

CORS Preflight Request

개요

PROJ-80 단일상품 등록 중, Mobile SO 를 구현하면서 생긴, CORS Preflight request 표준에 대한 이슈를 정리한 문서

Preflight Request는 무엇인가?

w3c가 2014년 초에 낸 웹 표준으로, 익히 잘 알고 있는 크로스도메인 AJAX 통신 spec이 존재하는 CORS(Cross-Origin Resource Sharing) 에 명시되어있다.

영어문서로는 당연히 w3c의 페이지 <https://www.w3.org/TR/cors/> 가 표준이고

한글문서로는 https://developer.mozilla.org/ko/docs/Web/HTTP/Access_control_CORS 추천.

크로스 도메인 간, Access-Control-Allow-Origin: * 헤더를 붙여서, 다른 도메인에서도 AJAX 요청을 주고 받을 수 있는 방법은 다들 경험해 본 적이 많을 것이다.

허나, 이번 Mobile SO를 개발하며, Preflight Request가 발생하는 경우를 체험해 보았고, 이에 대해 약간의 시간을 할애하여 Preflight Request가 무엇인지, 어떤 상황에서 발생하는지 정리가 필요해 보여 정리를 시작 함.

Mobile SO에서의 Preflight Request 발생 환경

front영역 프레임워크 없이 일반적으로 사용하는 웹 환경에서 우리가 AJAX 통신을 할 때에는, XMLHttpRequest나, jQuery의 \$.ajax 를 이용하는 경우가 99.99%.

물론 우리가 구현한 Mobile SO 도 그 범위에서 크게 다르진 않으나, 한가지 정책에 의해 Preflight Request에 대한 고려를 강제로 해야 했음

- Form전송(application/x-www-form-urlencoded or multipart/form-data) 지양. 모든 HTTP Request & Response body 는 JSON(application/json)으로 한다.

모든 통신을 JSON으로 한다는 것은 Content-Type 가 application/json으로 되어 통신한다는 것이고, 이것은 CORS표준에 의거, Preflight request조건에 들어맞는 케이스임.

그럼 어떤 상황에 Preflight Request가 발생할까

일단, cross origin 상황에서, 클라이언트가 날리는 request의 종류는 크게 2가지로 나뉜다.

Simple Request, Preflight Request

Simple Request는 다음과 같은 spec의 조건에 부합해야 함.

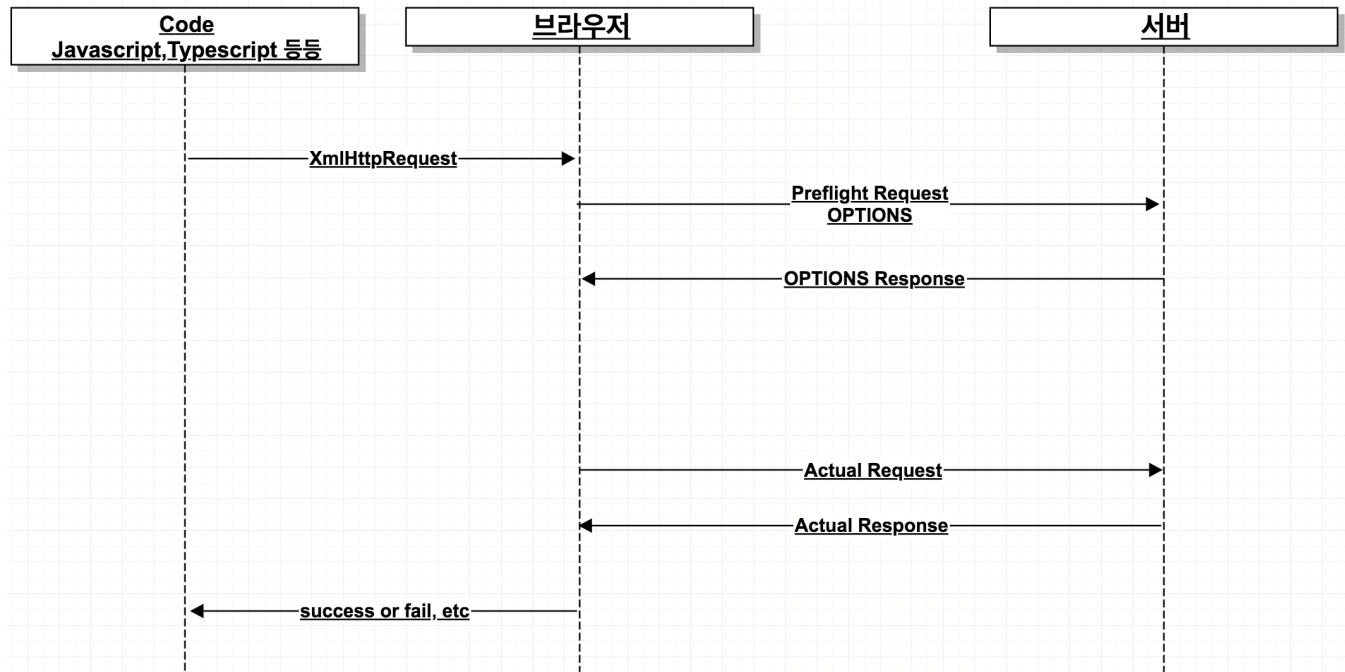
1. (당연히) 요청하는 주소가 Cross-domain이어야 함
2. HTTP Method는 다음 세가지 중 하나
 - GET
 - HEAD
 - POST
3. Content-Type Header 가 다음 셋중의 하나여야 한다
 - application/x-www-form-urlencoded
 - multipart/form-data
 - text/plain
4. 브라우저(정확히 말하면 User Agent)가 기본으로 내려주는 값을 제외하고, 사용가능한 헤더
 - Accept (<https://developer.mozilla.org/ko/docs/Web/HTTP/Headers/Accept>)
 - Accept-Language (<https://developer.mozilla.org/ko/docs/Web/HTTP/Headers/Accept-Language>)
 - Content-Language (<https://developer.mozilla.org/ko/docs/Web/HTTP/Headers/Content-Language>)
 - Content-Type (<https://developer.mozilla.org/ko/docs/Web/HTTP/Headers/Content-Type>)
5. 또한, 상기 허용한 헤더를 제외하고, 임의로 설정한 표준이 아닌 헤더가 존재하지 않아야 함

