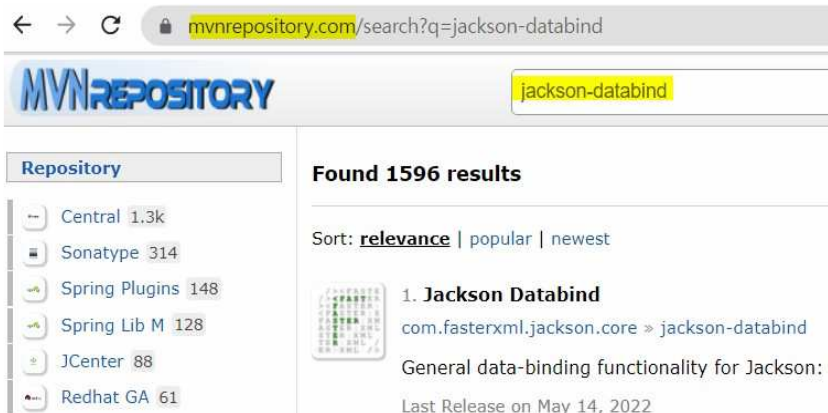


Controller의 리턴타입

- String : jsp를 이용하는 경우에는 jsp파일의 경로와 파일이름을 나타내기 위해서 사용
- void : 호출하는 URL과 동일한 이름의 jsp를 의미
- **VO, DTO 타입** : 주로 **JSON타입**의 데이터를 만들어서 반환하는 용도
- ResponseEntity 타입 : response할 때 Http헤더정보와 내용을 가공하는 용도

객체 타입

Controller의 메서드 리턴 타입을 VO(Value Object)나 DTO(Data Transfer Object)타입 등 복합적인 데이터가 들어간 객체 타입으로 지정할 수 있는데, 이 경우는 주로 JSON 데이터를 만들어 내는 용도로 사용합니다.



- jackson-databind API를 pom.xml에 추가하기 -

```
<!--  
https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-  
databind -->  
<dependency>  
  <groupId>com.fasterxml.jackson.core</groupId>  
  <artifactId>jackson-databind</artifactId>  
  <version>2.9.8</version>  
</dependency>
```

Jackson-databind를 통해 JSON으로 변환

Board

```
@Data
public class Board {
    private int idx; // 번호
    private String title; // 제목
    private String content; // 내용
    private String writer; // 작성자
    private String indate; // 작성일
    private int count; // 조회수
}
```

String(JSON)

← → ↺ localhost:8081/controller/boardList.do 🔍 📄 ⚙️ EX 🏠 매일 ⋮

```
[{"idx":6,"title":"게시판_수정","content":"게시판_수정\r\n게시판_수정\r\n게시판_수정","writer":"박매일","indate":"2022-05-25 21:55:42","count":4}, {"idx":3,"title":"게시판 연습","content":"게시판 연습","writer":"선생님","indate":"2022-05-24 22:16:47","count":0}, {"idx":2,"title":"게시판 연습","content":"게시판 연습","writer":"박매일","indate":"2022-05-24 22:16:46","count":0}, {"idx":1,"title":"게시판 연습","content":"게시판 연습","writer":"관리자","indate":"2022-05-24 22:16:45","count":2}]
```

Board	idx	title	content	writer	indate	count
	6	게시판_수정	게시판_수정	박매일	2022-05-25	4

{ key : value, key : value , , , }

JSON(jackson-databind)

String

```
[{"idx":6,"title":"게시판_수정","content":"게시판_수정\r\n게시판_수정\r\n게시판_수정","writer":"박매일","indate":"2022-05-25 21:55:42","count":4}, {"idx":6,"title":"게시판_수정","content":"게시판_수정\r\n게시판_수정\r\n게시판_수정","writer":"박매일","indate":"2022-05-25 21:55:42","count":4}, {"idx":6,"title":"게시판_수정","content":"게시판_수정\r\n게시판_수정\r\n게시판_수정","writer":"박매일","indate":"2022-05-25 21:55:42","count":4}]
```

@ResponseBody를 통해 클라이언트로 전달

```
@RequestMapping("/boardList.do")
public @ResponseBody List<Board> boardList(){
    List<Board> list=boardMapper.getLists();
    return list;
}
```

List<Board>

```
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-databind</artifactId>
  <version>2.9.8</version>
</dependency>
```

