

Server design  
-leeopop-

# 서버 vs 클라이언트

- 서버
  - 데이터의 생성 및 저장
- 클라이언트
  - 데이터 보여주기
  - 유저의 요청을 서버에 전송

# 프로토콜 (protocol)

- 서버와 클라이언트간의 통신 방법
- 쉽게 말해서 어떠한 통신 채널을 통해서 연속된 문자열을 보낸다고 생각하면 됨

# 프로토콜 예제 (ping)

- 주기적으로 ‘안녕 W0’ 을 전송
  - 시각은 메시지를 전송하는 시각
- ‘안녕’을 받은 쪽에서는 ‘안녕’을 다시 전송
- 
- 문제점 ?

# 이전 프로토콜의 문제점

- 응답 '안녕'과 요청 '안녕'을 구분할 수 없다
- 현재 보낸 '안녕'에 대한 대답인지 이전 대답이 늦게 도착한 것인지 알 수 없다.

# 새로운 안녕

- 보낼 때 : ‘ 안녕 ( 요청 ), 512313’
  - 요청임을 밝히고 이 요청의 고유 id 를 생성
- 응답할 때 : ‘ 안녕 ( 응답 ), 512313’
  - 응답임을 밝히고 어느 요청에 대한 응답인지를 알림

# 파일 전송

- 파일 이름
- 파일 내용
  
- 실습 try

# solution

- [name\_length\_64bit][name\_byte\_array]  
[last\_modified\_64bit]  
[file\_length\_64bit][file\_byte\_array]

