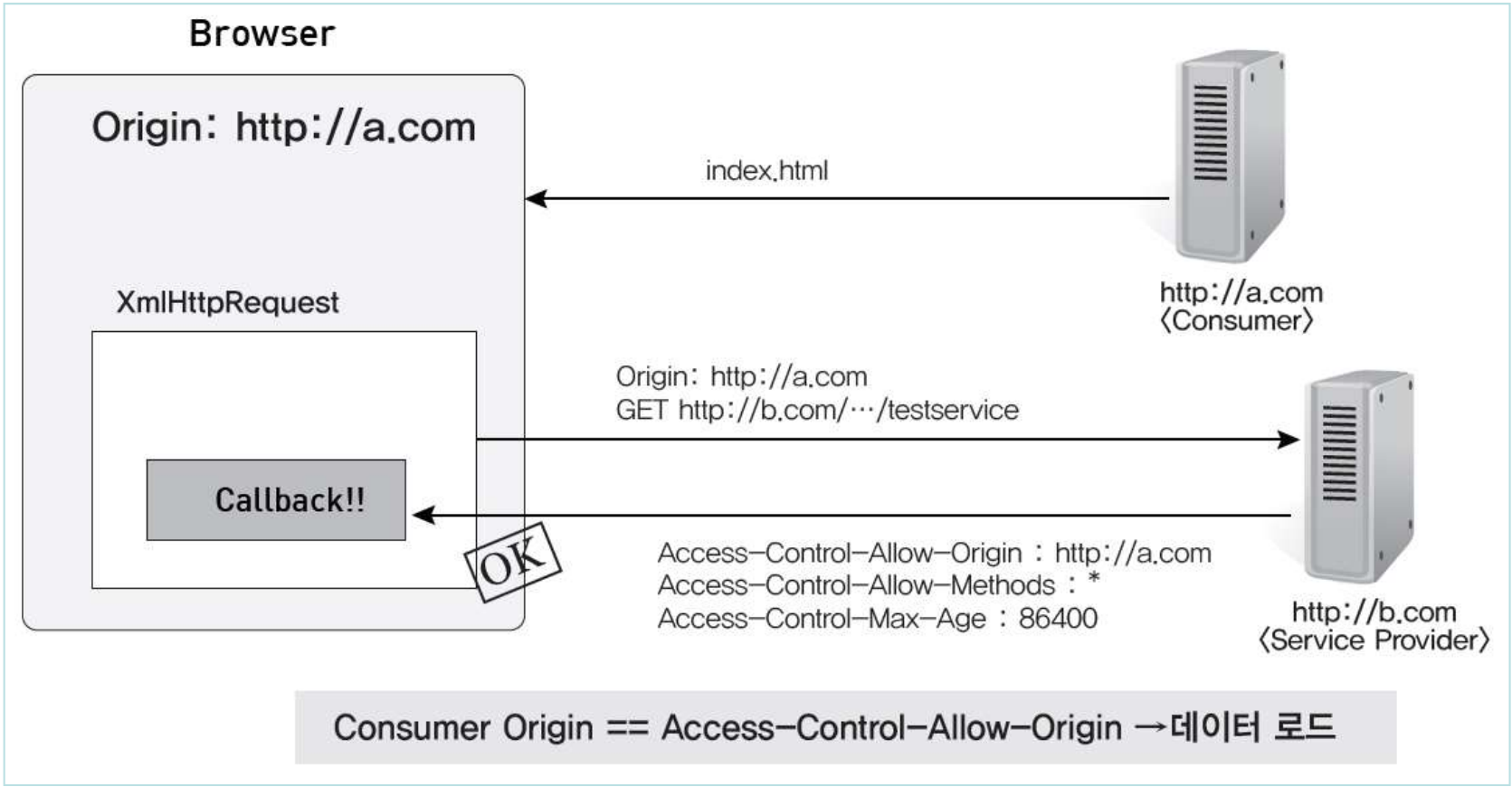
4.3 CORS(1)

CORS(Cross Origin Resource Sharing)란?

- "서비스 제공자(Service Provider)가 허락하는 경우에 브라우저가 데이터를 로드할 수 있도록 한다."
  - 서비스 제공자가 Access-Control-Allow-Origin 헤더로 브라우저의 오리진을 지정해 응답하면 브라우저가 거부하지 않고 데이터를 로드할 수 있음
  - 서비스 제공자는 브라우저가 전송해온 Origin 헤더, Referer 헤더를 확인해 컨슈머를 확인하는 것이 바람직하다.
- 브라우저는 자신의 오리진과 Access-Control-Allow-Origin 헤더값을 비교해 일치하면 데이터를 로드함
- Access-Control로 시작하는 헤더를 이용해 추가적인 정보를 제공할 수 있음
  - Access-Control-Allow-Methods : 이 값을 GET으로 지정하면 GET 방식만 지원

4.3 CORS(2)

개념도


Browser
Origin: http://a.com
XmlHttpRequest
Callback!!
OK
index.html
http://a.com
〈Consumer〉
Origin: http://a.com
GET http://b.com/…/testservice
Access-Control-Allow-Origin : http://a.com
Access-Control-Allow-Methods : *
Access-Control-Max-Age : 86400
http://b.com
〈Service Provider〉
Consumer Origin == Access-Control-Allow-Origin →데이터 로드

4.3 CORS(3)

Simple Request

- Preflight Request를 수행하지 않는 경우
- 다음 3가지 조건을 만족시키면 Simple Request
  - GET, HEAD, POST 중의 한 가지 방식을 사용해야 함.
  - POST 방식일 경우 Content-type이 아래 셋 중의 하나여야 함.
    - application/x-www-form-urlencoded
    - multipart/form-data
    - text/plain
  - 커스텀 헤더를 전송하지 말아야 함.
- 한번의 요청과 한번의 응답으로 종료

4.3 CORS(4)

Preflight Request

- Simple Request 조건에 해당되지 않는 경우
- 예비 요청에서 OPTIONS 메서드로 요청
  - 서버는 가능한 요청 방식을 제공함.
  - 클라이언트는 서버로부터 응답받은 정보를 바탕으로 본 요청을 수행함.

더 자세한 내용은 다음 페이지 참조

- Mozilla 공식 문서
  - https://developer.mozilla.org/ko/docs/Web/HTTP/Access_control_CORS
- 오명운님 블로그
  - https://homoefficio.github.io/2015/07/21/Cross-Origin-Resource-Sharing/

4.3 CORS(5)

CORS 작동 과정

- 참고 : https://www.securityninja.io/understanding-cross-origin-resource-sharing-cors/