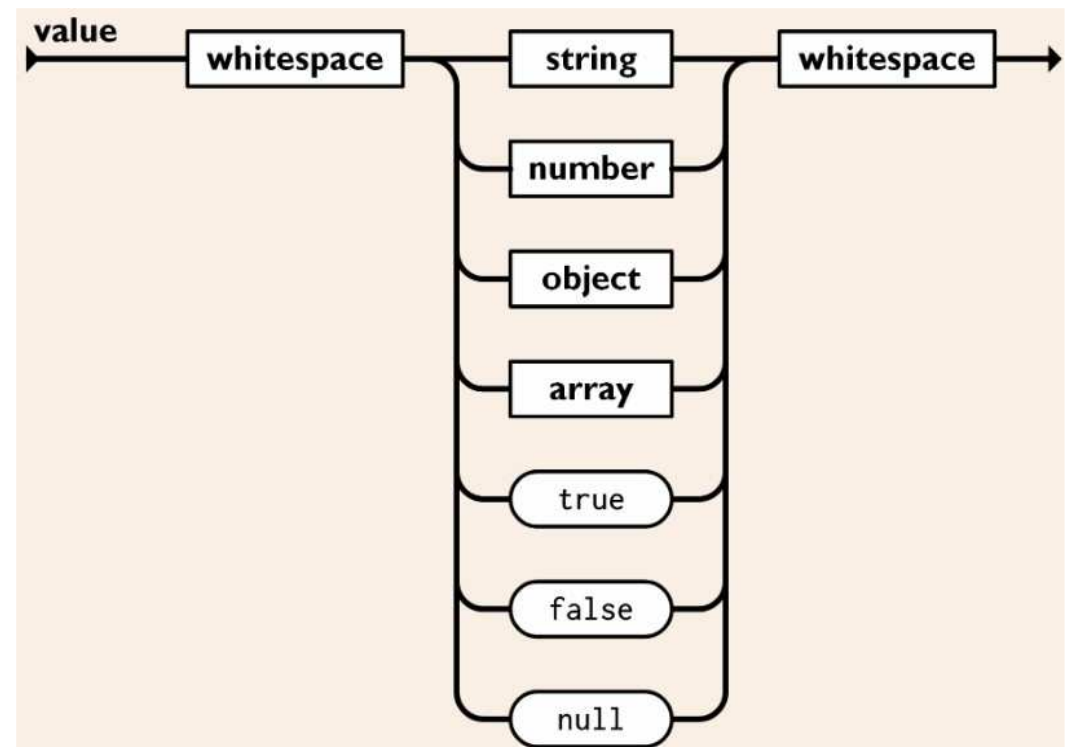
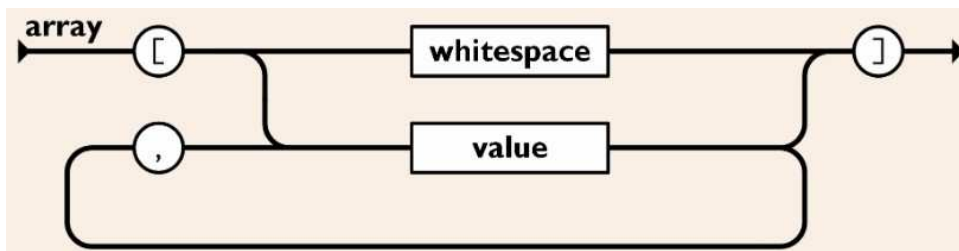
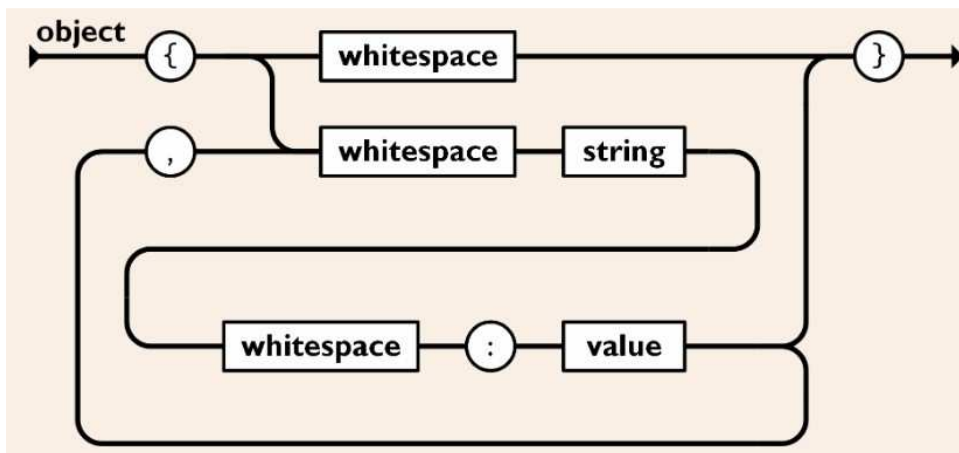
String(JSON)