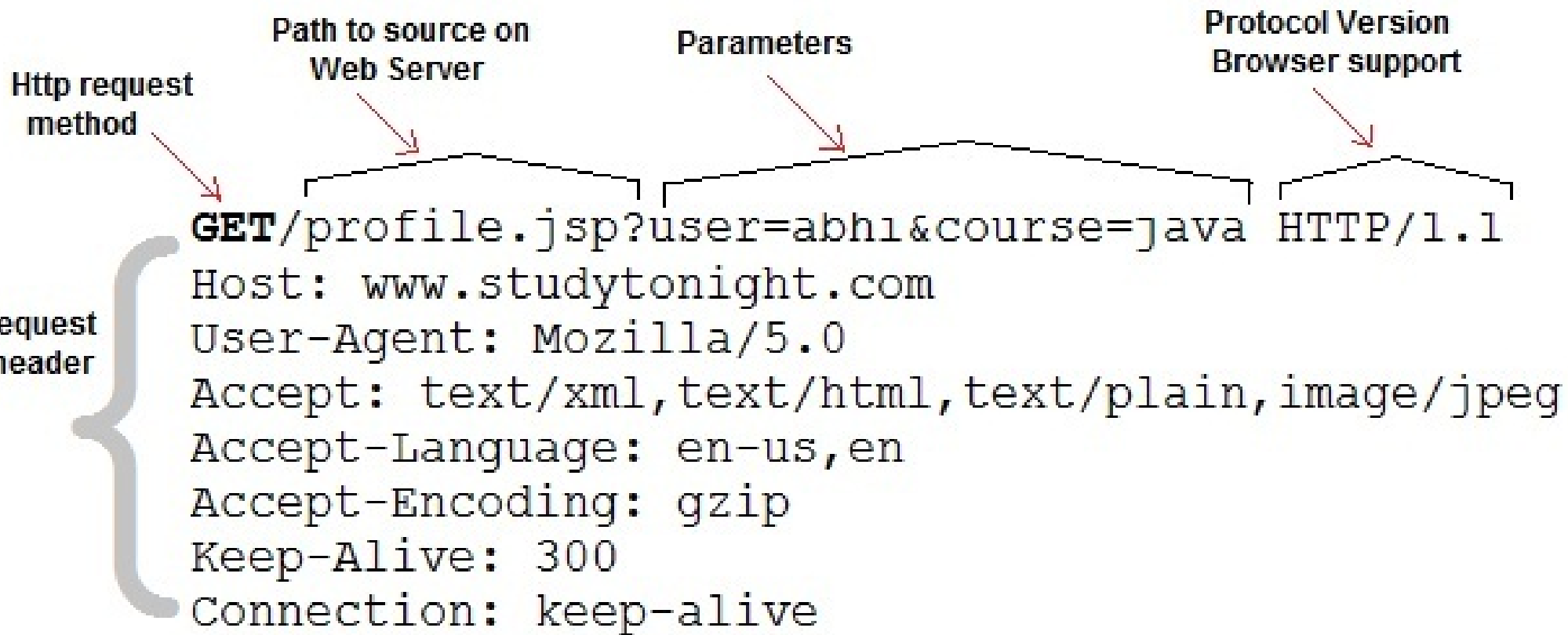


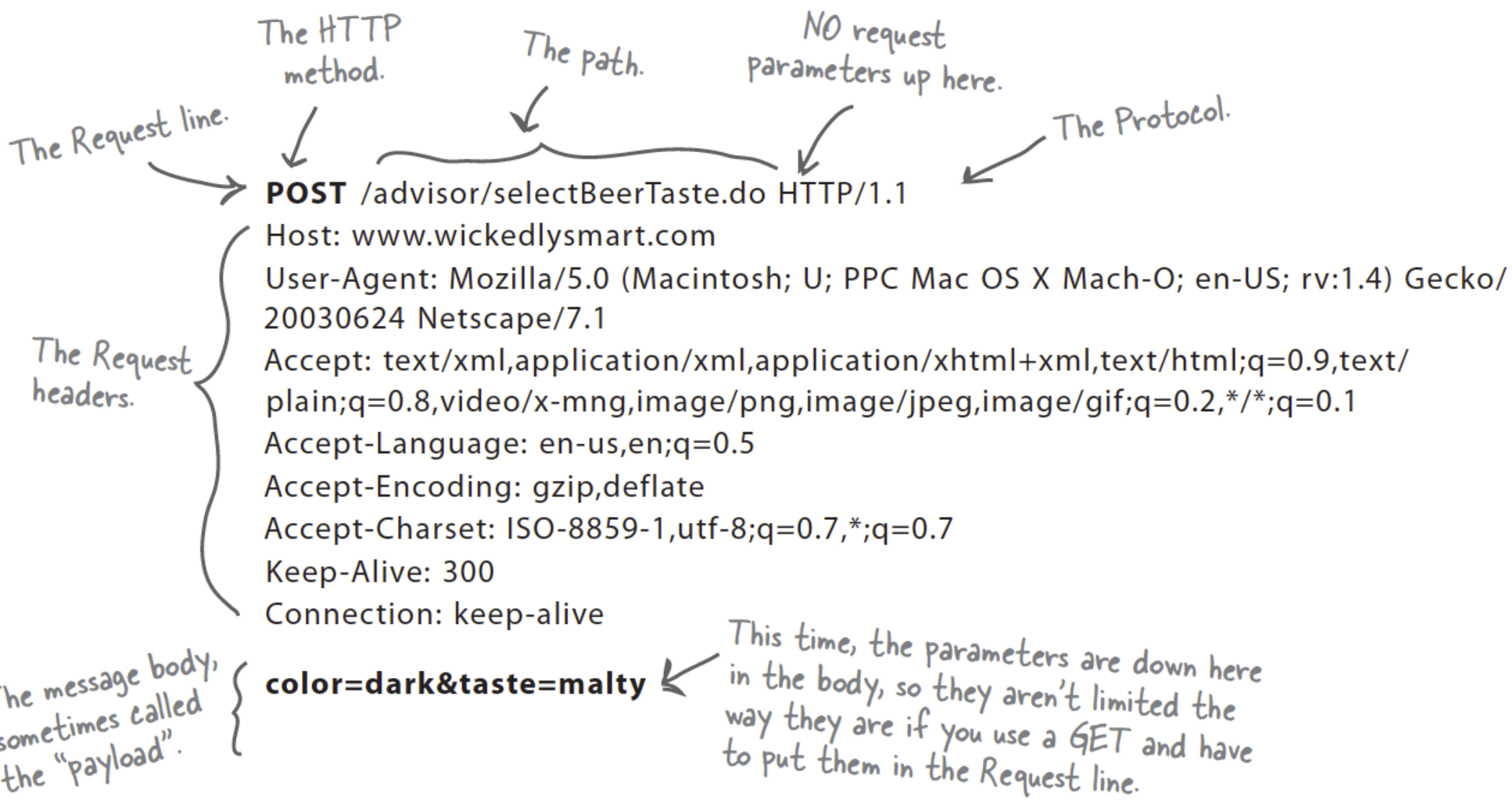
# 프로토콜의 필드를 끝맺는 방법

- Fixed length
  - 처리가 빠르다
  - 구현하기 쉽다
- Termination character
  - 짧은 필드에 대하여 유용
  - 필드의 추가, 삭제가 용이함
- Pre-mentioned length of field
  - 필드의 등장 순서가 중요하다
  - 매우 긴 길이의 내용을 담을 수 있다

# HTTP 를 베이스로 프로토콜 만들기

- Header – termination character
