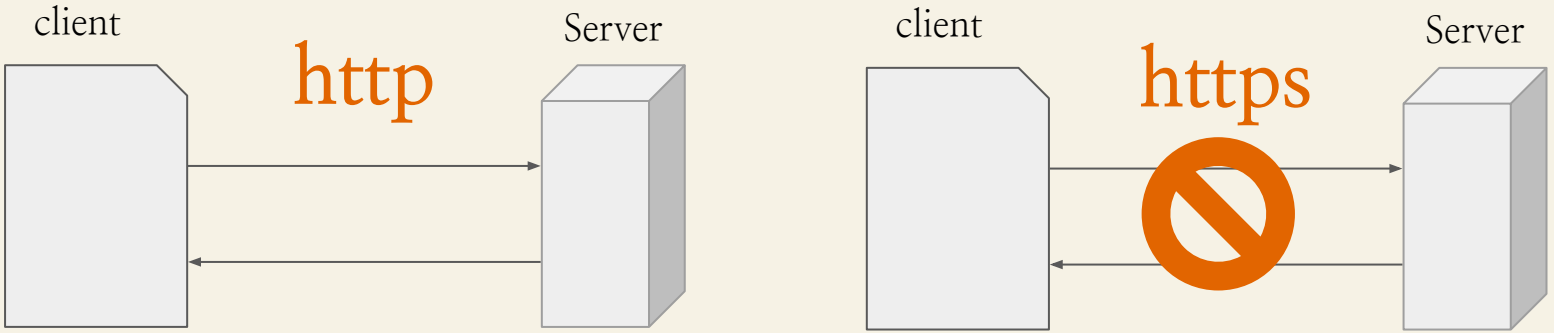
HTTP(HyperText Transfer Protocol)

- HTTP는 웹브라우저와 웹 서버가 HTML로 작성된 웹 페이지나 동영상 음성파일등을 주고받기 위한 통신 규약(프로토콜, 약속, 규칙)이다.
- 이 프로토콜을 SSL(Secure Socket Layer)로 암호화 하여 보안성을 확보한 것을 가리켜 HTTPS라고 한다.



HTTP 요청 메시지

요청행

GET https://www.kopo.ac.kr HTTPS/1.1
요청 메소드 + URL + HTTP의 버전

요청 메소드(송수신 방법) : GET, POST, PUT, DELETE 등

- GET : https://www.kopo.ac.kr/userInfo?id='kopo01'&dept=da
  웹 서버에 페이지를 요청한다. 요청할 때 필요한 데이터는 URL에 덧붙여 보내며 텍스트 데이터만 전송할 수 있다.

- POST : 서버의 데이터를 갱신하거나 보내는 데이터의 양이 많을 때, 비밀번호 등의 개인정보를 보낼 때 사용한다. 요청할 때 필요한 데이터는 메시지 본문에 담아서 보내며 텍스트 데이터와 바이너리 데이터를 모두 보낼 수 있다.

요청 헤더

메시지의 제어정보, 메시지 본문에 저장한 데이터 정보(데이터 종류나 문자 코드)등이 텍스트 형식으로 저장
헤더의 문자열은 Host : https://www.kopo.ac.kr 처럼 필드 이름 : 필드 내용을 쌍으로 구성한 필드의 집합으로 구성되어 있다.

공백 행

메시지 본문

메시지 본문에는 보내는 데이터가 저장된다. 데이터 형식은 요청 헤더에 지정된 타입을 따른다.

HTTP 응답 메시지

응답행	HTTP/1.1 200 OK - HTTPS 버전 : HTTP/1.1 - 상태코드 : 200 - 보충메시지 : OK
응답 헤더	메시지의 제어정보, 메시지 본문에 저장한 데이터 정보 (데이터 종류나 문자 코드)등이 텍스트 형식으로 저장
공백 행	
메시지 본문	메시지 본문에는 받을 데이터가 담기며, 그 데이터 형식은 요청 헤더에 지정된 타입을 따릅니다. GET 메소드 요청에 대한 응답 메시지는 HTML 파일, CSS 파일, 자바스크립트 파일, 이미지 파일 처럼 웹 페이지를 정의하는 데이터입니다.