← → ↻ ⓘ localhost:8081/controller/boardList.do 🔍 ↗ ☆ 📄 📺 S EX ⚙️ 🖱️ 매일 ⋮

```
[{"idx":6,"title":"게시판_수정","content":"게시판_수정\r\n\r\n게시판_수정\r\n\r\n게시판_수정","writer":"박매일","indate":"2022-05-25 21:55:42","count":4}, {"idx":3,"title":"게시판 연습","content":"게시판 연습","writer":"선생님","indate":"2022-05-24 22:16:47","count":0}, {"idx":2,"title":"게시판 연습","content":"게시판 연습","writer":"박매일","indate":"2022-05-24 22:16:46","count":0}, {"idx":1,"title":"게시판 연습","content":"게시판 연습","writer":"관리자","indate":"2022-05-24 22:16:45","count":2}]
```

<https://www.json.org/json-en.html>

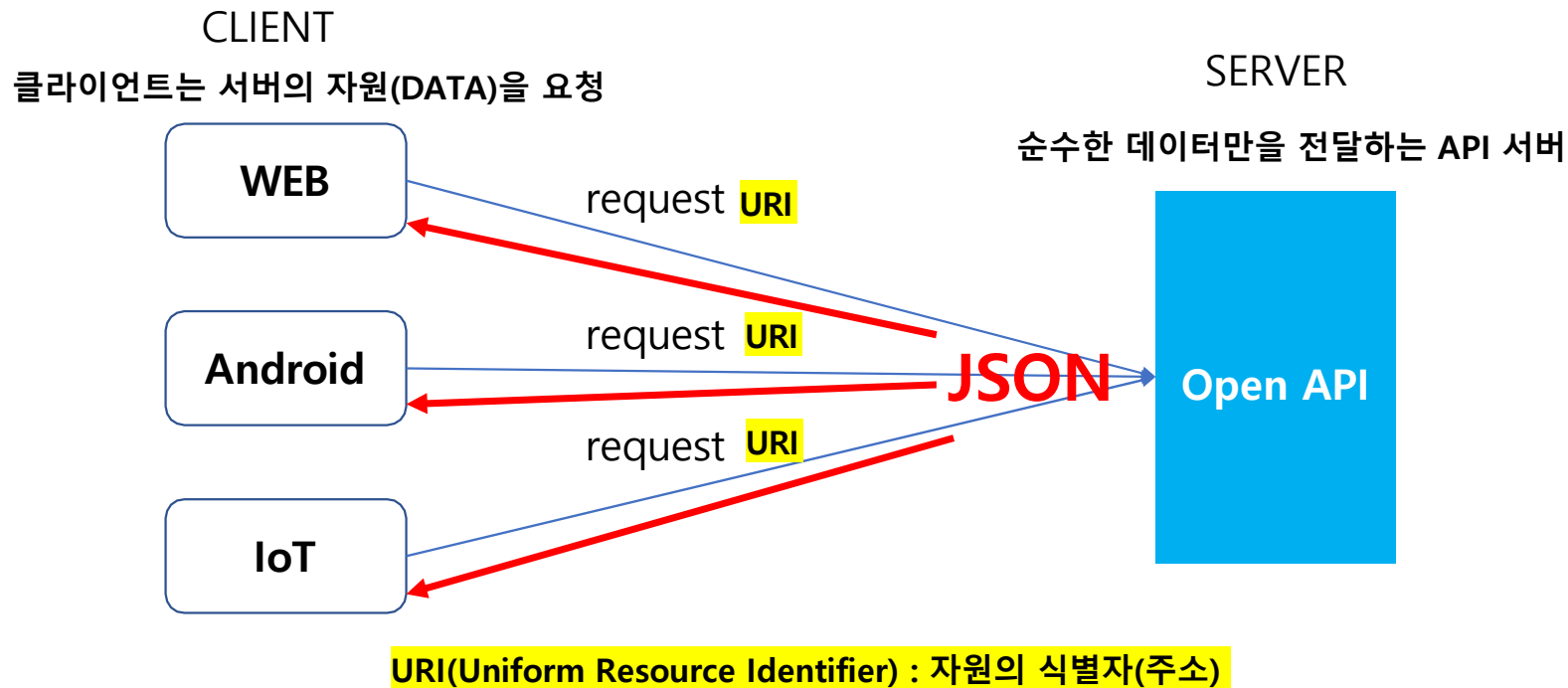
JSON (JavaScript Object Notation) is a lightweight **data-interchange format**.