  - 요청의 간단한 옵션을 담을 수 있음
  - 그러나 기존 http 와 헷갈릴 수 있으므로 손대지 않는 것을 추천
- URL – key-value
  - 주로 대부분의 옵션 , 파라미터가 들어가는 곳
- Content – pre-mentioned length
  - 실제 데이터를 담게 될 부분





The HTTP method.

The path.

NO request parameters up here.

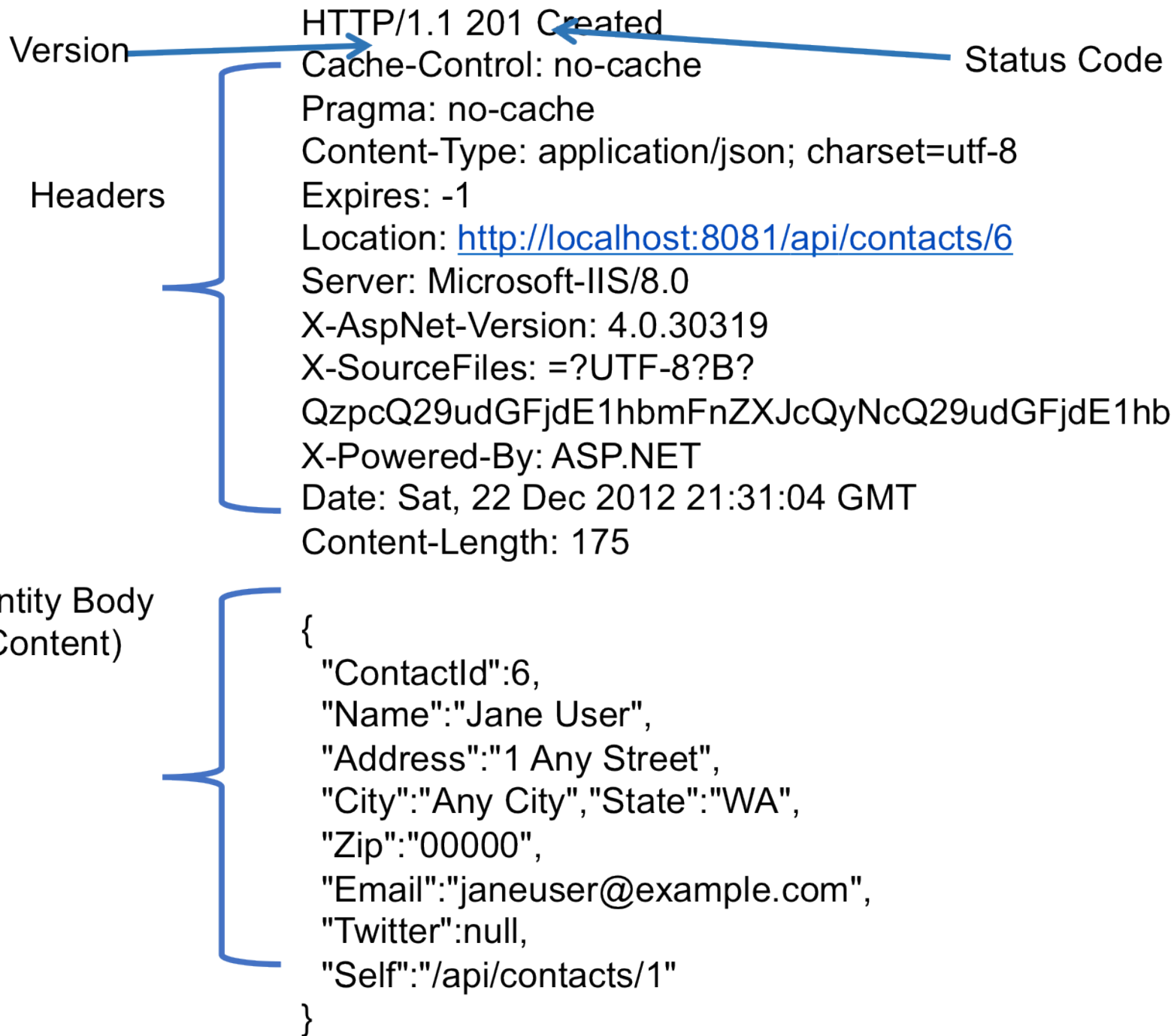
The Protocol.

