

# REST

## 리소스 원형

- 컬렉션 리소스 (복수명사)
  - 컬렉션 리소스는 서버에서 관리하는 디렉터리라는 리소스다. 클라이언트는 새로운 리소스를 제안해서 컬렉션에 포함시킬 수 있다. 그러나 새로운 리소스를 생성할지는 컬렉션에 달려 있다. 컬렉션 리소스에 포함하고 싶은 것을 선택하고, 포함된 각 리소스의 URI를 결정한다.
  - 예
    - <http://api.soccer.restapi.org/leagues>
    - <http://api.soccer.restapi.org/leagues/seattle/teams>
    - <http://api.soccer.restapi.org/leagues/seattle/teams/trebuchet/players>
- 도큐먼트 리소스 (단수명사)
  - 도큐먼트 리소스는 객체 인스턴스나 데이터베이스 레코드와 유사한 단일 개념이다. 일반적으로 도큐먼트의 상태 표현은 값을 가진 필드와 다른 관련 리소스와의 링크 둘 다를 가지게 된다. 기본적인 필드와 링크 기반 구조로 인해, 도큐먼트 타입은 다른 리소스 원형들의 기반 원형이 된다. 즉, 서로 다른 리소스 원형 세 개는 도큐먼트 원형에서 분리된 것이라 볼 수 있다.
  - 예
    - <http://api.soccer.restapi.org/leagues/seattle>
    - <http://api.soccer.restapi.org/leagues/seattle/teams/trebuchet>
    - <http://api.soccer.restapi.org/leagues/seattle/teams/trebuchet/players/mike>
- 컨트롤러 리소스 (동사, 동사구)
  - 컨트롤러 리소스는 절차적인 개념을 모델화한 것이다. 컨트롤러 리소스는 실행 가능한 함수와 같아서 파라미터(입력 값)와 반환 값(출력 값)이 있다. 전통적인 웹 애플리케이션이 'HTML form'을 사용하듯이, REST API는 CRUD라고 알려진 표준적인 메서드와는 논리적으로 매핑되지 않는 애플리케이션 고유의 행동을 컨트롤러 리소스의 도움을 받아 수행한다. 일반적으로 컨트롤러 이름은 URI 경로의 제일 마지막 부분에 표시되며, 계층적으로 뒤따르는 자식 리소스는 없다.
  - POST /alerts/245743/resent

## URI 형태

- 슬래시 구분자(/)는 계층 관계를 나타내는 데 사용한다.
- URI 마지막 문자로 슬래시(/)를 포함하지 않는다.
- 하이픈(-)은 URI 가독성을 높이는 데 사용한다.
- 밑줄(\_)은 URI에 사용하지 않는다.
- URI 경로에는 소문자가 적합하다.
- 파일 확장자는 URI에 포함시키지 않는다.

## URI Query 디자인

- URI 쿼리 부분으로 컬렉션이나 스토어를 필터링 할 수 있다.
- URI 쿼리는 컬렉션이나 스토어의 결과를 페이지로 구분하여 나타내는데 사용해야 한다.

## HTTP를 이용한 인터렉션 설계

- 요청 메서드
  - GET 메서드나 POST 메서드를 사용하여 다른 요청 메서드를 처리해서는 안 된다.

- GET 메서드는 리소스의 상태 표현을 얻는 데 사용해야 한다.
- 응답 헤더를 가져올 때는 반드시 HEAD 메서드를 사용해야 한다.
- PUT 메서드는 리소스를 삽입하거나 저장된 리소스를 갱신하는 데 사용해야 한다.
- PUT 메서드는 변경 가능한 리소스를 갱신하는 데 사용해야 한다.
- POST 메서드는 컬렉션에 새로운 리소스를 만드는 데 사용해야 한다.
- POST 메서드는 컨트롤러를 실행하는 데 사용해야 한다.
- DELETE 메서드는 그 부모에서 리소스를 삭제하는 데 사용해야 한다.
- OPTIONS 메서드는 리소스의 사용 가능한 인터렉션을 기술한 메타데이터를 가져오는 데 사용해야 한다.
- 응답 상태 코드
  - 200("OK")는 일반적인 요청 성공을 나타내는 데 사용해야 한다.
  - 200("OK")는 응답 바디에 에러를 전송하는 데 사용해서는 안 된다.
  - 201("Created")는 성공적으로 리소스를 생성했을 때 사용해야 한다.
  - 202("Accepted")는 비동기 처리가 성공적으로 시작되었음을 알릴 때 사용해야 한다.
  - 204("No Content")는 응답 바디에 의도적으로 아무것도 포함하지 않을 때 사용한다.
  - 301("Moved Permanently")는 리소스를 이동시켰을 때 사용한다.
  - 302("Found")는 사용하지 않는다.
  - 303("See Other")은 다른 URI를 참조하라고 알려줄 때 사용한다.
  - 304("Not Modified")는 대역폭을 절약할 때 사용한다.
  - 307("Temporary Redirect")는 클라이언트가 다른 URI로 요청을 다시 보내게 할 때 사용해야 한다.
  - 400("Bad Request")는 일반적인 요청 실패에 사용해야 한다.
  - 401("Unauthorized")는 클라이언트 인증에 문제가 있을 때 사용해야 한다.
  - 403("Forbidden")은 인증 상태에 상관없이 액세스를 금지할 때 사용해야 한다.
  - 404("Not Found")는 요청 URI에 해당하는 리소스가 없을 때 사용해야 한다.
  - 405("Method Not Allowed")는 HTTP 메서드가 지원되지 않을 때 사용해야 한다.
  - 406("Not Acceptable")은 요청된 리소스 미디어 타입을 제공하지 못할 때 사용해야 한다.
  - 409("Conflict")는 리소스 상태에 위반되는 행위를 했을 때 사용해야 한다.
  - 412("Precondition Failed")는 조건부 연산을 지원할 때 사용한다.
  - 415("Unsupported Media Type")은 요청의 페이로드에 있는 미디어 타입이 처리되지 못했을 때 사용해야 한다.
  - 500("Internal Server Error")는 API가 잘못 동작할 때 사용해야 한다.

자 이제 <https://developer.github.com/v3/pulls/> 을 함께 분석해 보자.