9장 - HTTP에 기능을 추가한 프로토콜

HTTP를 기본으로 하는 프로토콜

HTTP를 기반으로 추가되어 새로운 포로토콜이 구현되었다.

HTTP의 병목 현상을 해소하는 SPDY

구글이 2010년에 발표한 SPDY(스피디)

- HTTP의 병목 현상을 해소하고 웹 페이지의 로딩 시간을 50% 단축한다는 목표로 개발

HTTP의 병목 현상

- 1개의 커넥션으로 1개의 리퀘스트만 보낼 수 있다.
- 리퀘스트는 클라이언트에서만 시작할 수 있다. 리스폰스만 받는 것은 불가능
- 리퀘스트/리스폰스 헤더를 압축하지 않은 체로 보낸다. 헤더에 정보가 많을 수록 지연
- 장황한 헤더를 보낸다. 매번 같은 헤더를 보내는 것은 낭비
- 데이터 압축을 임의로 선택할 수 있다. 압축해서 보내는 것이 강제적이지는 않다.

Ajax에 의한 해결 방법

Ajax(Asynchronous JavaScript + XML)은 JavaScript나 DOM(Document Object Model) 조작 등을 활용하는 방식

웹 페이지의 일부분만 고쳐쓸 수 있는 비동기 통신 방법

리스폰스로 전송되는 데이터 양은 줄어든다는 장점

핵심 기술

- XMLHttpRequest라는 API로 JavaScript 등의 스크립트 언어로 서버와 HTTP 통신을 할 수 있다.
- 이미 읽어 들인 웹 페이지로부터 리퀘스트를 발행할 수 있기 때문에 페이지의 일부 데이터만 받는 것이 가능

Comet에 의한 해결 방법

서버 측의 콘텐츠에 갱신이 있었을 경우, 클라이언트로부터 리퀘스트를 기다리지 않고 클라이언트에 보내기 위한 방법

응답을 연장시킴으로써 서버에서 통신을 개시하는 서버 푸시 기능을 유사하게 따르고 있다.

리스폰스를 보류 상태로 해 두고, 서버의 콘텐츠가 갱신되었을 때에 리스폰스를 반환한다.

이것에 의해서 서버에서 갱신된 콘텐츠가 있으면 바로 클라이언트에 반영할 수 있다.

SPDY 설계와 기능

TCP/IP의 애플리케이션 계층과 트랜스포트 계층 사이에 새로운 세션 계층을 추가하는 형태로 동작

SPDY는 보안을 위해서 SSL을 사용하도록 되어 있다.

데이터의 흐름을 제어하지만 HTTP의 커넥션은 확립되어 있다.

다중화 스트림

단일 TCP 접속을 통해서 복수의 HTTP 리퀘스트를 무제한으로 처리할 수 있다.

한 번의 TCP 접속으로 리퀘스트를 주고 받는 것이 가능하기 때문에 TCP의 효율이 높아진다.

리퀘스트의 우선 순위 부여

무제한으로 리퀘스트를 병렬 처리할 수 있지만, 각 리퀘스트에 우선 순위를 할당할 수 있다.

복수의 리퀘스트를 보낼 때 대역폭이 좁으면 처리가 늦어지는 현상을 해결하기 위해서이다.

HTTP 헤더 압축

압축으로 인해 보다 적은 패킷 수와 송신 바이트 수로 통신할 수 있다.

서버 푸시 기능

서버에서 클라이언트로 데이터를 푸쉬하는 서버 푸시 기능을 지원

서버 힌트 기능

서버가 클라이언트에게 리퀘스트 해야 할 리소스를 제안할 수 있다.

클라이언트가 자원을 발견하기 전에 리소스의 존재를 알 수 있기 때문에 이미 캐시를 가지고 있는 상태라면 불필요한 리퀘스트를 보내지 않아도 된다.

브라우저에서 양방향 통신을 하는 WebSocket

새로운 프로토콜과 API에 의해 이 문제를 해결하기 위한 기술로서 개발

HTML5의 사양의 일부로서 책정되었지만, 현재는 단독 프로토콜로서 규격 책정이 진행

Websocket의 설계와 기능

웹 브라우저와 웹 서버를 위한 양방향 통신 규격

Ajax나 Comet에서 사용하는 XMLHttpRequest의 결점을 해결하기 위한 기술로서 개발이 진행

Websocket 프로토콜

웹 서버와 클라이언트가 한 번 접속을 확립하면 그 뒤의 통신을 모두 전용 프로토콜로 하는 방식으로 JSON, XML, HTML이나 이미지 등 임의 형식의 데이터를 보내게 된다.