REST 방식과 Ajax

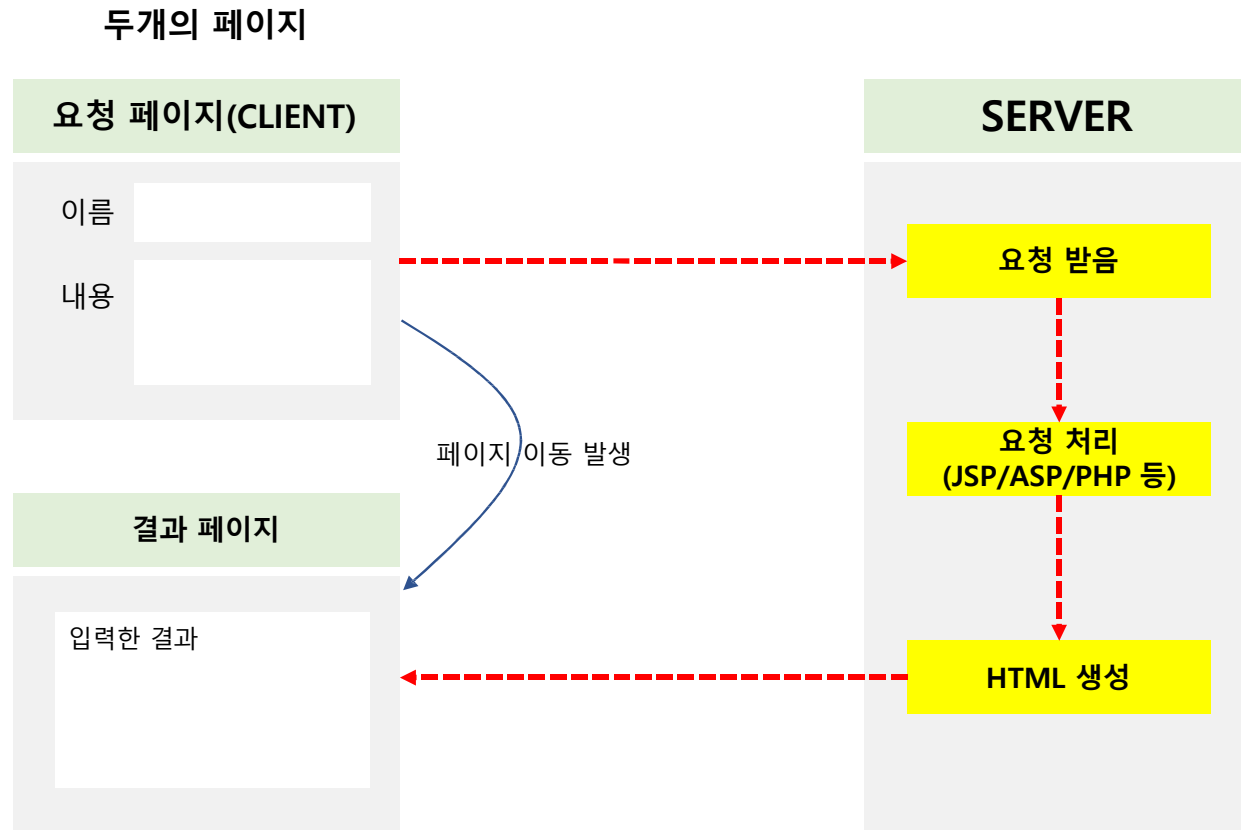
모바일 환경의 대두로 서버 역할의 변화

서버는 브라우저나 모바일에서 필요한 순수한 데이터만을 전달하는 API 서버의 형태로 변화
서버는 클라이언트의 요청 결과를 **XML**이나 **JSON**의 형태로 전달하고, 브라우저나 모바일에서는 이를 가공해서 사용자에게 보여주는 방식



