



스터디 9주차

2021.10.10

섹션 6: 스프링 MVC – 기본 기능

섹션 7: 스프링 MVC – 웹 페이지 만들기

주 제

리다이렉션(Redirection)

“http 웹 기본 지식” 강의 내용을 재구성 하였습니다.
잘못된 내용은 피드백 주세요

리다이렉션(Redirection)

- 웹 브라우저는 3XX 응답의 결과에 Location 헤더가 있으면, Location 위치로 자동 이동
 - ➔ 리다이렉트(re + direct = 다시 지시하는 것)
- 영구 / 일시 / 특수(기타) 리다이렉션

예를 들어, 브라우저가 `www.test.com/page1` URL을 웹 서버에 요청

➔ 서버는 HTTP 응답 메시지를 통해 `www.test.com/page2` 로 다시 요청하라고 브라우저에게 다른 URL을 지시

영구 리다이렉션

- 특정 리소스의 URI가 영구적으로 이동
- 원래의 URL 사용 안함.
- 검색 엔진 등에서도 변경을 인지한다

301 Moved Permanently

- 리다이렉트시 요청 메서드가 POST라면 GET으로 변하고, 본문이 제거될 수 있다.(MAY)

GET으로 변할 때 있고, 아닐 때 있고..
본문이 제거될 때 있고, 아닐 때 있고..
근데 대부분 변하고 제거됨

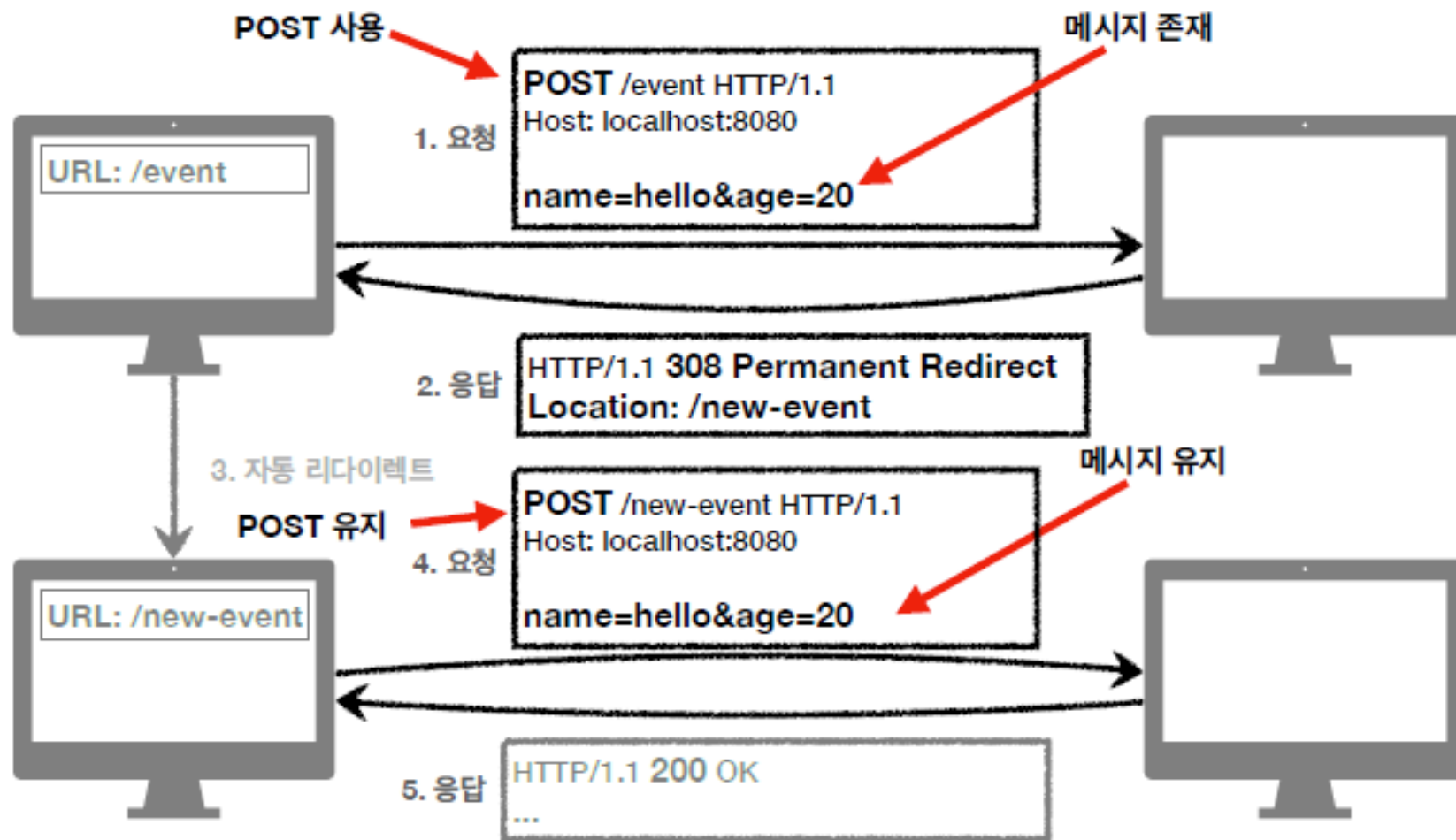
308 Permanent Redirect

- 301에서의 문제를 308로 해결
- 301과 기능은 같음
- 처음 요청 메서드가 POST라면 리다이렉트도 POST로 유지되고, 본문 내용(HTTP body)도 유지

301 흐름



308 흐름



- 그러나! 이건 스펙 내용인거고, 실무적으로는 308 사용하지 않는다
- 왜냐하면 /event → /new event로 바뀔 때 어차피 내부적으로 전달해야 하는 데이터는 전부 바뀌게 된다!