이 규칙에서 예외가 하나라도 존재한다면 모두 다 Preflight Request 라고 간주하면 된다.

예)

- 메소드는 GET이지만, 커스텀헤더 'Hello World' 라는 헤더가 포함된 Request를 날리는 경우
 - 조건 5에 부합하지 않음 → Preflight Request
- 메소드가 POST이고, 별도의 커스텀 헤더가 없으면서, Content-Type: application/x-www-form-urlencoded 로 Request 를 날리는 경우
 - 모든 Simple Request 조건에 부합함 → Simple Request
- 메소드가 DELETE이고, 별도의 커스텀 헤더가 있고, Content-Type: application/json 로 Request 를 날리는 경우 ()
 - 조건 2,3,5 에 부합하지 않음 → Preflight Request

그럼 어떤 메커니즘으로 Preflight Request가 발생하느냐



그럼 Preflight Request 는 어떻게 구현해야 하나?

- 클라이언트
Preflight Request 요건에 맞는 Request를 클라이언트가 작성하면, 브라우저가 알아서 판단하여 날려줌
심지어, Actual Request 를 날릴 수 있는지 & 날릴지 말지에 대한 조건들도 브라우저 레벨에서 판단함.
- 서버 : 서버 컨테이너나, 사용하는 프레임워크가 기본적인 것들을 지원해 준다던지 하는 경우에 따라 다를 수 있지만, 클라이언트가 Preflight Request를 날릴 시, 규약에 맞는 헤더를 응답으로 줘야 함 (<https://www.w3.org/TR/cors/#syntax>)
응답에 내려오는 대표적인 헤더들은 다음과 같음
 - Access-Control-Allow-Origin
 - Access-Control-Allow-Credentials
 - Access-Control-Allow-Methods
 - Access-Control-Allow-Headers

Access-Control-Allow-Origin

허용된 Origin(도메인) 을 명시해주는 헤더

기본적으로 'All Or Nothing' 임을 상기하자. '명확하게 입력된 URL 타입의 scheme, domain, port' 를 완벽하게 적어야 하거나, null , * 만 허용

The **Access-Control-Allow-Origin** header indicates whether a resource can be shared based by returning the value of the **Origin** request header, ******, or **"null"** in the response. ABNF:

```
Access-Control-Allow-Origin = "Access-Control-Allow-Origin" ":" origin-list-or-null | "**"
```

*Note: In practice the **origin-list-or-null** production is more constrained. Rather than allowing a space-separated list of **origins**, it is either a single **origin** or the string **"null"**.*

Access-Control-Allow-Credentials

Actual Request에 대한 인증여부를 허용할지 판단하는 헤더. value는 boolean 타입으로 true | false 이다.