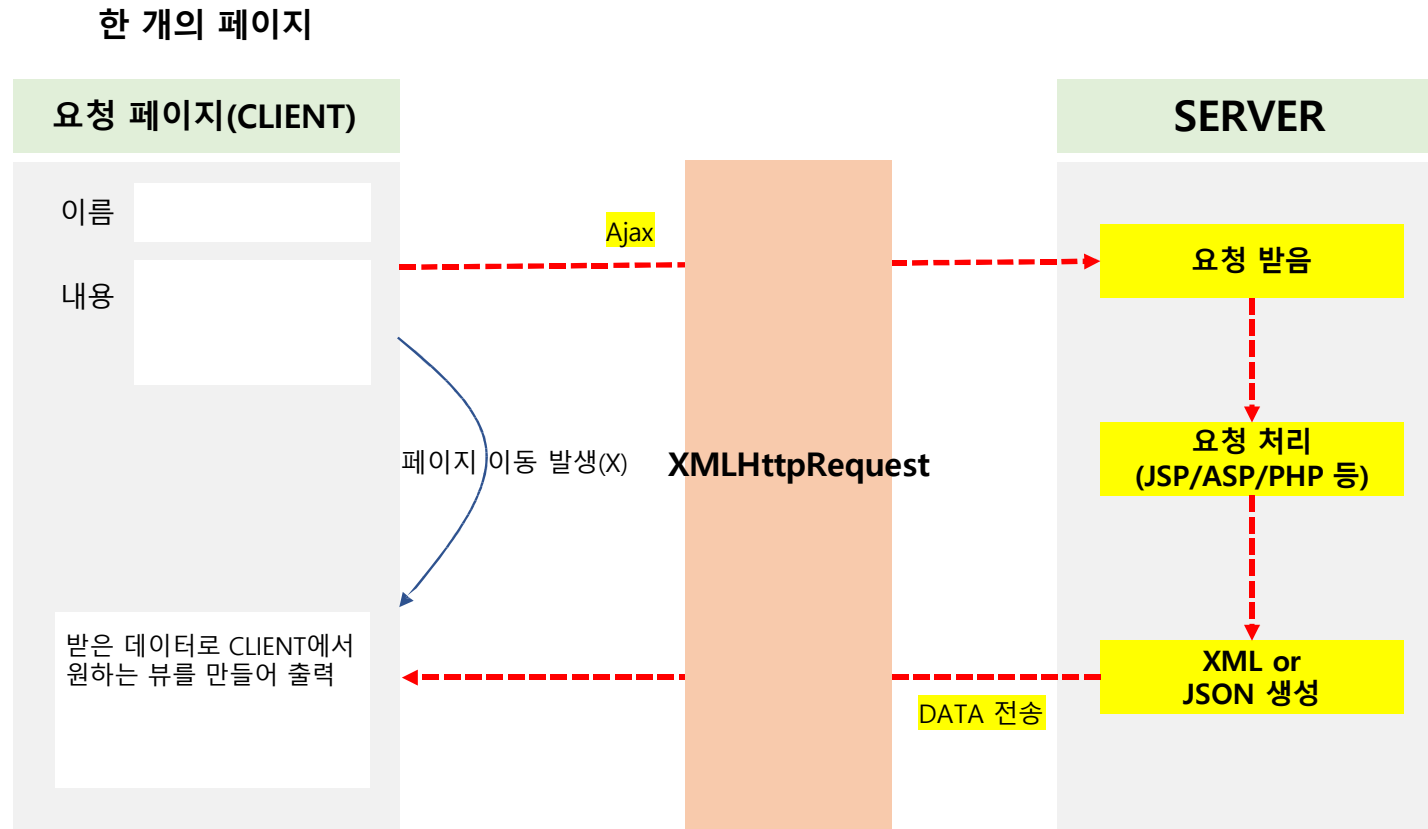
jQuery Ajax 기능

기존 웹 페이지 동작 방식



jQuery Ajax 기능

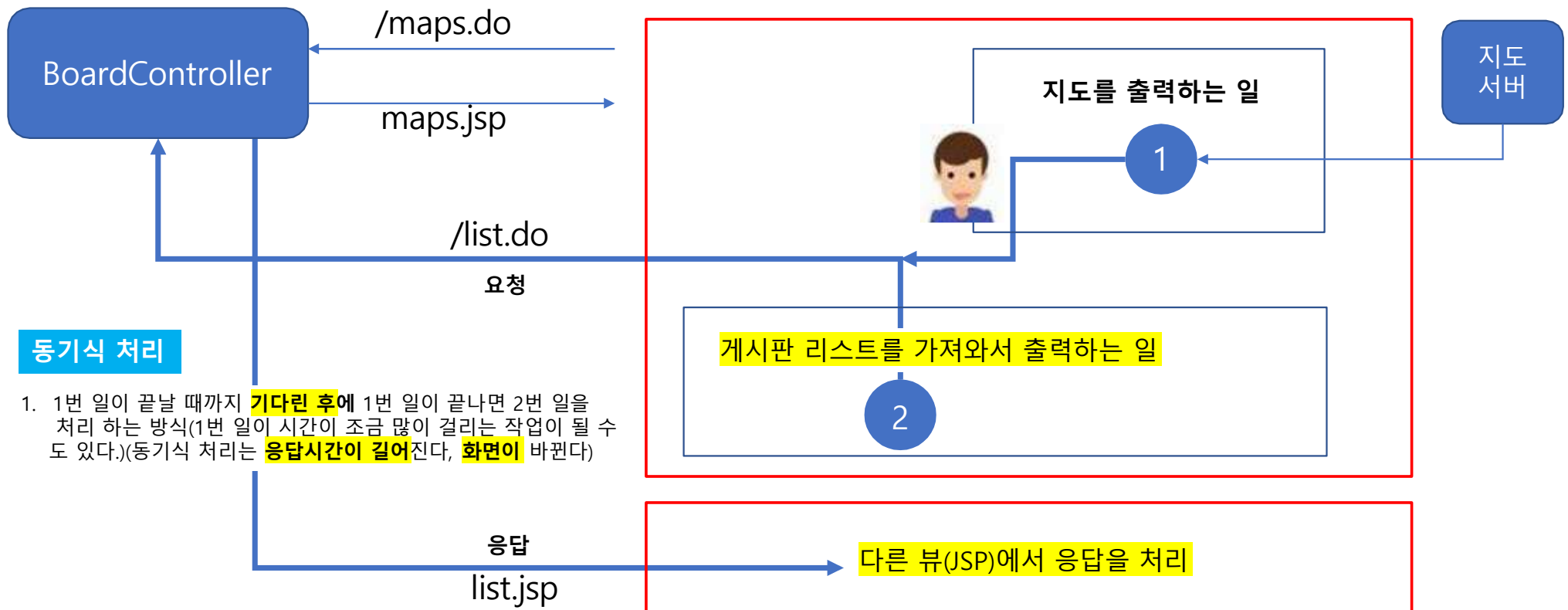
Ajax 웹 페이지 동작 방식



동기식 처리방법

하나의 뷰 페이지에서(JSP) 여러가지 처리를 해야 되는 경우에 어떻게 2가지 일을 처리 할까?
