

به نام خدا

امیرحسین باریکلو 9730003

تمرین اول

(سوال 1)

http1.0

*این نسخه امنیت کمتری نسبت به بقیه دارد

*پس از دریافت پاسخ قطع میشود

*پاسخ آن میتواند غیر از html باشد و میتواند media ... باشد

مفهوم هدر HTTP هم برای درخواست ها و هم برای پاسخ ها معرفی شد. فراداده می تواند منتقل شود و پروتکل بسیار انعطاف پذیر و قابل توسعه شد.

*داشتن header برای request , response (version number,content type , status code)

http1.1

*یک اتصال می تواند دوباره استفاده شود که باعث صرفه جویی در زمان می شود. دیگر نیازی به باز شدن چندین بار برای نمایش منابع تعبیه شده در سند اصلی نیست.

*خط لوله اضافه شد. این اجازه داد تا قبل از ارسال کامل پاسخ اول، درخواست دوم ارسال شود. این امر تأخیر ارتباط را کاهش داد.

* Chunked responses پشتیبانی شدند.

*مکانیسم های کنترل کش اضافی معرفی شدند.

*مذاکره محتوا، از جمله زبان، رمزگذاری و نوع، معرفی شد. یک مشتری و یک سرور اکنون می توانند در مورد اینکه کدام محتوا را مبادله کنند به توافق برسند.

*به لطف هدر Host، امکان میزبانی دامنه های مختلف از یک آدرس IP یکسان، امکان اشتراک سرور را فراهم می کند.

*در حال حاضر رایج است

http2.0

* این یک پروتکل باینری است تا یک پروتکل متنی. نمی توان آن را به صورت دستی خواند و ایجاد کرد. با وجود این مانع، امکان اجرای تکنیک های بهینه سازی بهبود یافته را فراهم می کند.

* این یک پروتکل چندگانه است. درخواست‌های موازی را می‌توان روی یک اتصال ایجاد کرد و محدودیت‌های پروتکل HTTP/1.x را حذف کرد.

* هدرها را فشرده می‌کند. از آنجایی که اینها اغلب در میان مجموعه‌ای از درخواست‌ها مشابه هستند، این کار تکراری و سربار داده‌های ارسال شده را حذف می‌کند.

* این به سرور اجازه می‌دهد تا از طریق مکانیزمی به نام فشار سرور، داده‌ها را در کش کلاینت پر کند.

http 3.0

* تفاوت اصلی http3 این است که به جای پروتکل لایه ی انتقال TCP ، روی QUIC اجرا میشود که سریعتر و امن تر است.

سوال 1 قسمت 2)

استفاده از QUIC به این معنی است که HTTP/3 به پروتکل Datagram کاربر UDP مبتکی است نه پروتکل کنترل انتقال (TCP) تغییر به UDP اتصالات سریعتر و تجربه کاربری سریعتر را هنگام مرور آنلاین فعال می‌کند.

چون که در پروتکل udp کنترل تراکم و ازدحام را نداریم در نتیجه سرعت ارسال افزایش می یابد (بسته ها منتظر رسیدن بسته های دیگر نمیشوند)

سوال 2)الف

Statefull

* پروتکل های Stateful به سرور نیاز دارند تا وضعیت یک فرآیند را ذخیره کند.

* Stateful منتظر پاسخ است و در صورت عدم دریافت پاسخ، درخواست مجدداً ارسال می‌شود.

* این امر طراحی را سنگین و پیچیده می‌کند زیرا داده‌ها باید ذخیره شوند.

* سرور برای ذخیره و ذخیره اطلاعات وضعیت و جزئیات جلسات مورد نیاز است.

* سرور و کلاینت به شدت به هم مرتبط هستند، همانطور که به شدت به یکدیگر وابسته هستند.

* رسیدگی به تراکنش در پروتکل حالت دار نسبتاً کند است.

* به سرور برای ذخیره وضعیت یک فرآیند نیاز ندارند.

* مشتری درخواستی را به سرور ارسال می کند که سرور بر اساس وضعیت درخواست به آن پاسخ می دهد.

* طراحی سرور در این مورد ساده شده است.

* هیچ سروری برای ذخیره سازی اطلاعات مورد نیاز نیست.

* سرور و کلاینت مستقل تر هستند و از این رو، به طور ضعیفی جفت می شوند.

* این در پروتکل بدون حالت نسبتاً سریعتر است.

ب) از Stateless استفاده میشود

برای حل این مشکل از کوکی ها در سمت کاربر استفاده میشود یا از jwt

کوکی ها اطلاعات کاربر را در سمت کلاینت ذخیره میکند و در صورت نیاز سرور اون ها رو برای سرور ارسال میکند

سوال(3)

در مدل کلاینت سرور ، سرویس گیرنده منابع را از یک سرور درخواست می کند، سپس سرور درخواست را پردازش می کند و داده ها را پس می فرستد .تا زمانی که سرویس گیرنده به ارسال درخواست ها ادامه دهد، سرور به ارسال داده های برگشت ادامه خواهد داد و مشتری به مصرف اطلاعات ادامه خواهد داد.

اما چگونه یک برنامه می داند که یک برنامه دیگر داده های جدید یا به روز شده ای دارد که نیاز به مصرف دارد؟ اگر داده های درخواست شده توسط مشتری هنوز آماده نشده باشد چه؟ و اگر برنامه نیاز به داده های زمان واقعی داشته باشد چه؟ آیا مشتری باید به ارسال درخواست ها ادامه دهد؟

ارسال درخواست ها به سرور نیاز به منابع در سمت مشتری دارد .تصور کنید برنامه بانکی یا برنامه رسانه های اجتماعی شما بر روی تلفن شما همچنان در حال اجرا است تا تازه ترین اطلاعات را داشته باشد.

Webhooks این موضوع را با هشدار دادن به مشتری هر زمان که داده ها برای مصرف آماده است حل و فصل می کنند، تغییری صورت گرفته است، یا رویدادی رخ داده است که حکم تبادل داده ها را صادر می

کند. به جای اینکه برنامه مشتری دائماً از سرور برای داده های جدید یا به روز شده نظرسنجی کند، سیستمی از اعلان های فشار را تصور کنید که کلاینت را به روز نگه می دارد.

(ب)

در واقع، webhook یک API است که به جای «pulling» داده ها، آن ها را «push» می کند.

>به API ها و webhooks به عنوان خواهر و برادر فکر کنید، به همین دلیل است که webhooks گاهی اوقات به عنوان API "معکوس" گفته می شود<<

برای این کار میتوان در سمت سرور راه حل های متفاوتی را بهره گرفت که لز جمله آن ها میتوان به subscribe,publisher اشاره کرد یا اگر عملیاتی ناموفق بود باید دوباره به سرور اطلاع دهیم

.

سوال 4)

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

protocol scheme

user

password

host address(domain)

port

one fragment of page

سوال (5)

internal server error(500) , service unavailable(503)=1

Unauthorized(401)=2

Moved permanetly(301)=3

Too many request(429)=4

Ok(200)=5

Forbidden(403)=6

سوال (6)

Reverse proxy=1

Forward proxy=2

Reverse proxy=3