

가상 면접 사례로 배우는  
대규모 시스템 설계 기초

이정민

## 8. URL 단축기 설계

# 1단계) 문제 이해 및 설계 범위 확정

## ■ 시스템의 기본기능

- URL 단축 : 주어진 긴 URL을 훨씬 짧게 줄임
- URL 리디렉션 : 축약된 URL로 HTTP 요청이 오면  
원래 URL로 안내
- 높은 가용성과 규모 확장성, 장애 감내가 요구됨

## ■ 개략적 추정

- 쓰기 연산 : 매일 1억 개의 단축 URL 생성
- 초당 쓰기 연산 :  $1\text{억}/24/3600 = 1160\text{회}$
- 읽기 연산 : 읽기 연산과 10:1 비율 시 11600회
- 10년 운영 시 :  $1\text{억} * 365 * 10 = 3650\text{억개}$  레코드
- 축약 전 URL의 평균 길이는 100
- 10년 동안 저장 용량 :  $36.5\text{TB}$  ( $3650\text{억} * 100\text{바이트}$ )

## ■ ID는 64비트로 표현될 수 있어야 함

## ■ ID는 발급 날짜에 따라 정렬 가능해야 함

## ■ 초당 10,000개의 ID를 만들 수 있어야 함

## 2단계) 개략적 설계안 제시 및 동의

### ■ API 엔드 포인트(REST)

- URL 단축용 엔드포인트 : POST /api/v1/data/shorten
  - 파라미터 : { longUrl : longURLstring }
  - 리턴 : 단축 URL
- URL 리디렉션용 엔드포인트 : GET /api/v1/shortUrl
  - 리턴 : HTTP 리디렉션 목적지가 될 원래 URL

## 2단계) 개략적 설계안 제시 및 동의

### ■ URL 리디렉션

- 응답 헤더 **location**에 리디렉션 될 **URL**을 넣어 응답
- **Status Code 301 Permanently Moved** : 영구적으로 **URL** 이전. 브라우저가 캐시하여 추후 사용
  - 서버 부하를 줄임
- **Status Code 302 Found**: 일시적으로 **URL**로 리디렉션
  - 트래픽 분석 시 추적에 유리
- 구현 방법 : 해시 테이블 사용 (**key**: 단축 **URL**, **value**: 원래 **URL**)

## 2단계) 개략적 설계안 제시 및 동의

### ■ URL 단축

- 긴 URL을 해시 값으로 대응시킬 해시 함수 필요
  - URL이 다르면 해시값도 달라야 함
  - 해시 값은 원래 입력으로 주어진 URL로 복원 될 수 있어야 함

## 3단계) 상세 설계

### ■ 데이터 모델

- 관계형 데이터베이스 사용
  - id(PK)
  - shortURL
  - longURL

## 3단계) 상세 설계

### ■ 해시 함수

- 해시 값 길이 : 7글자
  - [0-9, a-z, A-Z] 사용 가능 문자 개수는 62개 ( $10 + 26 + 26$ ), 62의  $n$ 승  $\geq 3650$ 억인  $n$ 의 최소값
- 해시 후 충돌 해소 기법 : 원래 URL을 7글자로 줄이는 해시 함수 사용
  - CRC32, MD5, SHA-1 해시함수 사용 가능하나 7글자로 표현 불가능, 처음 7개만 이용하여 계산 (DB에 저장)
  - DB 값과 중복 발생 시 사전 정의 문자열 추가하여 재계산 (DB 질의 성능문제, 블룸필터 사용가능)
- base-62 변환 기법 : ID를 62진법으로 62개 문자로 표현 (0->0, 10->a, 36->A, 61->Z)
- 해시후 충돌 해소 기법 vs base-62 변환 기법
  - 길이 : 고정 / 가변, 유일성 보장 ID 필요 : 없음 / 있음, 충돌해소 필요 : 있음 / 없음, URL 유추 : 불가능 / 가능



## 3단계) 상세 설계

### ■ URL 단축기 상세 설계

1. 긴 URL 입력

2. DB에 해당 URL 있는 지 검사

2-a. 데이터베이스에 있으면 단축 URL을 리턴. 종료

2-b. 데이터베이스에 없으면 유일 ID 생성 (DB 기본키)

3. base-62로 ID를 단축 URL로 생성

4. ID, 단축 URL, 원래 URL을 DB에 저장하고, 단축 URL을 리턴. 종료

## 3단계) 상세 설계

### ■ URL 리디렉션 상세 설계

1. 사용자가 단축 URL을 클릭
2. 로드 밸런서가 해당 클릭으로 발생한 요청을 웹 서버에 전달
3. 단축 URL이 캐시에 있는 경우 원래 URL을 바로 리턴. 종료
4. 캐시에 해당 단축 URL이 없는 경우 DB에서 조회 (DB에 없는 경우 잘못된 단축 URL을 사용한 경우)
5. DB에서 조회한 URL을 캐시에 저장 후 원래 URL을 리턴. 종료

## 4단계) 마무리

### ■ 추가 논의

- 처리율 제한 장치(**rate limiter**) : URL 단축 요청 집중 시 무력화 될 수 있는 상황방지 (IP 필터링 등)
- 웹 서버의 규모 확장 : 본 시스템은 무상태이므로 웹 서버를 자유롭게 증설/제거 가능
- 데이터베이스의 규모 확장 : 데이터베이스 다중화 또는 샤딩 적용 가능
- 데이터 분석 솔루션 : URL 단축기에 데이터 분석 솔루션을 통합하여 주요 정보 분석 가능
- 가용성, 데이터 일관성, 안정성