- false로 응답값을 주게 되면, 클라이언트(정확히는 브라우저단에서는)는 인증이 포함된 Actual Request를 보낼 수 없음.

- true로 응답을 주면, 클라이언트는 인증이 포함된 Actual Request를 보낼 수 있고, 인증이 포함되지 않은 Actual Request 또한 전송가능.

Access-Control-Allow-Methods

허용하는 HTTP 메소드들을 기술하여 응답을 주는 헤더

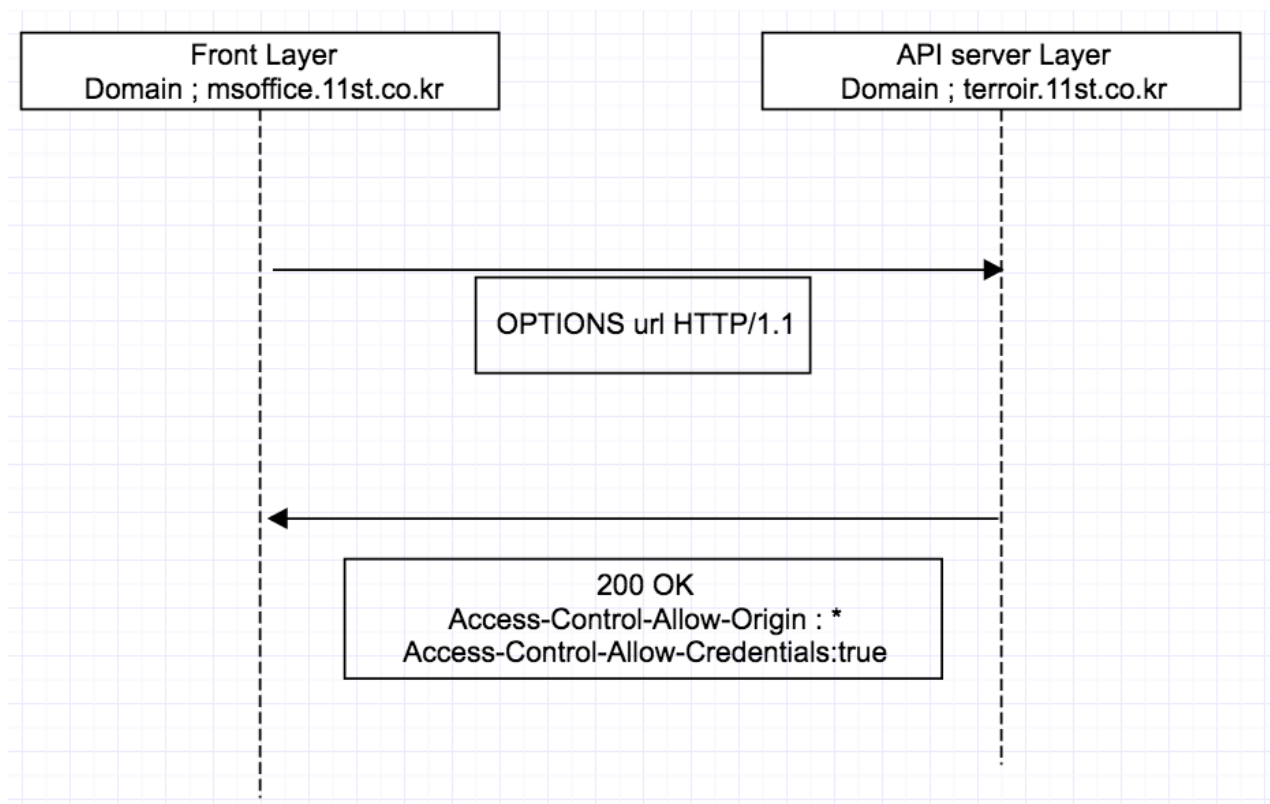
Access-Control-Allow-Headers

허용하는 헤더들(커스텀 헤더도 포함)을 기술하여 응답을 주는 헤더

이슈 1. Access-Control-Allow-Origin & Access-Control-Allow-Credentials

상기 2가지 헤더가 합쳐진 경우, 특이사항이 존재한다.