→ 그러므로 웬만하면 post로 와도 get으로 돌리는 게 낫다 (= 301)
- 원래 처음 301 스펙을 설계할 때 post로 보내면 post로 응답하는 거로 생각했는데 막상 브라우저에서는 대부분이 get으로 반환한다.
- 그래서 http를 깰 수 없어서 나중에는 스펙을 바꾼 것이다. (308의 등장)

일시 리다이렉션

- 리소스의 URI가 일시적으로 변경
- 따라서 **검색 엔진(크롤링)** 등에서 URL을 변경하면 안됨
→ 다르게 말하면 클라이언트 입장에서는 별 차이가 없다

302 Found

- 리다이렉트시 요청 메서드가 POST라면 GET으로 변하고, 본문이 제거될 수 있다.(MAY)

307 Temporary Redirect

- 302와 기능은 같음
- 리다이렉트시 요청 메서드가 POST라면 무조건 POST여야 하고 본문 내용이 **유지**됨. (MUST NOT)
요청 메서드를 변경하면 안된다

303 See Other

- [illegible]

PRG: Post/Redirect/Get - 1

- Post로 상품을 등록하여 저장 후 웹 브라우저를 새로고침하면 중복 등록이 되는 것을 확인할 수 있다.

상품 상세

상품 ID

3

상품명

Smart TV

가격

5000

수량

3

상품 수정

목록으로

상품 상세

상품 ID

4

상품명

Smart TV

가격

5000

수량

3

상품 수정

목록으로

▼ General

Request URL:

http://localhost:8080/basic/items/add

Request Method:

POST

Status Code:

● 200

Remote Address:

[::1]:8080

Referrer Policy:

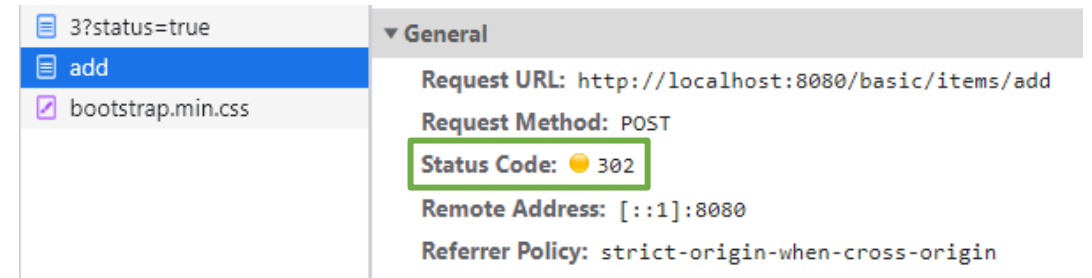
strict-origin-when-cross-origin

- 새로 고침이란 마지막에 했던 행위를 다시 하는 것
➔ 'POST /add + 상품 데이터' 를 계속 재실행해서 계속 쌓이게 되는 것이다.
- 클라이언트와 서버는 POST로 주문 후 새로 고침으로 인한 중복 주문을 방지해야 한다.
- 클라이언트 차원에서 방지하는 방법이 바로 PRG

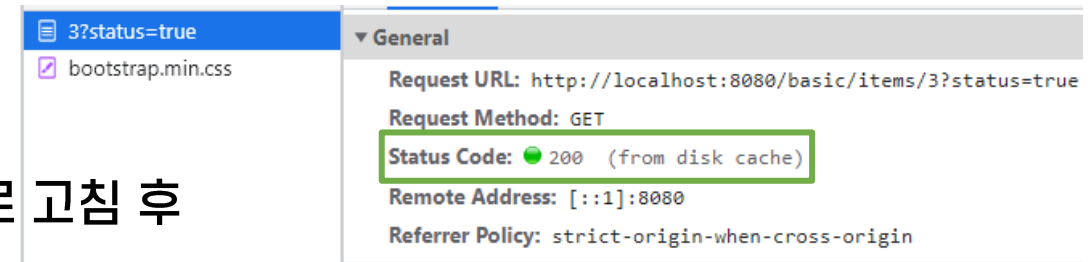
PRG: Post/Redirect/Get – 2

- 앞서 말했듯이 새로 고침은 마지막에 서버에 전송한 데이터를 다시 전송하는 것이므로 상품 저장 후에 뷰 템플릿으로 이동하는 것이 아닌 **상품 상세 화면으로 리다이렉트 호출**
→ 마지막 호출 메소드가 **GET**이 된다.
- GET은 **조회하는 목적을 가진 메소드**이기 때문에 새로 고침을 해도 아무런 변화가 없다.

```
@PostMapping("/add")
public String addItemV6(Item item, RedirectAttributes redirectAttributes) {
    Item savedItem = itemRepository.save(item);
    redirectAttributes.addAttribute("itemId", savedItem.getId());
    redirectAttributes.addAttribute("status", true);
    return "redirect:/basic/items/{itemId}";
}
```



새로 고침 후



- 서버 차원에서 예방하는 방법은 같은 **주문 번호를 체크**하는 방식 등이 있다.

기타 리다이렉션

300 Multiple Choices

- 서버가 요청에 따라 여러 조치를 응답으로 선택할 수 있다.
- 서버가 사용자 에이전트에 따라 수행할 작업을 선택하거나, 요청자가 선택할 수 있는 작업 목록을 제공한다.
- 응답 중 하나를 선택하는 방법이 표준화되어 있지 않다.
- 결론 → 사용 안함

304 Not Modified

- 진짜 많이 사용
- 클라이언트에게 서버가 리소스가 수정되지 않았다고 캐시에 있는거 재사용해서 쓰라고 알려준다.(캐시로 리다이렉트)
- 로컬 캐시를 사용해야 해서 응답에 메시지 바디를 포함하면 안된다!
- 조건부 GET, HEAD 요청 시 사용한다.