

AJAX(Asynchronous Javascript And XML)

1. AJAX

AJAX는 웹페이지가 전체를 다시 불러오지 않고도, 서버에서 필요한 정보만 살짝 가져오게 해주는 기술이다. 예를 들어, 웹페이지에서 날씨 정보를 보여주는 작은 부분을 예로 들 수 있다. 날씨 정보 부분만 서버에서 새로운 데이터를 받아와 업데이트하면, 나머지 웹페이지는 그대로 유지하면서 정보만 새로 고친다.

AJAX를 사용함으로써 페이지를 새로 고치지 않아도 된다. 자연스럽게 빠르게 정보를 업데이트 할 수 있어 사용자가 더 편하게 사이트를 사용할 수 있기 때문이다.

또한 필요한 부분만 정보를 주고받기 때문에 서버의 부담이 줄어들지만 반대로 대용량의 데이터를 전송할 경우 서버의 부담이 증가된다.

2. AJAX Workflow

2-1) 요약

사용자의 행동

- 웹페이지에서 버튼을 클릭하거나 글을 쓰는 등의 행동을 한다.

요청 보내기

- 이 행동에 반응하여 웹페이지가 서버에 정보를 요청한다. 이 요청은 웹페이지의 나머지 부분을 방해하지 않고 조용히 이루어진다.

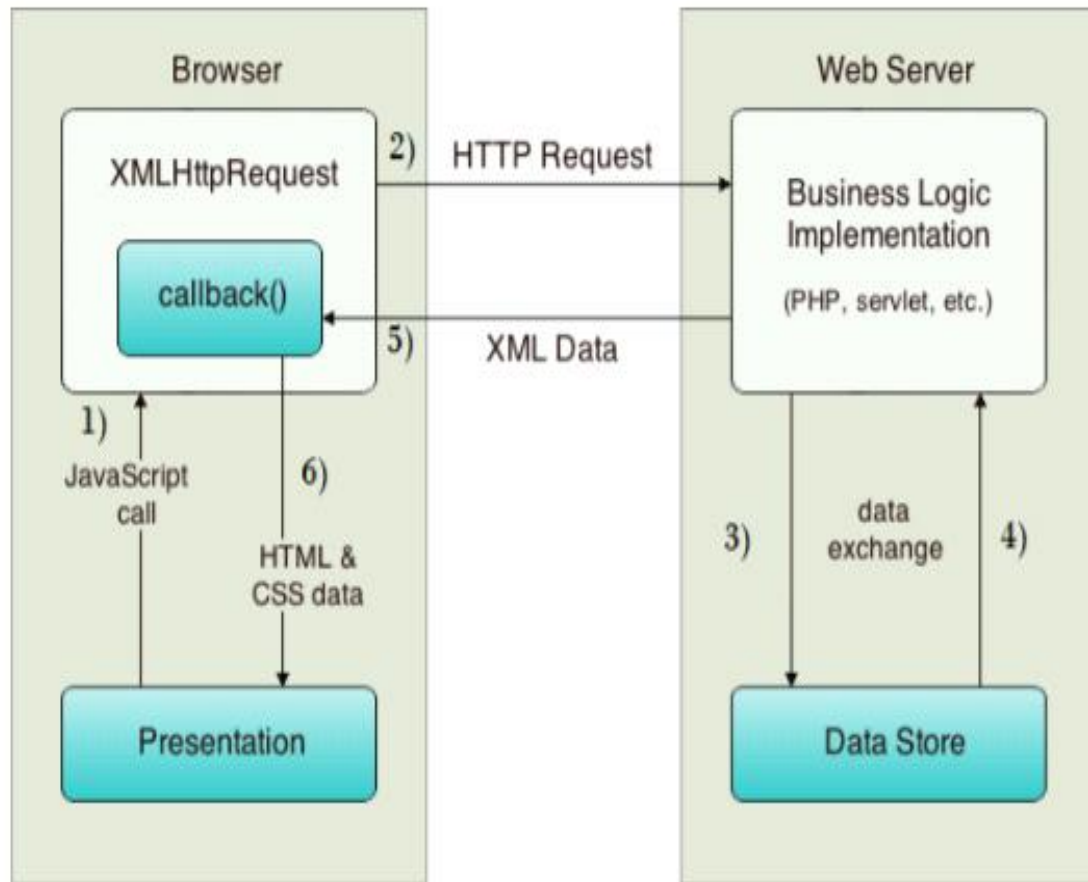
서버의 반응

- 서버는 요청받은 정보를 처리하여 다시 웹페이지로 보낸다.