MSO 로컬 개발 도중에 경험하였었는데 다음과 같은 상황 발생함



test-req

Request URL: http://terroir.11st.co.kr/product/test-req
Request Method: OPTIONS
Status Code: 200
Remote Address: 127.0.0.1:80
Referrer Policy: no-referrer-when-downgrade

▼ Response Headers view source
Access-Control-Allow-Credentials: true
Access-Control-Allow-Headers: Content-Type, Accept, x-requested-with
Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Origin: *
Allow: GET, HEAD, POST, PUT, DELETE, OPTIONS, PATCH
Connection: keep-alive
Content-Length: 0
Date: Tue, 06 Feb 2018 07:43:02 GMT
Server: nginx/1.12.2
X-Application-Context: application:local:9090

▼ Request Headers view source
Accept: */*
Accept-Encoding: gzip, deflate
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
Access-Control-Request-Headers: content-type
Access-Control-Request-Method: POST
Connection: keep-alive
Host: terroir.11st.co.kr
Origin: http://msoffice.11st.co.kr
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36

1 requests | 431 B transferred

⋮ Console

🔍 Filter Default levels ▾ Group similar

✖ Failed to load http://terroir.11st.co.kr/product/test-req: Response to preflight request doesn't pass access control check: The value of the 'Access-Control-Allow-Origin' header in the response must not be the wildcard '*' when the request's credentials mode is 'include'. Origin 'http://msoffice.11st.co.kr' is therefore not allowed access. The credentials mode of requests initiated by the XMLHttpRequest is controlled by the withCredentials attribute.

✖ Failed to load http://terroir.11st.co.kr/product/test-req: Response to preflight request doesn't pass access control check: The value of the 'Access-Control-Allow-Origin' header in the response must not be the wildcard '*' when the request's credentials mode is 'include'. Origin 'http://msoffice.11st.co.kr' is therefore not allowed access. The credentials mode of requests initiated by the XMLHttpRequest is controlled by the withCredentials attribute.

영어 요약 : credentials mode(클라이언트측에서 XMLHttpRequest.credentials플래그를 세팅함. 이 값을 true로 세팅한 경우에만 Actual Request에 쿠키를 포함하여 전송)를 '포함' 하고 있을 때엔, Access-Control-Allow-Origin 은 * 이 아니어야 한다.

브라우저 자체에서 차단하게 되므로(크롬 64버전 기준) 클라이언트 코드 내에서 이를 해결할 수 있는 근본적인 방법은 존재하지 않음.

Cross-Origin Resource Sharing x

JaeHo

< > ↺ 🏠

안전함 | <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>

☆ ⚙️ 🔍 📄 📱

[Jump to:](#) [Who should read this article?](#) [What requests use CORS?](#) [Functional overview](#)

[Examples of access control scenarios](#) [The HTTP response headers](#) [The HTTP request headers](#) [Specifications](#)

[Browser compatibility](#) [See also](#)

31 | [text/plain payload]

Although line 11 contains the Cookie destined for the content on `http://bar.other`, if `bar.other` did not respond with an `Access-Control-Allow-Credentials: true` (line 19) the response would be ignored and not made available to web content.

Credentialed requests and wildcards

When responding to a credentialed request, the server **must** specify an origin in the value of the `Access-Control-Allow-Origin` header, instead of specifying the `"*"` wildcard.

Because the request headers in the above example include a `Cookie` header, the request would fail if the value of the `Access-Control-Allow-Origin` header were `"*"`. But it does not fail: Because the value of the `Access-Control-Allow-Origin` header is `"http://foo.example"` (an actual origin) rather than the `"*"` wildcard, the credential-cognizant content is returned to the invoking web content.

Note that the `Set-Cookie` response header in the example above also sets a further cookie. In case of failure, an exception—depending on the API used—is raised.

Third-party cookies

Note that cookies set in CORS responses are subject to normal third-party cookie policies. In the example above, the page is loaded from `foo.example`, but the cookie on line 22 is sent by `bar.other`, and would thus not be saved if the user has configured their browser to reject all third-party cookies.

https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS#Who_should_read_this_article



해결 방법 : Access-Control-Allow-Origin 헤더 값을 *가 아닌, '명확히' 기술된 Origin을 명시하여야 한다.