HTTP 상태 코드

HTTP의 주요 상태코드와 상태 설명

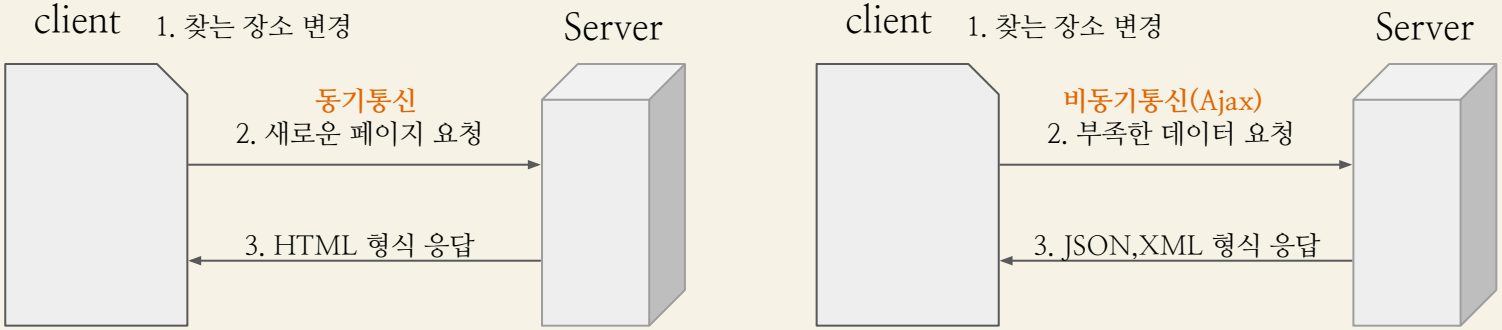
분류	상태 코드	상태 설명	내용
성공	200	OK	요청을 성공함
클라이언 오류	401	unauthorized	인증되지 않음
	403	Forbidden	엑세스가 허용되지 않음
	404	Not Found	요청한 리소스를 찾지 못함
	408	Request Timeout	요청 시간을 초과함
서버 오류	500	Internal Server Error	서버 내부에서 오류가 발생함
서버 오류	503	Service Unavailable	서비스를 일시적으로 사용할 수 없음

Ajax(Asynchronous Javascript and XML)

- Ajax란 XMLHttpRequest라는 자바스크립트의 객체를 이용하여 웹 서버와 비동기로 통신하고 DOM을 이용하여 웹페이지를 동적으로 갱신하는 프로그래밍 기법을 말한다.
- 2005년에 등장
- XML(eXtensible Markup Language) : 데이터를 보내는 형식
- 현재는 XML을 사용하는 경우는 매우 드물고, 주로 JSON과 텍스트 데이터를 사용한다.

Ajax의 특징

- 구글지도(Google Maps), 네이버 지도 등의 서비스가 있다.
- 기존의 지도 서비스
  - 사용자가 찾아보려는 장소 변경 -> 클라이언트는 서버에 새로운 지도 정보 요청 -> 서버에서 새로운 지도를 포함한 웹페이지 전체 생성 -> 클라이언트로 전송
- Ajax를 활용한 지도 서비스
  - 사용자가 찾아보려는 장소 변경 -> 클라이언트측 자바스크립트가 현재 가지고 있는 지도 데이터에서 부족한 부분 파악 후 그 부분만 요청 -> 서버에서 클라이언트가 필요로 하는 데이터만 전송 -> 클라이언트는 이 데이터를 받아서 필요한 부분만 DOM으로 변경



Ajax의 장점

- 최소한의 데이터 통신만 하므로 처리 속도가 빠르고 서버 부하와 통신 트래픽 부하가 적다.
- 비동기로 통신하므로 클라이언트 측에서 다른 작업을 할 수 있다.
- 웹 페이지 갱신을 클라이언트 측이 담당한다. 페이지를 전환하는 대신에 페이지 일부분만 변경하므로 고속 렌더링이 가능하다.
- 또한 Ajax를 활용하면 화면 전환 빈도가 줄어들기 때문에 사용자에게 데스크톱 애플리케이션을 사용하는 듯한 편의성을 제공할 수 있다.