HTTP에 의한 접속의 출발점이 클라이언트에 있다는 것에는 변함이 없지만 한 번 접속을 확립하면 WebSocket을 사용하여 서버와 클라이언트 어느 쪽에서도 송신을 할 수 있게 된다.

서버 푸시 기능

서버에서 클라이언트에 데이터를 푸시하는 서버 푸시 기능을 제공

클라이언트의 리퀘스트를 기다리지 않고 데이터를 보낼 수 있다.

통신량의 삭감

접속을 한 번 확립하면 접속을 유지하려고 한다.

HTTP에 비해 자주 접속을 하는 오버헤드가 감소, 헤더의 사이즈 감소로 인하여 통신량 감소

WebSocket에 의해 통신을 하기 위해서 핸드쉐이크 절차가 필요

핸드쉐이크/리퀘스트

HTTP의 Upgrade 헤더 필드를 사용해서 프로토콜을 변경하는 것으로 핸드쉐이크를 실시

Sec-WebSocket-Key에는 핸드쉐이크에 필요한 키가 저장

Set-WebSocket-Protocol에는 사용하는 서브 프로토콜이 저장

서브 프로토콜은 WebSocket 프로토콜에 의한 커넥션을 여러 개로 구분하고 싶을 때에 이름을 붙여서 정의

핸드쉐이크/리스폰스

앞선 리퀘스트에 대한 리스폰스는 상태 코드 [101 Switching Protocols]로 반환

Sec-WebSocket-Accept는 Sec-WebSocket-Key에서 생성된 값이 저장

핸드쉐이크에 의해 WebSocket 커넥션이 확립된 후에는 HTTP가 아닌, WebSocket 독자적인 데이터 프레임을 이용해 통신을 한다.

등장이 기다려지는 HTTP/2.0

HTTP/2.0의 특징

웹을 이용할 때의 체감 속도의 개선을 목표

다음의 프로토콜이 베이스가 되어 사양이 검토

- SPDY
- HTTP Speed + Mobility
  - 마이크로소프트사가 제안하고 있는 모바일 통신에서 통신 속도와 효율성 개선을 위한 규격
- Network - Friendly HTTP Upgrade
  - 모바일 통신에서 HTTP 효율 개선을 위한 규격

논의되는 7가지 기술

- 다중화
- TLS의 의무화
- 네고시에이션
- 클라이언트 풀/서버 푸시
- 흐름 제어
- WebSocket

웹 서버 상의 파일을 관리하는 WebDAV

WebDav (Web-based Distributed Authoring and Versioning)

웹 서버의 콘텐츠의 대해서, 직접 파일 복사나 편집 작업 등을 할 수 있는 분산 파일 시스템

HTTP/1.1을 확장한 프로토콜로 RFC4918로서 정의

파일 작성이나 삭제 등 기본적인 기능 이외에 파일 작성자 등의 관리나 편집 중에 다른 유저가 다시 고쳐 쓰지 못하도록 잠금 기능, 갱신 정보를 관리하는 리비젼 기능 등이 준비되어 있

다.

HTTP/1.1을 확장한 WebDAV

- Collection(컬렉션)
  - 여러 개의 리소스를 한꺼번에 관리하기 위한 개념
  - 각종 조작은 컬렉션 단위로 할 수 있다.
- Resource(자원)
  - 파일이나 컬렉션을 리소스라고 명칭
- Property(프로퍼티)
  - 리소스의 프로퍼티를 정의
  - “이름 = 값” 의 형식
- Lock(잠금)
  - 파일을 편집할 수 없는 상태로 한다.
  - 여러 명의 사람이 동시에 편집하는 경우 등 동시에 작성되는 걸 예방

WebDAV에서 추가된 메소드와 상태 코드

추가된 메소드

- PROPFIND : 프로퍼티 취득
- PROPPATCH : 프로퍼티 변경
- MKCOL : 컬렉션 작성
- COPY : 리소스 및 프로퍼티 복제
- MOVE : 리소스 이동
- LOCK : 리소스 잠금
- UNLOCK : 리소스 잠금 해제

추가된 상태 코드

- 102 Processing : 리퀘스트는 정상적으로 수신되었지만 아직 처리 중
- 207 Multi-status : 복수의 스테이터스를 가지고 있다.
- 422 Unprocessable Entity : 서식은 올바르지만 내용이 틀리다.

- 423 Locked : 리소스가 잠금되어 있다.
- 424 Failed Dependency : 어떤 리퀘스트와 관련된 리퀘스트가 실패했기 때문에 의존 관계를 유지 못한다.
- 507 Insufficient Storage : 기억 영역이 부족하다.