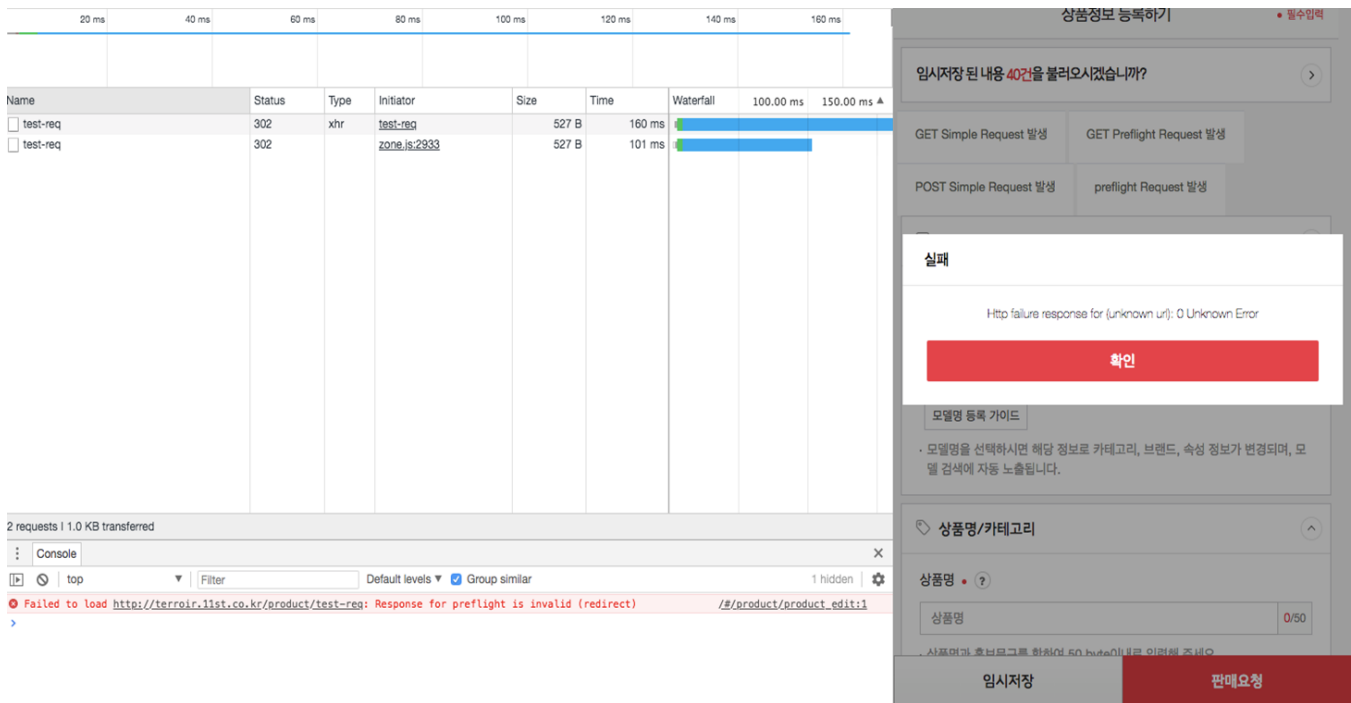
(https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin_policy 의 Definition of an Origin 참고)

해결 후 Preflight & Actual Request

Name	×	Headers	Preview	Response	Timing
☒ test-req ☐ test-req		Request URL: http://terroir.11st.co.kr/product/test-req Request Method: OPTIONS Status Code: 200 Remote Address: 127.0.0.1:80 Referrer Policy: no-referrer-when-downgrade		Response Headers view source Access-Control-Allow-Credentials: true Access-Control-Allow-Headers: Content-Type, Accept, x-requested-with Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS Access-Control-Allow-Origin: http://msoffice.11st.co.kr Allow: GET, HEAD, POST, PUT, DELETE, OPTIONS, PATCH Connection: keep-alive Content-Length: 0 Date: Tue, 06 Feb 2018 09:02:38 GMT Server: nginx/1.12.2 X-Application-Context: application:local:9090	
2 requests 1007 B transferred		Request Headers view source Accept: */* Accept-Encoding: gzip, deflate Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7 Access-Control-Request-Headers: content-type Access-Control-Request-Method: POST Connection: keep-alive Host: terroir.11st.co.kr Origin: http://msoffice.11st.co.kr User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36			