예를 들어서 지도를 출력하는 일, 게시판 리스트를 출력하는 일을 동시에 하는 방법을 생각해보자.

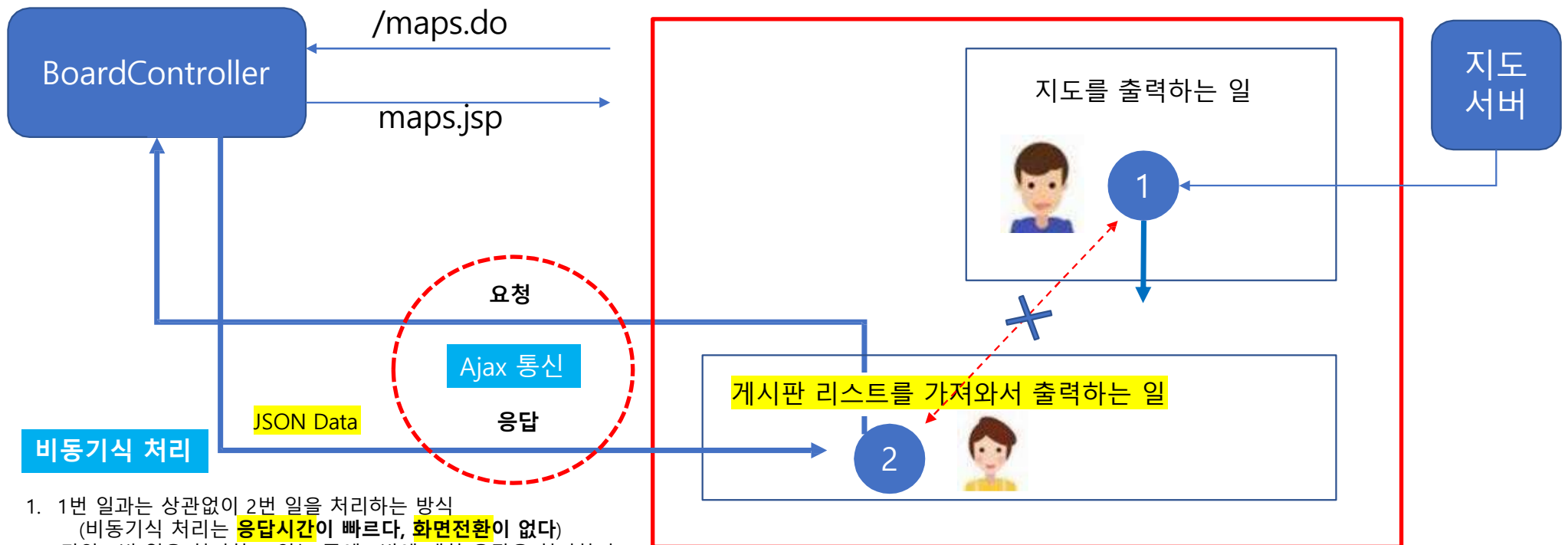
2개의 뷰 페이지(JSP)



비동기식 처리방법

하나의 뷰 페이지에서(JSP) 여러가지 처리를 해야 되는 경우에 어떻게 2가지 일을 처리 할까?
예를 들어서 지도를 출력하는 일, 게시판 리스트를 출력하는 일을 동시에 하는 방법을 생각해보자.

