

به نام خدا
تمرین HTTP
تمرین سری اول مهندسی اینترنت

-۱

http 1.0:

non-persistent, support GET, HEAD, POST methods

http 1.1:

persistent but loads resources one after another, supporting more methods than previous version, messages in plaintext

http 2.0:

persistent but with multiplexing, messages in binary format

http 3.0:

برای افزایش سرعت از QUIC استفاده میکنه (Quick UDP Internet Connections) که در آن مشکل blocking (وقتی یک مشکل در ارسال یک پیام به هر علتی رخ دهد تمام ارسال ها باید متوقف شوند و البته از سر گرفته شوند تا دوباره آن بسته ارسال شود و اگر چندین جریان داشته باشیم همه ی آن ها باید متوقف شوند) را دیگر نداریم. کانکشن ها سریع تر شده_تعداد handshake های کمتر)

-۲

در پروتکل stateful سرور نیاز دارد تا تمام اطلاعات درمورد یک session را نگهداری کند. طراحی سرور را پیچیده میکند. به منابع بیشتری نیاز دارد. به وضعیت سرور همیشه وابسته است. اطلاعات کانکشن را نگه می دارد و برای همین performance بهتری را برای کلاینت فراهم میکند. در هنگام crash به خوبی عمل نمیکند چونب اید اطلاعات مربوط به session را نگهداری کند. کندتر بودن تراکنش

Stateless : در این پروتکل نیاز نیست که سرور اطلاعات یک session یا وضعیت ارتباط را نگهداری کند - طراحی ساده تر سرور - به منابع بیشتری نیاز دارد. هر پکت اطلاعات بدون اشاره به پکت دیگری مسیر را طی میکند. هر ارتباط در این پروتکل گسسته است بی ارتباط با قبلی ها و بعدی ها. در زمان crash بهتر عمل میکند چون هیچ وضعیتی نیست که نیاز به بازیابی داشته باشد و به سادگی ریستارت میکند. سریع تر بودن تراکنش

(ب)

Stateless است و تاریخچه یه وضعیت کلاینت را نگهداری نمیکند. و با استفاده از کوکی ها یا JWT مشکلات مربوط به آن را حل میکند. سرور اطلاعاتی را به کاربر ارسال میکند کهدر درخواست های بعدی کلاینت آن ها را دوباره به سرور ارسال میکند.

-۳

(الف)

همیشه باید کلاینت ارتباط را استارت بزند چون سرور که نمیداند چه کسی قرار است بهش وصل شود. چالش آن این است که روند آن اینگونه است که کاربر حتما باید درخواست بزند و سرور به آن جواب دهد. و سرور فقط در پاسخ به درخواست های کاربر میتواند چیزی بفرستد. برای نمونه در یک سیستم اخطار دهی، به محض رسیدن اخطار باید سرور به کاربر خبر دهد یا مثلا در یک پت اپلیکیشن سرور به محض رسیدن پیام لازم هست که پیام را به کلاینت برساند که با http این شدنی نیست و این چالش است.

(ب)

Long polling:

Client pull معادل اجرای نرم افزاری است که کودک مزاحم در صندلی عقب ماشین شما نشسته است و دائماً می پرسد: "آیا ما هنوز آنجا هستیم؟" به طور خلاصه، کلاینت از سرور اطلاعات می خواهد. سرور هیچ داده ای ندارد و قبل از ارسال پاسخ، مدتی منتظر می ماند: اگر در طول انتظار چیزی ظاهر شود، سرور آن را ارسال می کند و درخواست را می بندد. اگر چیزی برای ارسال وجود نداشته باشد و حداکثر زمان انتظار به دست آید، سرور پاسخی ارسال می کند که داده ای وجود ندارد. در هر دو مورد، مشتری درخواست بعدی برای داده را باز می کند.

:Web socket

WebSocket یک پروتکل ارتباطی کامپیوتری است که کانال های ارتباطی تمام دابلکس را از طریق یک اتصال TCP فراهم می کند.