Name	x	Headers	Preview	Response	Cookies	Timing
☐ test-req ☒ test-req		▼ General Request URL: http://terroir.11st.co.kr/product/test-req Request Method: POST Status Code: 200 Remote Address: 127.0.0.1:80 Referrer Policy: no-referrer-when-downgrade ▼ Response Headers view source Access-Control-Allow-Credentials: true Access-Control-Allow-Headers: Content-Type, Accept, x-requested-with Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS Access-Control-Allow-Origin: http://msoffice.11st.co.kr Connection: keep-alive Content-Type: application/json; charset=UTF-8 Date: Tue, 06 Feb 2018 09:02:38 GMT Server: nginx/1.12.2 Transfer-Encoding: chunked X-Application-Context: application:local:9090 ▼ Request Headers view source Accept: application/json, text/plain, */* Accept-Encoding: gzip, deflate Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7 Connection: keep-alive Content-Length: 31 Content-Type: application/json Cooki...PCID=15175383835495138597938; _ga=GA1.3.460319070.1517538387; plab.ver=0.2.11; plab.uid=2185101f-f929-4f2b-8d0e-0a6ef3ceba77; plab.h.11st_web=; TT=CONN_IP_LOC%7CD0M; cto_lwid=a2e0826a-9715-4a6e-a570-dd86083ba405; _gid=GA1.3.615788947.1517796358; WMONID=Q99As__uid; TD=TB_ACT_DATA%7C0%3AN%3A-1%3A0%3AN%3A-1; RAKE_SID=15179076209745939425658; RAKE_SID_XSITE=15179076209745939425658; TMALL_MNC=MTAyNDAYODI2MjQ; TMALL_STATIC=Y; TMALL_AUTH=BuZPWiphdgapRZ8yrSLEeaa0loqnGxhLjDY1cuklJZyX08Tt6JJJa5TnSIJl3LcPaNGlIx4NU6NJ2CkyYrUHBNOXkPjrUK5Kziux6PvbMhIMRH4jC8TXPGKa9hUqpMb10VJUuacPLQHGaalRznagbZG6J6ayfaY0kesnKjUaz5g0TaIKqLzda8%2B0BRyVKj8vICv%2B48CWChcC3KUa4LT0ybTwMmp3%2Ba90oKQJARSWlxUHFj9ds5yTde6lyxkG%2B8vFpaALJLWRN0ZhWVFJShpLCnyd1%2B21LCjA%2B03kubQAm12F00juQJCzrDKBqDuRPKks10JtGkGI80jasrDdrtdw3PPurZU6wfe%2Bb2NoQBUBF8VMWdcxyHGW150rIpy4NDcdor2WT0PLip%2FxfahSG4wYQP5%2Fpg4db%2ByBJGGyWAh95ieJG9NKS8M10DeXxeipV%2FKn%2FiD6Pqujk%2B%2FIZUSYbE%2FHo0oJuarVdNbFvqSQI224bb0%2B9Vlf1qU%2Fiocy8oFP08GJrB8qH9qA9FVARMia4jYVyuZ9d2Pq1Ycw%2BjQdfntdvHXJ57vKuxaH9makr37YNxKmfhtEy4tJKJ2XD%2FGeVDJ%2B%2F5B113V47fYwUkUmpF3h0rR%2B364E0dXVMWp3dBMZ5sifVzEfF8lTbY9k5NfbpbeXDYf9p0sWfUyF32tJos05KZRpSZNZDbg%2Fli1i0FepqFJbc2VuXCxM56%2FvAEuCOsdhcDPzsADso12Dqi8yY4BDnC4NqIBM1PsZSJX3pzxM31pPCR3%2Bp8UNOWw0iSyvv2BAUjTN0Lfdsgcuq9q2j6U8k945pgFhnm14LeaZ7kvQUPk0PtITIsZx2U%2BQBE8mgyLKPWKS%2Fm9IjP2GGk6KbbQrVm6gQSLUM0Fqqkfukun801fELa426HgjSwKpfSdzufG5I4j1mRqpzhdBvmfU4teD%2F9HJr6UI0NwgrT9asn4P9wiHZXqYE6glZG8XZFg%3D%3D; TMALL_KEY_VALUE=aAh79WtUZ4Hir9LcNvASKQUGRnf5mdHvt1kRkabu0oY%3D; TP=TARGETING_INV%7CNNN%23scrnChk%7CY%23TB_DATA_CHK%7CN%3AY%23ENCRYPYPT_VER%7C1%23M_N%7C10000276%23ACUSR%7CY%23TM_SGMT%7C48bd054%23GA_BGC%7C1%23GA_BED%7C122%23GA_R0D%7CP0M1%23GA_AGL%7CZZZ%23GTYN%7CY%23uid%7C5000a128168e6a0ed78c1dd5f35fe849; appVCA= Host: terroir.11st.co.kr Origin: http://msoffice.11st.co.kr Referer: http://msoffice.11st.co.kr/ User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 ▼ Request Payload view source {key: "hello", value: "world"} key: "hello" value: "world"				
2 requests 1007 B transferred						

