند شامل مذاکره محتوای عامل محور باشند. با ذکر یک فایل از پسوندهای مختلف نام پرونده ، می توان از یک نمونه از این موارد برای جلوگیری از چندین گزینه قالب ویدیو استفاده کرد.

301: پاسخ 301 به طور دائم منتقل شده به این معنی است که این درخواست و کلیه درخواست های آینده باید به URL انتخاب شده هدایت شوند.

302: پاسخ 302 نشان می دهد که مشخصات (RFC 1945) HTTP / 1.0 به مشتری نیاز دارد تا یک هدایت موقت را انجام دهد. این پاسخ نمونه ای از اقدامات صنعت در تضاد با استاندارد است. در اصل این پاسخ به عنوان "به طور موقت منتقل شد" توصیف شد. با این حال ، مرورگرها 302 را به عنوان یک پاسخ و با عملکرد 303 پیاده سازی کردند. در نتیجه ، HTTP / 1.1 کدهای وضعیت 307 و 303 را برای تمایز بین این دو اضافه کرد. برخی از برنامه های وب همچنان از کد وضعیت 302 استفاده می کنند ، اگرچه گویی که 303 است.

303: پاسخ 303 را می توان با استفاده از روش GET تحت URI دیگر یافت. هنگامی که این در پاسخ به / PUT DELETE یا POST بدست می آید ، مشتری باید تصور کند که سرور داده را دریافت کرده است و باید منتظر هدایت مجدد با یک پیام GET جداگانه باشد.

304: پاسخ 304 Not Modified به این معنی است که از زمان ارائه نسخه های ارائه شده توسط عناوین درخواست پاسخ های "اگر هیچکدام از آنها مطابقت نداشته باشد" یا "اگر از آن زمان تغییر داده شده است" منبع تغییر نکرده است. در این شرایط ، نیازی به انتقال مجدد منبع نیست زیرا مشتری هنوز یک نسخه بارگیری شده دارد.

305: پاسخ 305 به این معنی است که منبع درخواستی فقط از طریق پراکسی در دسترس است. پروکسی آدرس ارائه شده در پاسخ است. Mozilla و Internet Explorer دو مشتری HTTP هستند که به دلایل امنیتی پاسخگویی را با این کد وضعیت انجام نمی دهند.

306: این پاسخ وضعیت دیگر استفاده نمی شود. در ابتدا نشان داد که درخواستهای بعدی باید به پروکسی مشخص شده برگردند.

307: درخواستی که پاسخ 307 دریافت می کند باید با URI دیگری تکرار شود. با این حال ، در درخواست های آینده باید همچنان از URI اصلی استفاده شود. در گذشته ، هنگام درخواست مجدد درخواست ، اجازه تغییر در درخواست های 302 داده نمی شد. یک مثال از این درخواست POST است که باید با استفاده از درخواست POST دیگر تکرار شود.

308: پاسخ 308 نشان می دهد که درخواست فعلی و همه درخواست های آینده باید با استفاده از URI دیگر تکرار شود. پاسخ های زیر 307 و 308 موازی اقدامات 301 و 302 پاسخ هستند. اما این اجازه نمی دهد که روش HTTP تغییر کند.

.ً 4توضیح دهید که سرآیندهای زیر اولا در کدام یک از بسته های درخواست یا پاسخ استفاده می شوند و ثانیا کاربردی دارند؟

Request Headers: Host _ Referrer _ Accept-Encoding _ Content-Type _Content-Length _ Cache Control

Response Header : Content-Type _ Content-Length_Location_Last-Modified_Cache Control _Content-Range

Host:مشخص کننده نام هاستی که به آن درخواست ارسال می شود .

Referrer: از کجا به اینجا آمده اید، مثال از صفحه اصلی سایت به یکی از صفحات فرعی رفته ایم. معادل دکمه Back در مرورگر می باشد.

Accept-Encoding: مرورگر یک درخواست ارسال می کند تا بررسی کند آیا سرور فشرده سازی gzip را پشتیبانی می کند یا نه. اگر درخواست از سمت سرور مثبت شناسایی شد، سرور فایل های فشرده شده را به سمت مرورگر ارسال می کند و این امر می تواند تا 70درصد در سرعت بارگذاری صفحات وب تاثیرگذار باشد.

Content-Type: نوع سند ارسال شده را بر اساس استاندارد MIME در داخل آن قرار گرفته شده است

Content-Length: طول داده ها بر حسب بایت و بر اساس استاندارد MIME در داخل آن قرار گرفته شده است.

Cache-Control: زمان و نحوه ی کش شدن را برای ی ک فایل مشخص می کند. مثالی از این روش، کد زیر است : Cache-Control: max-age=2592000, publicزمان ی که فایلی توسط مرورگر ارسال می شود header HTTP ها ی آن فایل ن یز به همراه فایل دریافت می شوند. زمان ی که مرورگر header مخصوص cache را به همراه فایل میبیند، طبق دستور العمل آن عمل خواهد کرد.

Location: نشانگر URL برای هدایت به یک صفحه است.

Last-modified: آخرین تاریخ اصالح در آن قرار داده می شود و برای مقایسه چند نسخه از منبع استفاده می شود.

Content-Range: نشان می دهد که داخل یک پیام، تمام بدنه یک پیام، جزئی از کل است.