XMLHttpRequest

- Ajax 기법을 사용할 때는 데이터의 송수신에 XMLHttpRequest 객체를 사용한다.
- XMLHttpRequest 처리 프로세스
  - XMLHttpRequest 객체를 생성한다.
  - 서버와 통신할 때 사용할 처리 방법을 등록한다.
  - 요청을 전송하고 통신을 시작한다.

XMLHttpRequest 객체의 프로퍼티

프로퍼티	설명	읽기 전용 여부
readyStateHTTP	통신 상태를 가져온다(0~4 사이의 값).	읽기 전용
response	응답 내용을 가져온다.	읽기 전용
responseText	응답 내용을 텍스트 형식으로 가져온다.	읽기 전용
responseType	응답 타입을 가져오거나 설정한다.	쓰기 가능
responseXML	응답 내용을 XMLDocument 객체로 가져온다.	읽기 전용
status	요청에 대한 HTTP 상태 코드를 가져온다.	읽기 전용
statusText	요청에 대한 보충 메시지를 가져온다.(HTTP 상태 코드 내용)	읽기 전용
timeout	요청을 자동으로 끝내는 데 걸린 시간(ms)을 가져오거나 설정한다.	쓰기 가능
withCredentials	크로스 오리진 통신에 대해 인증 정보를 사용할지를 뜻하는 논리값	쓰기 가능
onreadystatechange	readyState 값이 바뀔 때마다 호출되는 이벤트 처리기	쓰기 가능
ontimeout	요청 시간이 초과할 때마다 호출되는 이벤트 처리기	쓰기 가능

XMLHttpRequest 객체의 메소드

메소드	설명
`abort()`	현재 실행 중인 비동기 통신을 중단한다.
`getAllResponseHeaders()*`	수신한 모든 HTTP 응답 헤더를 가져온다.
`getResponseHeader(header)*`	특정 HTTP 응답 헤더를 가져온다.
`open(...)`	HTTP 요청을 초기화 한다.
`send(data)`	HTTP 요청을 보낸다.
`setRequestHeader(header,value)`	요청 헤더에 정보를 추가한다.

* send 메소드가 성공했을 때만 사용할 수 있다.

readyState의 프로퍼티 값

값	설명
0	초기화 되지 않음 : open 메소드가 호출된 상태가 아니다.
1	로드 중 : open 메소드는 호출되었지만 send 메소드가 호출되지 않았다.
2	로드 완료 : send 메소드는 호출되었지만 응답이 되돌아오지 않았다.
3	응답 수신 중 : 응답 행과 응답 헤더는 가져왔지만 메시지 본문을 가져오지 못했다.
4	수신완료 : 모든 응답 메시지를 수신한다.

XMLHttpRequest 객체의 이벤트

- XMLHttpRequest 객체는 readystatechange 이벤트 외에도 아래와 같은 이벤트를 사용할 수 있다.

이벤트	이벤트가 발생하는 시기
abort	요청이 취소되었을 때
error	요청이 실패했을 때
loaded	요청이 완료되었을 때(성공과 실패에 관계없이 발생함)
load	요청을 성공하여 응답을 가져올 수 있을 때
loadstart	start 요청을 보낼 때
progress	데이터를 주고 받는 중
timeout	요청 시간을 초과했을 때