이슈 2. 서버의 Preflight Request 구분(Cookie)



Preflight Request를 날렸는데, 오류 메시지로 'Response for preflight is invalid' 라는 브라우저 오류 메시지가 노출.

<https://www.w3.org/TR/cors/#cross-origin-request-with-preflight-0>

해당 문서를 보면, Preflight Request의 응답으로 200번대 code가 아닌 것은 모두 오류로 간주하므로, 브라우저는 오류 메시지를 노출시킨다.

또한, Preflight Request는 다음과 같은 제약조건을 표준으로 가진다.

Otherwise, make a **preflight request**. **Fetch** the **request URL** from **origin source origin** using **referrer source** as **override referrer source** with the **manual redirect flag** and the **block cookies flag** set, using the method **OPTIONS**, and with the following additional constraints:

- Include an **Access-Control-Request-Method** header with as header field value the **request method** (even when that is a **simple method**).
- If **author request headers** is not empty include an **Access-Control-Request-Headers** header with as header field value a comma-separated list of the header field names from **author request headers** in lexicographical order, each **converted to ASCII lowercase** (even when one or more are a **simple header**).
- Exclude the **author request headers**.
- Exclude **user credentials**.
- Exclude the **request entity body**.

기본적으로 Actual Request가 credentials : true 로 세팅되어있다고 하여도, Preflight Request는 인증정보를 제외하고 전송을 한다.

해결방법 : 서버측에서(Mobile SO에서는 인터셉터 레벨에 적용함) 클라이언트의 request가 Actual Request인지, 아니면 Preflight Request인지 구분을 하여

Preflight Request인 경우에는 별도의 인증여부를 판단하지 않고 Response 를 내려 주는 구분로직의 구현이 필수이다.

Preflight Request의 구분은 그저 Request method가 ' OPTIONS' 인지 아닌지만 파악하면 된다.

해결 후 Preflight & Actual Request

Name	×	Headers	Preview	Response	Timing
☒ test-req					
☐ test-req					
2 requests 1007 B transferred					
		Request URL: http://terroir.11st.co.kr/product/test-req Request Method: OPTIONS Status Code: 200 Remote Address: 127.0.0.1:80 Referrer Policy: no-referrer-when-downgrade ▼ Response Headers view source Access-Control-Allow-Credentials: true Access-Control-Allow-Headers: Content-Type, Accept, x-requested-with Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS Access-Control-Allow-Origin: http://msoffice.11st.co.kr Allow: GET, HEAD, POST, PUT, DELETE, OPTIONS, PATCH Connection: keep-alive Content-Length: 0 Date: Tue, 06 Feb 2018 09:02:38 GMT Server: nginx/1.12.2 X-Application-Context: application:local:9090 ▼ Request Headers view source Accept: */* Accept-Encoding: gzip, deflate Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7 Access-Control-Request-Headers: content-type Access-Control-Request-Method: POST Connection: keep-alive Host: terroir.11st.co.kr Origin: http://msoffice.11st.co.kr User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36			