The Request line.

The Request headers.

the message body, sometimes called the "payload".

This time, the parameters are down here in the body, so they aren't limited the way they are if you use a GET and have to put them in the Request line.



# HTTP 의 또다른 기능들

- 세션
  - 연결이 끊겨도 같은 사용자가 다시 연결한 것인지 구분할 수 있음
- 보안 연결
  - https 를 덧씌우면 자동으로 보안 연결이 됨
  - 프로그래머 입장에서는 http 를 쓰는 것과 동일
- 캐싱
  - 이미 받은 데이터는 절약해서 안받을 수 있음 (클라이언트에서 처리)
  - 이미 생성한 데이터를 또 생성하지 않고 재활용 (서버에서 처리)

# 프로토콜 설계 (ftp)

- 로그인 , 로그아웃
- 파일 받기
- 파일 올리기
- 파일 지우기
- 디렉토리 탐색 (ls)

# DB

- Table – tuple + dictionary
  - Tuple: columns
  - Dictionary: primary keys
- Index
  - Constraint
  - Search
- Foreign key
  - Pointer to other table's row



# 유저 관리

- 아이디
- 비밀번호
  - Plain
  - Hash
  - Hash + Salt
- \*optional: 유저 고유 번호

# FTP Template

- 파일 브라우징
- 파일 업로드
- 파일 다운로드

# 실습 1: ftp 에 로그인 기능 추가

- 로그인 , 로그아웃
- 파일 브라우징 시 , 올린 사람 아이디로 보이도록
- 삭제는 올린 사람만이 가능
- 다운로드 는 제한없이 가능
- 로그인 하지 않으면 업로드 금지
- 상위 디렉토리 소유자는 하위 디렉토리 마음대로 삭제 가능 ( 곧 , 남의 디렉토리에 업로드 가능 )

## 실습 2. 메신저 기능 추가

- 그룹챗 기능
- 자신이 속한 그룹 리스트를 모두 보여줌
- 유저 1, 유저 2, 유저 3, 에 메시지를 보내면 자동으로 자신과 세 유저를 포함하는 그룹 생성
  - 이미 그 그룹이 있으면 재사용
- 그룹에 유저 추가, 제거 자유롭게 가능
- 그룹을 나가면 더이상 메시지를 받지 못함
- 그룹에 다시 들어오면 이전 메시지에 접근 가능  
(없는 동안 오고간 대화는 볼 수 없음)