

به نام خدا
دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

برنامه نویسی وب

تمرین ۱

سید محمد حجازی حسینی

۹۷۳۳۰۲۰

زمستان ۱۴۰۰

سوال (۱)

(الف)

HTTP 1.0

- پروتکل‌های browser-friendly
- وجود header ها که شامل metadata درباره‌ی هر دو request و response هستند.
- response محدود به hypertext نیست و می‌توان نوع آن را از طریق content-type مشخص کرد.
- متدهای پشتیبانی شده: GET, HEAD, POST
- بعد از response، connection از بین می‌رود.

HTTP 1.1

- در حال حاضر این نسخه به صورت معمول استفاده می‌شود.
- قابلیت‌های جدیدی برای بهینه کردن آن، اضافه شده است؛ مانند persistent and pipelined connections
- متدهای پشتیبانی: GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS
- نوع connection به صورت long-lived است.

HTTP 2.0

- Multiplexing که باعث می‌شود بر خلاف HTTP از یک TCP connection برای ارسال چندین data stream به صورت همزمان استفاده کند. ترتیب response ها یکسان با ترتیب درخواست‌ها نیست و واقعا موازی انجام می‌شوند.
- server push که از این طریق server می‌تواند داده‌هایی را بدون اینکه کاربر از آن درخواست کند به او بفرستد.
- header compression که برای فشرده‌سازی پیام‌ها استفاده می‌شود و پیشرفته‌تر از HTTP 1.1 است (HPACK) و باعث می‌شود چند بایت کمتر به ازای هر packet ارسال شود.

HTTP 3.0

- تفاوت اصلی آن با نسخه‌های قبلی این است که روی پروتکل QUIC به جای TCP قرار دارد. این پروتکل سریع‌تر و امن‌تر از TCP است که مطابق با نیازهای امروزه طراحی شده است.

(ب)

البته از QUIC استفاده می‌شود که مخفف Quick UDP Internet Connections است و در آن از UDP استفاده می‌شود. علت آن سرعت بیشتر آن نسبت به TCP است ولی کمتر reliable است و چندین data stream فرستاده می‌شود که جبران داده‌های از دست رفته در راه شود. در واقع از همان multiplexing استفاده می‌شود. همچنین امنیت بیشتری دارد. معمولا باید از HTTPS برای امنیت بیشتر استفاده کنیم، اما QUIC همیشه از encryption استفاده می‌کند.

سوال (۲)

(الف)

Stateful

stateful application داده‌های مربوط به client session را مانند request های قبلی client، login/authentication status را ذخیره می‌کند. در بعضی از سیستم‌های stateful، این داده‌ها بر روی سرور، جایی که application اجرا می‌شود ذخیره می‌شوند. در سیستم‌های گسترده‌تر این داده‌ها در cache سرور ذخیره می‌شوند. از این داده‌ها برای request های جدید استفاده می‌شود که باعث سرعت بخشیدن به فرآیندها می‌شود.

Stateless

یک stateless application هیچ داده‌ای مربوط به client session را روی سرور، جایی که application اجرا می‌شود ذخیره نمی‌کند. همگی داده‌های مورد نیاز در قسمت back-end database ذخیره یا در cache خود client برای کار با آن ذخیره می‌شود. مانند cookie ها در browser.

(ب)

پروتکل HTTP stateless است. همانطور که در قسمت الف گفته شد، از cache در طرف client استفاده می‌شود تا مشکلات ناشی از آن رفع شود؛ مانند cookie ها.

سوال (۳)

(الف)

از آنجایی webhooks, alerting و چت همگی نیاز به ارسال داده از طرف سرور به کلاینت دارند، بدون اینکه request ای به سرور فرستاده شود، در نتیجه پروتکل HTTP نمی‌تواند این نیازمندی‌ها را فراهم کند. همانطور که در صورت سوال هم گفته شده، این پروتکل client/server است و در ازای یک request یک response وجود دارد که برای این کاربردها باید از polling مداوم استفاده کنیم که بهینه نیست و عملکرد خوبی ندارد.

(ب)

برای مثال در webhook ها در application خود یک URL را که public است به وسیله‌ی یک API یا backend panel مشخص می‌کنیم و آن را به provider می‌دهیم. در صورتی که داده‌ی جدیدی نیاز به ارسال باشد، به این URL فرستاده می‌شود. نوع و ساختار داده را خود provider مشخص می‌کند.

یک روش دیگر استفاده از WebSockets است که connection را alive نگه می‌دارد و باعث می‌شود که data push back با تاخیر کمتری اتفاق افتد. پروتکل WebScket به ما امکان یک ارتباط full-duplex را روی پروتکل HTTP می‌دهد که برای کاربرد های گفته شده مناسب است.

سوال (۴)

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

- Scheme: ss
- Username: asghar
- Password: 1234!!
- Hostname: ss.myproxy.com
 - Subdomain: ss
 - Domain: myproxy.com
 - Tld: com
- Port: 1234
- Hash: #shadowSocks

سوال (۵)

دیتابیس وبسایت شما مشکل پیدا کرده و درخواست های لاگین انجام نمی شوند.

503: Service Unavailable زیرا درخواستی به سرور ارسال شده است که در حال حاضر down است و نمی تواند درخواست را اجرا کند.

نام کاربری ای که کاربر وارد کرده داخل سایت وجود ندارد و لاگین موفقیت آمیز نبود.

401: Unauthorized زیرا کاربر اطلاعات درست برای دسترسی به منبع مورد نیاز را به درستی وارد نکرده است.

وبسایت به دامنه دیگری منتقل شده است.

301: Moved permanently زیرا URL آن برای همیشه عوض شده است، البته اگر موقت است باید از 302: Found استفاده کنیم.

تعداد درخواست هایی که از طرف این کاربر ارسال شده از نرخ مجاز بیشتر است.

429: Too many requests زیرا این کد برای استفاده بیش از مجاز یک API است.

درخواست تمدید توکن (JWT) موفقیت آمیز بوده و توکن تمدید شده داخل هدر بازگردانده شده.

200: OK زیرا درخواست موفق بوده و token جدید برگردانده شده است.

کاربر از رنج IP ممنوعه برای دسترسی به سرویس است. (تحریم شده است)

403: Forbidden این کد برای این است که کاربر حق استفاده از سرویس را ندارد.

سوال ۶)

- استفاده از سرور های داخلی دانشگاه بجای سرور های مرکزی برای دریافت پکیج ها از پکیج منیجر مانند apt برای Forward Proxy :Ubuntu
- تقسیم درخواست ها بر اساس مبدا جغرافیایی درخواست ها: Reverse Proxy
- اضافه کردن قابلیت احراز هویت برای یک سرویس بسیار قدیمی: Reverse Proxy