Name	×	Headers	Preview	Response	Cookies	Timing
☐ test-req						
☒ test-req						
2 requests 1007 B transferred						
▼ General Request URL: http://terroir.11st.co.kr/product/test-req Request Method: POST Status Code: 200 Remote Address: 127.0.0.1:80 Referrer Policy: no-referrer-when-downgrade ▼ Response Headers view source Access-Control-Allow-Credentials: true Access-Control-Allow-Headers: Content-Type, Accept, x-requested-with Access-Control-Allow-Methods: POST, GET, PUT, PATCH, DELETE, OPTIONS Access-Control-Allow-Origin: http://msoffice.11st.co.kr Connection: keep-alive Content-Type: application/json; charset=UTF-8 Date: Tue, 06 Feb 2018 09:02:38 GMT Server: nginx/1.12.2 Transfer-Encoding: chunked X-Application-Context: application:local:9090 ▼ Request Headers view source Accept: application/json, text/plain, */* Accept-Encoding: gzip, deflate Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7 Connection: keep-alive Content-Length: 31 Content-Type: application/json Cookie: PCID=15175383835495138597938; _ga=GA1.3.460319070.1517538387; plab.ver=0.2.11; plab.uid=2185101f-f929-4f2b-8d0e-0a6ef3ceba77; plab.h.11st_web=; TT=CONN_IP_LOC%7CD0M; cto_lwid=a2e0826a-9715-4a6e-a570-dd86083ba405; _gid=GA1.3.615788947.1517796358; WMONID=Q99As__uid; TD=TB_ACT_DATA%7C0%3AN%3A-1%3A0%3AN%3A-1; RAKE_SID=15179076209745939425658; RAKE_SID_XSITE=15179076209745939425658; TMALL_MNC=MTAyNDAYODI2MjQ; TMALL_STATIC=Y; TMALL_AUTH=BuZPWiphdgapRZ8yrSLEeaa0loqnGxhLjDY1cukLjZYx08Tt6JJJa5TnSIJl3LcPaNGLIx4NU6NJ2CkyYrUHBNOXkPjrUK5Kziux6PvbMhIMrH4jC8TXPGKa9hUqpMbL0VJUuacPLQHGaaLRznagbZG6J6ayfaY0kesnKjUaz5g0TaIKqLzda8%2B0BRyVKj8VlCv%2B48CWChcC3KUa4LT0ybTwwMp3%2Ba90oKQJARSWlxUHfJ9ds5YtDe6lyxkG%2B8vFpaALJLWRN0ZhWfJShpLCnyd1%2B21LCJA%2B03kubQAm12F00juQJCzrDKBqDuRPKks10JtGkGI80jasrDdrtdw3PPurzu6Wf%2Bb2NoQBUBF8VMwdcxyHGW150rIpy4NDcdor2WT0PLip%2FxfahSG4wYQP5%2Fpg4db%2ByBJGGyWAh95ieJG9NKS8M100eXxeip%2FkN%2FiD6PQujk%2B%2FIZUSYbE%2FHo0oJuarVdNbFvqSQI224bb0%2B9VlfiqU%2Fiocy8oF0P08Gjrb8qHn9qA9FVARMia4YjVYuZn9d2Pq1YcW%2BjQdfntdvHJ57vKuxaH9makr37YnXkmoftHEy4tJKJ2XD%2FGeVDJ%2B%2F5B113V47fYwUkmYpF3h0rR%2B364E0dXVMWp3HbMZ5sifVzEfF8lTy9k5NfbpbeXDYf9p0swfUyF32tJos05KZRpSNZDbg%2Fli1i0FepqFJbc2VuXCxM56%2FvAEuC0sdhcDPzsAdso12Dqi8yY4BDnC4NqIBM1PsZSJX3pzxM31pPCR3%2Bp8UNOWw0iSyvv2BAUjTN0Lfdsgcuq9q2j6U8k945pgFhnm14LeaZ7kvQUPk0PtITIsZx2U%2BQBE8mgYlKPWKS%2Fm9IjP2GGk6KbbQrVm6gQSLUM0Fqqkfukun801fELa426HgjSwKpfSdzufGS14j1mRqpzhdBvmfU4teD%2F9HJr6UI0NwgrT9asn4P9wiHZXqYE6glZGBXZFg%3D%3D; TMALL_KEY_VALUE=aAh79WtUZ4Hir9LcNvASKQUGRNf5mdHvt1kRkabu0oY%3D; TP=TARGETING_INV%7CNNN%23scrnChk%7CY%23TB_DATA_CHK%7CN%3AY%23ENCRYPYPT_VER%7C1%23M_N%7C10000276%23ACUSR%7CY%23TM_SGMT%7C48bd054%23GA_BGC%7C1%23GA_BED%7C122%23GA_R0D%7CP0M1%23GA_AGL%7CZZ%23TGTYN%7CY%23uid%7C5000a128160e6a0ed78c1dd5f35fe849; appVCA= Host: terroir.11st.co.kr Origin: http://msoffice.11st.co.kr Referer: http://msoffice.11st.co.kr/ User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 ▼ Request Payload view source ▼ {key: "hello", value: "world"} key: "hello" value: "world"						