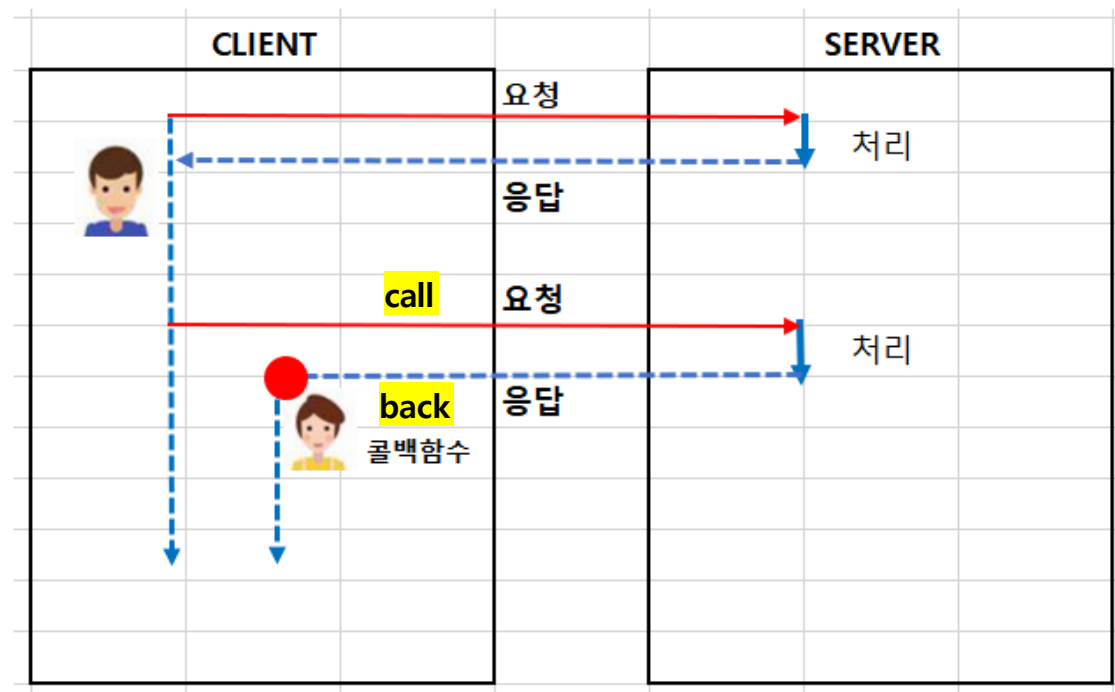
하나의 뷰 페이지(JSP)



1. 1번 일과는 상관없이 2번 일을 처리하는 방식
(비동기식 처리는 **응답시간이 빠르다**, **화면전환이 없다**)
만약 1번 일을 처리하고 있는 중에 2번에 대한 응답을 처리하기 위해서는 콜백함수를 만들어서 처리하게 해야 한다.

콜백함수(callback) 란?

: 클라이언트가 서버로 요청 후 서버에서 응답하는 데이터를 클라이언트에서 받아서 처리하는 함수
-> 클라이언트에서는 JavaScript(자바스크립트)에서 처리한다.



jQuery Ajax 기능

SERVER

```
// @ResponseBody->jackson-databind(객체를->JSON)
@RequestMapping("/boardList.do")
public @ResponseBody List<Board> boardList(){
    List<Board> list=boardMapper.getLists();
    return list; // JSON 데이터 형식으로 변환(API)
}
```

```
18 function loadList(){
19     // 서버와 통신 : 게시판 리스트 가져오기
20     $.ajax({
21         url : "boardList.do",
22         type : "get",
23         dataType : "json",
24         success : makeView,
25         error : function(){ alert("error"); }
26     });
27 }
```

요청

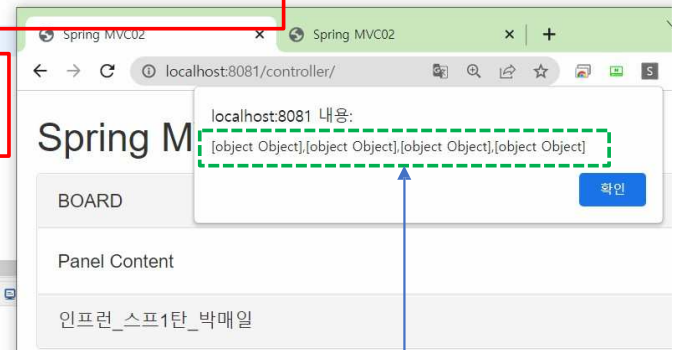
Ajax(비동기전송)