POST 메소드로 보낼 수 있는 주요 데이터

데이터 타입	설명
`DOMString`	문자열 데이터
`FormData`	폼 데이터
`ArrayBufferView`	바이너리 데이터
`Blob`	Blob 객체, File 객체
`Document`	HTML의 Document 객체

response 프로퍼티의 값

- send 메소드로 데이터를 보내기 전에 responseType 프로퍼티에 값을 받고자 하는 데이터 타입을 설정해 준다.
- 예를 들어 응답 데이터로 JSON 객체를 받고자 할때는 아래와 같이 설정한다.
  - `req.responseType = “json”;`

response의 데이터 타입	설명	responseType의 값
DOMString	텍스트	“text”(기본값)
JSON 객체	JSON 문자열을 파싱한 JSON 객체	“json”
ArrayBuffer	형식화 배열(TypedArray)	“arraybuffer”
Blob	Blob 객체	“blob”
Document	HTML의 Document 객체	“document”

CORS(Cross Origin Resource Sharing)

- '데이터를 가져오는 대상이 악의적인 웹사이트라면 읽기와 쓰기를 금지해야 하지만 그 상대를 신뢰할 수 있다면 데이터의 읽기와 쓰기를 허용하자'는 것
- XMLHttpRequest level1 에서는 크로스 오리진 통신이 금지되어 있지만 XMLHttpRequest level2 부터는 크로스 오리진 통신을 제한적으로 허용한다.
- 이 사양을 가리켜 CORS라고 한다.

postMessage

- 부모 윈도우가 open 메소드로 자식창을 열면 부모 창과 자식창이 서로의 Window 객체를 참조하여 조작할 수 있다.
- 이는 iframe 요소로 읽어들인 웹페이지에도 똑같이 적용된다.
- 그러나 이렇게 오픈된 윈도우라고 하더라도 그 안의 페이지 출처가 다르면 동일 출처 정책에 따라 상대방 창의 콘텐츠를 가져오는 것조차 불가능해진다.
- 프로그램에서 다른 윈도우에 있는 객체의 프로퍼티를 읽거나 메소드를 실행하는 것도 불가능하다.
- 이때 HTML5로부터 추가된 Window 객체의 postMessage 메소드를 활용하면 출처가 다른 창끼리 통신할 수 있다.
- 여전히 상호간의 참조와 메소드 실행은 불가능하지만 메시지를 비동기적으로 주고 받을 수 있다.

postMessage

```
// postMessage 메소드로 다른 윈도우에 메시지를 보내는 코드 형식
target.postMessage(message,origin);

// 포트를 붙일수도 있고, 안붙일수도 있다.
target.postMessage("hello","https://www.daum.net:8080");
```

postMessage

```
target.postMessage(message,origin);
```

- target
  - 메시지를 받을 윈도우의 Window 객체, open 메소드로 연 윈도우에 메시지를 보내려면 open 메소드가 반환한 값을 넘긴다. iframe 요소의 윈도우 안이라면 iframe 요소 객체의 contentView 프로퍼티의 값을 넘긴다.
- message
  - 보내는 메시지, HTML5 사양에는 모든 데이터 타입을 지정할 수 있다고 명시되어 있지만 문자열만 보낼 수 있는 웹 브라우저도 있다.
- origin
  - 메시지를 받을 윈도우 출처를 뜻하는 문자열. 이 문자열에는 URL 스킴(프로토콜)과 호스트 이름을 지정한다. 필요하면 포트 번호도 표기한다. 메시지를 보내는 윈도우 출처를 명시하지 않을 때는 *를 지정한다. 메시지를 보내는 윈도우와 메시지를 받는 윈도우 출처가 같다면 '/'를 지정할 수 있다.

postMessage

- message 이벤트 프로퍼티
  - data : postMessage 메소드의 첫 번째 인수로 전달한 메시지의 복사본
  - source : 메시지를 보낸 Window 객체
  - origin : 메시지를 보낸 웹 사이트의 출처를 뜻하는 문자열
- 메시지를 받는 윈도우에서 message 이벤트가 발생한다. 메시지를 받는 윈도우의 웹 페이지 안에 message 이벤트 리스너를 만들어 두면 이벤트 리스너 안에서 메시지를 받을 수 있다.
  - `window.addEventListener("message", function(e){...}, false);`