Chap43 Ajax

▼ Ajax란

자바스크립트를 사용하여 브라우저가 서버에게 비동기 방식으로 데이터를 요청하고, 서버가 응답한 데이터를 수신하여 웹페이지를 동적으로 갱신하는 프로그래밍 방식

ajax는 브라우저에서 제공하는 Web API인 XMLHttpReqeust 객체를 기반으로 동작한다. Ajax는 브라우저에서 제공하는 Web API인 XMLHttpRequest 객체를 기반으로 동작하다.

장점

- 변경할 부분을 갱신하는 데 필요한 데이터만 서버로부터 전송받기 때문에 불필요한 데이터 통신이 발생하지 않는다.
- 변경할 필요가 없는 부분은 다시 렌더링하지 않는다. 따라서 화면이 순간적으로 깜박이는 현상이 발생하지 않는다.
- 클라이언트와 서버와의 통신이 비동기 방식으로 동작하기 때문에 서버에게 요청을 보낸 이후 블로킹이 발생하지 않는다.

▼ JSON

클라이언트와 서버 간의 HTTP 통신을 위한 텍스트 데이터 포맷이다. 자바스크립트에 종속되지 않는 언어 독립형 데이터 포맷으로, 대부분의 프로그래밍 언어에서 사용할 수 있다.

▼ 표기 방식

```
{
 "name":"Lee",
 "age":
 "alive":true,
```

```
  "hobby":["traveling","tennis"]
}
```

→ 키는 반드시 큰따옴표로 묶어야 ㅏ 한다.

→ 값은 객체 리터럴과 같은 표기법을 그대로 사용할 수 있다. 하지만 문자열은 큰따옴표로 묶어야 한다.

▼ JSON.stringify

객체,배열을 JSON 포맷의 문자열로 변환한다. 직렬화 해준다.

클라이언트가 서버로 객체를 전송하려면 객체를 문자열화 해야하는데 이를 **직렬화**라고 한다.

▼ JSON.parse

JSON포맷의 문자열을 객체로 변환하는 메서드. 역직렬화 해준다.

서버로부터 클라이언트에게 전송된 JSON데이터는 문자열이다. 이 문자열을 JSON포맷의 문자열을 객체화해야한다. 이를 **역직렬화**라고 한다.

▼ XMLHttpRequest

HTTP 요청 전송과 HTTP 응답 수신을 위한 다양한 메서드와 프로퍼티를 제공한다.

▼ http

하이퍼텍스트 전송 프로토콜 (HTTP) 은 HTML과 같은 하이퍼미디어 문서를 전송하기 위한 애플리케이션 계층 프로토콜입니다

▼ XMLHttpRequest 객체 생성

- XMLHttpRequest 생성자 함수 호출하여 생성
- 브라우저에서 제공하는 Web API이므로 브라우저 환경에서만 정상적으로 시행된다.

```
const xhr=new XMLHttpRequest();
```

▼ HTTP 요청 전송

HTTP 요청을 전송하는 경우 다음 순서를 따른다.

1. XMLHttpRequest.prototype.open 메서드로 HTTP요청을 초기화한다.
2. 필요에 따라 XMLHttpRequest.prototype.setRequestHeader 메서드로 특정 HTTP 요청의 헤더 값을 설정한다.
3. XMLHttpRequest.prototype.send 메서드로 HTTP요청을 전송한다.

▼ XMLHttpRequest.prototype.open

- open 메서드는 서버에 전송할 HTTP 요청을 초기화한다.
- 호출 방법'

```
xhr.open(method,url[,async])
```

method - HTTP 요청 메서드("GET","POST","PUT","DELETE"등)

url - HTTP 요청을 전송할 URL

async - 비동기 요청 여부, 옵션으로 기본값은 true이며, 비동기 방식으로 동작한다.

▼ XMLHttpRequest.prototype.send

- open메서드로 초기화된 HTTP 요청을 서버에 전송한다.
- 기본적으로 서버로 전송하는 데이터는 GET,POST 요청 메서드에 따라 전송 방식에 차이가 있다.
  - GET요청 메서드의 경우 데이터를 URL의 일부분인 쿼리 문자열로 서버에 전송한다.
  - POST 요청 메서드의 경우 데이터를 요청 몸체에 담아 전송한다.
- 요청 몸체에 담아 전송할 데이터를 인수로 전달할 수 있다. 페이로드가 객체인 경우 반드시 JSON.stringify 메서드를 사용하여 직렬화한 다음 전달해야한다.
  - 단, GET인 경우 send메서드에 페이로드로 전달할 인수는 무시되고 요청 몸체는 null로 설정된다.

▼ 페이로드

페이로드(영어: payload)는 **전송되는 '순수한 데이터'**를 뜻한다. 페이로드는 전송의 근본적인 목적이 되는 데이터의 일부분으로 그 데이터와 함께 전송되는 헤더, 메타데이터와 같은 부분은 제외한다. 컴퓨터 보안에서 페이로드는 멀웨어의 일부를 뜻한다.

```
xhr.send(JSON.strigify({id:1,content:'HTML',comple
```

▼ XMLHttpRequest.prototype.setRequestHeader

- 특정 HTTP 요청의 헤더 값을 설정한다.
- 반드시 open메서드를 호출한 이후에 호출해야 한다.
- HTTP 요청 헤더
  - Content-type : 요청 몸체에 담아 전송할 데이터의 MIME 타입의 정보를 표현한다.

MIME 타입	서브타입
text	text/plain, text/html, text/css, text/javascript
application	applicationjson, application/x-www-form-urlencode
multipart	multipart/formed-data

```
    //XMLHttpReqeust 객체 생성
const xhr=new XMLHttpRequest();
//HTTP 요청 초기화
xhr.open('POST','/users');
//HTTP 요청 헤더 설정
//클라이언트가 서버로 전송할 데잍의 MIME 타입 지정:json
xhr.setRequestHeader('content-type','application/j
//HTTP 요청 전송
xhr.send(JOSN.stringify({id:1,content:'HTML',compl
```

  - Accept

```
xhr.setReqeustHeader('accept','application/json');
```

▼ HTTP 응답 처리

서버가 전송한 응답을 처리하려면 **XMLHttpRequest 객체가 발생시키는 이벤트를 캐치**해야 한다.

- 이벤트 핸들러 프로퍼티 중에서 HTTP요청의 현재 상태를 나타내는 readyState프로퍼티 값이 변경된 경우 발생하는 readystatechange 이벤트를 캐치하여 다음과 같이 HTTP응답을 처리할 수 있다.

- readyStatechange 이벤트 대신 load 이벤트를 캐치해도 좋다. load이벤트를 캐치하는 경우 xhr.readyState가 XMLHttpRequest.DONE인지 확인할 필요가 없다.