응답(JSON)

```
function makeView(data){
    alert(data);
}
```

```
[{"idx":6,"title":"게시판_수정","content":"게시판_수정\r\n게시판_수정\r\n게시판_수정","writer":"박매일","indate":"2022-05-25 21:55:42","count":4}, {"idx":3,"title":"게시판 연습","content":"게시판 연습","writer":"선생님","indate":"2022-05-24 22:16:47","count":0}, {"idx":2,"title":"게시판 연습","content":"게시판 연습","writer":"박매일","indate":"2022-05-24 22:16:46","count":0}, {"idx":1,"title":"게시판 연습","content":"게시판 연습","writer":"관리자","indate":"2022-05-24 22:16:45","count":2}]
```

CLIENT



jQuery Ajax 기능

Ajax 웹 페이지 동작 방식

```
$.ajax(  
  type : "post" or "get",  
  url : "요청할 URL",  
  data : { 서버로 전송할 데이터},  
  dataType : "서버에서 전송받을 데이터형식",  
  success : function(서버로부터 데이터 받기){  
    // 정상 요청, 응답시 처리  
  },  
  error : function(오류정보){  
    // 오류 발생시 처리  
  }  
);
```

REST(Representational State Transfer : 대표상태전송) = URI + GET/POST/PUT/DELETE/...

REST : 서버의 고유한 리소스를 접근하는 대표 상태 전송

REST 구성

URI

+

GET/POST/PUT/DELETE/...



서버의 고유한 리소스를 접근하는 주소



요청방식(GET, POST 방식)으로 결정

어노테이션	기능
@RestController	Controller가 REST방식을 처리하기 위한 것임을 명시합니다.
@ResponseBody	일반적인 JSP와 같은 뷰로 전달되는 게 아니라 데이터 자체를 전달하기 위한 용도
@PathVariable	URL 경로에 있는 값을 파라미터로 추출하려고 할때 사용
@CrossOrigin	Ajax의 크로스 도메인 문제를 해결해주는 어노테이션
@RequestBody	JSON 데이터를 원하는 타입으로 바인딩 처리

REST 전송방식(GET/POST/PUT/DELETE/...)

작업	전송방식	URI	의미
등록(Create)	POST	/boardInsert.do	등록 해주세요
조회(Read)	GET	/boardContent.do	해당(idx=5) 게시판 내용을 보여주세요
전체조회	GET	/boardList.do	전체리스트를 보여주세요
수정(Update)	POST	/boardUpdate.do	해당(idx=5) 게시판을 수정해주세요.
수정(조회수)	GET	/boardCount.do	해당(idx=5) 게시판 조회수를 누적해주세요.
삭제>Delete)	GET	/boardDelete.do	해당(idx=5) 게시판을 삭제해주세요.

작업	전송방식	URI
등록(Create)	POST	/board/new
조회(Read)	GET	/board/{idx}
전체조회	GET	/board/all
수정(Update)	PUT	/board/update + body(json 데이터 등)
수정(조회수)	PUT	/board/count/{idx}
삭제>Delete)	DELETE	/board/{idx}

Ajax(Asynchronous JavaScript and XML)

Ajax(Asynchronous JavaScript and XML)

: 웹에서 화면을 갱신하지 않고 Server로 부터 Data를 가져와서 동적으로 뷰를 만들 수 있는 방법

key	설명
url	요청이 전송되는 URL이 포함된 문자열 입니다.
type	Http요청방식 입니다. (Get/Post)
timeout	Http요청에 대한 제한 시간을 지정합니다.(단위 : ms)
success	Http요청 성공시 이벤트 핸들러 입니다.
error	Http요청 실패시 이벤트 핸들러 입니다.
complete	Http요청 완료시 이벤트 핸들러 입니다.
data	Http요청 후 return하는 값 입니다.
dataType	Http요청 후 return하는 데이터의 Type을 지정합니다. (xml,html,script,json,jsonp,text)
async	요청시 동기 유무를 선택할 수 있습니다.(True/False)
dataType	return된 데이터의 Type 입니다. (xml,html,json,jsonp,script,text)
cache	브라우저에 의해 요청되는 페이지를 캐시할 수 있습니다 (True/False)
beforeSend	Http요청 전에 발생하는 이벤트 핸들러 입니다.
global	전역함수 활성 여부를 설정합니다. (True/False)

\$.ajax({



Ajax통신시 필요한
대표적인 Property

});