

سید نامی مدرسی

۹۷۳۳۰۶۹

زمستان ۱۴۰۰

تمرین HTTP

مهندسی اینترنت

سوال ۱)

کلید واژه های هر بخش با underline مشخص شده است.

ویژگی های هر نسخه به شرح زیر است :

HTTP v1.0 :

نوع ارتباط non-persistent است.

در این ارتباط می توانیم هم درخواست ارسال کنیم و هم پاسخ دریافت کنیم.

متد هایی که در این ارتباط ممکن است GET , POST , HEAD هستند.

HTTP v1.1 :

نوع ارتباط در این نسخه به persistent تغییر کرد منتها هنوز به شکل سری کار می کند یعنی یکی پس از دیگری درخواست ها و پاسخ ها را داریم.

نسبت به نسخه قبلی متد های پشتیبانی شده افزایش داشته است.

در این نسخه درخواست ها و پاسخ ها در فایل plaintext نگهداری می شوند.

HTTP v2.0 :

نوع ارتباط در این نسخه مانند نسخه قبلی persistent است اما در این نسخه multiplexing اضافه شد.

درخواست ها همگی می توانند روی یک ارتباط ارسال و دریافت شوند.

درخواست ها و پاسخ ها در فرمت باینری هستند.

HTTP v3.0 :

در این نسخه از QUIC استفاده می شود. (Quick UDP Internet Connection)

دلیل استفاده از UDP :

استفاده از UDP باعث افزایش سرعت در این نسخه شده است زیرا handshake ها را نداریم در زمانی که ارتباط برقرار شده است و در مجموع تعداد handshake کمتری ارسال می شود.

مشکل دیگری هم که رفع شده این است که در نسخه قبلی اگر یک بسته به مقصد نمی رسید آن رشته از اطلاعات نادیده گرفته می شد که دیگر این مشکل وجود ندارد. (head-of-line blocking)

سوال ۲)

الف)

این دسته بندی به این معنی است که ما گذشته (history) یا وضعیت (state) کلاینت ها را نگهداری می کنیم یا خیر. اگر ما وضعیت کلاینت ها را در سرور نگهداری کنیم و وقتی یک کلاینت درخواستی داد بر مبنای گذشته و وضعیت آن کلاینت به آن پاسخ بدهیم در این صورت stateful است اما اگر هیچ وضعیتی از کلاینت ها نداشته باشیم و پاسخ های سرور مستقل از وضعیت و گذشته کلاینت باشد (انگار که هر درخواست را یک کلاینت جدید داده) در این صورت stateless است.

ب)

پروتکل HTTP، stateless است یعنی هیچ وضعیت و گذشته ای از کلاینت ها را نداریم. برا حل این موضوع از دو روش می توانیم استفاده کنیم :

۱ - استفاده از Cookie

۲ - استفاده از (JWT (JSON Web Token

سوال ۳)

الف)

در این مدل از ارتباط کلاینت باید شروع کننده باشد حال فرض کنید قرار است یک پیام اضطراری به همه کاربران داده شود. این حالت ممکن نیست چون در این حالت باید سرور شروع کننده باشد یعنی اگر نیاز باشد پیامی ارسال شود خودکاربران باید درخواست دریافت آن را بدهند! یا مثلا در چت ها تا زمانی که کلاینت درخواست آپدیت شدن چت هایش را ندهد چت هایش آپدیت نمی شود و مثال های دیگر نیز به این شکل است.

پس چالشی که در این حالت وجود دارد این است که تا زمانی که کلاینت درخواست را نداده نمی تواند آن محتوا (پیام اضطراری یا پیام های جدید در چت هایش یا ...) را دریافت کند.

ب)

روش های متفاوتی وجود دارد مثلا یک نمونه آن این است که ارتباط را قطع نکنیم یعنی ارتباط را برقرار نگه داریم تا بتوانیم در صورت نیاز سرور پیام به کلاینت بفرستیم. برای اینکار می توانیم از websocket استفاده کنیم که یک ارتباط full duplex بر بستر http است. راه دیگر استفاده از SSE است (server-sent event). این روش مشابه روش قبل است با این تفاوت که در این حالت تنها نیاز به یک ارتباط یک طرفه است.

یک راه دیگر هم سرکشی متناوب و همیشگی کلاینت است یعنی همیشه مثلا در بازه های زمانی مشخص درخواست بدهیم تا بفهمیم پیامی وجود دارد یا نه که این روش از کارایی خوبی برخوردار نیست و سر بار بسیار زیادی دارد .

سوال ۴)

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

بخش ss نوع پروتکل ارتباطی رو مشکل می کند.
عبارت asghar نام کاربری را نشان می دهد.
عبارت 1234!! رمز این کاربر را نشان می دهد.
بخش ss.myproxy.com مشخص کننده host است.
عدد 1234 پورت را مشخص می کند.
بخش shadowSocks1 هم fragment را مشخص می کند.

سوال ۵)

- ۱ - کد 500 مناسب است. کد 500 برای internal error است و چون دقیقاً مشکل از دیتابیس است یعنی مربوط به سرور است کد 500 مناسب است.
- ۲ - کد 401 مناسب است به این معنی که که عملیات authentication مشکل داشته است که به دلیل وجود نداشتن نام کاربری و رمز عبور در دیتابیس است چون کاربر جدید است.
- ۳ - برای این بخش می توان کد های متفاوتی را داد مانند 308 , 307 , 301 ولی چون وبسایت به جای دیگری منتقل شده است (فرض کردم این منتقل شدن همیشگی است) برای همین کد 308 را خروجی بدهیم بهتر است که در دفعات بعدی خود مرورگر به آدرس جدید درخواست را بزند و دیگر این عملیات درخواست و منتقل شدن دیگر تکرار نشود.
- ۴ - کد 429 مناسب است به این معنی که کاربر در یک بازه زمانی تعداد زیادی درخواست فرستاده است.
- ۵ - کد های 200,201,202 هم سه مناسب هستند ولی با توجه به اینکه کد 200 یکم کلی تر هستش این کد را انتخاب کردم. (البته دو کد دیگر هم کاملاً مناسب هستند)
- ۶ - کد 403 مناسب است به این معنی که کاربر به وبسایت مورد اجازه نظر دسترسی ندارد.

سوال ۶)

- ۱ - باید از reverse proxy استفاده کنیم به این شکل که به جای اینکه درخواست ها به سرور اصلی بروند و پکیج ها را دریافت کنیم سرور پراکسی پکیج ها را به ما می دهد.
- ۲ - باید از forward proxy استفاده کنیم به این شکل که درخواست ها به سرور های پراکسی بروند و سرور ها با توجه به منطقه جغرافیایی کاربران درخواست ها را تقسیم کنند و ارسال کنند.
- ۳ - باید از یک reverse proxy استفاده کنیم زیرا در forward proxy ما اطلاعات سرور را نداریم که بخواهیم از آن برای authentication استفاده کنیم برای همین باید یک reverse proxy قرار بدهیم تا اطلاعات را داشته باشد و کار authentication را انجام بدهد.