WebSocket از HTTP متمایز است. هر دو پروتکل در لایه ۷ مدل OSI قرار دارند و در لایه ۴ به TCP وابسته هستند. اگرچه آنها متفاوت هستند، RFC 6455 بیان می کند که "WebSocket برای کار بر روی پورت های 443 HTTP و ۸۰ و همچنین برای پشتیبانی از پروکسی ها و واسطه های HTTP طراحی شده است."، بنابراین آن را با HTTP سازگار می کند. برای دستیابی به سازگاری، دست دادن WebSocket از هدر ارتقاء [1] HTTP برای تغییر از پروتکل HTTP به پروتکل WebSocket استفاده می کند.

یک مرورگر وب (یا سایر برنامه های سرویس گیرنده) و یک وب سرور با سربر کمتر از گزینه های نیمه دابلکس مانند نظرسنجی HTTP را امکان پذیر می کند و انتقال داده ها را در زمان واقعی از و به سرور تسهیل می کند. این امر با ارائه روشی استاندارد شده برای سرور برای ارسال محتوا به مشتری بدون درخواست اولیه توسط مشتری، و امکان ارسال پیام ها به عقب و جلو و در عین حال باز نگه داشتن اتصال ممکن می شود. به این ترتیب، یک مکالمه مداوم دو طرفه می تواند بین مشتری و سرور انجام شود. ارتباطات معمولاً از طریق پورت TCP شماره ۴۴۳ (یا ۸۰ در مورد اتصالات ناامن) انجام می شود که برای محیط هایی که اتصالات اینترنت غیر وب را با استفاده از فایروال مسدود می کنند مفید است.

:SSE

رابط EventSource برای دریافت رویدادهای ارسال شده از سرور استفاده می شود. از طریق HTTP به یک سرور متصل می شود و رویدادها را در قالب متن/رویداد جریانی بدون بستن اتصال دریافت می کند.

تفاوت اصلی با polling این است که ما فقط یک اتصال دریافت می کنیم و جریان رویداد را از طریق آن ادامه می دهیم. نظرسنجی طولانی یک اتصال جدید برای هر کشش ایجاد می کند - به علاوه سرفصل ها و سایر مسائلی که در آنجا با آن مواجه بودیم.

رویدادهای ارسال شده توسط سرور رویدادهای بلندرنگی هستند که توسط سرور منتشر شده و توسط مرورگر دریافت می شوند. آنها شبیه WebSockets هستند که در زمان واقعی اتفاق می افتند، اما بسیار شبیه یک روش ارتباطی یک طرفه از طرف سرور هستند. ممکن است عجیب باشد اما پس از بررسی - جریان اصلی داده ما از سرور به مشتری و در موارد بسیار کمتر از مشتری به سرور است.

-۴

ss://asghar:1234!!@ss.myproxy.com:1234\#shadowSocks1

scheme:// user:pass @ host:port\ #frag

-۵

- کد ۵۰۰ یا ۵۰۳ را برگرداند؛ چون خطا از سمت سرور است کد ۵ اولش دارد. بین این دو ۵۰۰ کلی تر است ۵۰۳ به صورت جزئی تر خطا را بیان میکند.
- کد ۴۰۱ به نظرم درست تر می آید به این معنا که احراز هویت نشده اما به نظرم کد های ۴۰۰ یا ۴۰۶ هم میتوانند گزینه های دیگری باشند.
- کد ۳۰۷ (در صورت جابه جایی موقت سایت) و کد ۳۰۱ یا ۳۰۸ (در صورتیکه به صورت مداوم ادرس سایت جابه جا شده باشد).
- کد ۴۲۹ ؛ کاربر درخواست های زیادی داشته
- کد ۲۰۰، ۲۰۱ یا ۲۰۲؛ به ترتیب به معنای اوکی، ساخته شده، و مورد قبول واقع شده هستند
- کد ۴۰۳ که به این معنی که کاربر حق ندارد به آن دسترسی پیدا کند.

-۶

forward proxy -1

reverse proxy -2

reverse proxy -3