화면 업데이트

- 웹페이지는 받은 정보로 화면 일부분을 새로 고치거나 업데이트한다.

2-2) 상세



이미지 출처:

<https://www.devopsschool.com/blog/what-is-ajax-and-how-it-works-an-overview-and-its-use-cases/>

- 1) 사용자가 UI에서 요청을 보내고 javascript 호출이 XMLHttpRequest 객체로 이동합니다.
- 2) HTTP 요청은 XMLHttpRequest 객체를 통해 서버로 전송됩니다.
- 3) 서버는 JSP, PHP, Servlet, ASP.net 등을 사용하여 데이터베이스와 상호 작용합니다.
- 4) 데이터가 검색됩니다.
- 5) 서버는 XML 데이터 또는 JSON 데이터를 XMLHttpRequest 콜백 함수로 보냅니다.
- 6) HTML 및 CSS 데이터가 브라우저에 표시됩니다.

3. AJAX구성요소

```
$.ajax({  
  
    url : "",  
    type : "",  
    async : "",  
    cache : "",  
    contentType : "",  
    data : "",  
    dataType : "",  
    success : function(data , status , xhr){  
    },  
    error : function(xhr , status , errorThrown){  
    },  
    complete : function(xhr , status){  
    }  
  
});
```

1) url

- 서버의 URL을 지정한다. 이 URL로 AJAX 요청이 보내진다.

예: url: "http://example.com/api/data"

2) type

HTTP 요청 방식을 지정한다. 일반적으로 GET 또는 POST가 사용된다.

예: type: "POST"

3) async

- 요청이 비동기적으로 수행될지 동기적으로 수행될지를 지정한다. true는 비동기, false는 동기 요청이며 기본값은 true이다.

예: async: true

4) cache

- 브라우저가 서버로부터 받은 응답을 캐시(임시 저장)할지 결정한다. 기본적으로 GET 요청의 결과는 캐시되지만 이를 false로 설정하면 URL에 타임스탬프를 추가하여 캐시를 방지할 수 있다.

예: cache: false

5) contentType

서버로 전송할 데이터의 타입을 MIME 타입으로 지정한다. 예를 들어 "application/x-www-form-urlencoded; charset=UTF-8"는 웹 폼 데이터를 의미한다.

예: contentType: "application/json; charset=utf-8"

6) data

- 서버로 전송할 데이터입니다. 객체 형태로 전달하면 jQuery가 적절한 포맷으로 변환하여 서버에 전송한다.

예: data: { name: "John", location: "Boston" }

7) dataType

- 서버로부터 받을 응답의 데이터 타입을 지정한다. 예를 들어 "json"을 지정하면 응답을 JSON 형태로 자동 변환하여 제공한다.

예: dataType: "json"

8) success

- 요청이 성공적으로 완료되었을 때 호출되는 함수다. 함수는 세 개의 파라미터를 받는다.

data: 서버로부터 반환된 데이터이다.

status: 문자열로 표현된 요청 상태이다.

xhr: 사용된 XMLHttpRequest 객체이다.

9) error

- 요청이 실패했을 때 호출되는 함수이다. 함수는 세 개의 파라미터를 받는다.

xhr: 사용된 XMLHttpRequest 객체이다.

status: 요청 실패 상태를 나타내는 문자열이다.

errorThrown: 예외 정보를 나타내는 문자열이다.

10) complete

- 요청이 성공하든 실패하든, 완료되었을 때 호출되는 함수이다. 함수는 두 개의 파라미터를 받는다.

xhr: 사용된 XMLHttpRequest 객체이다.

status: 요청 완료 상태를 나타내는 문자열이다.