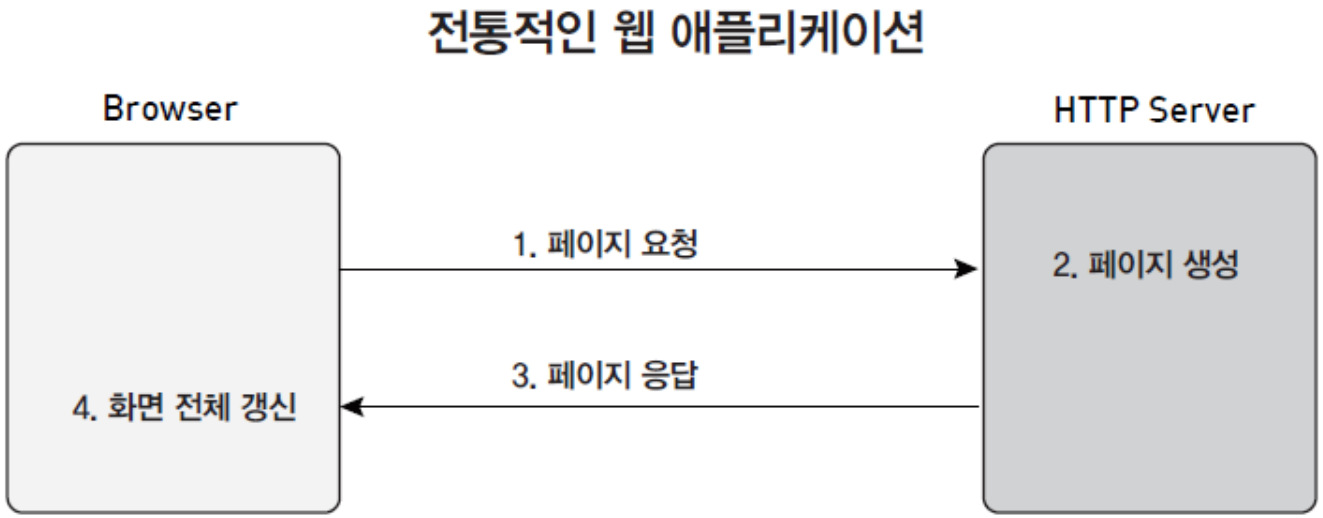
전통적인 웹 애플리케이션

전통적인 웹 애플리케이션

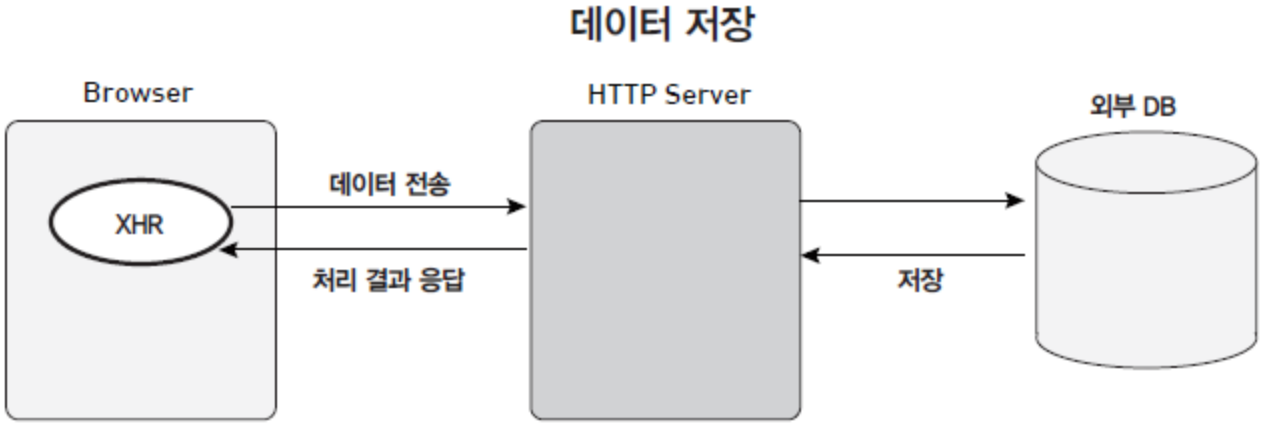
- 요청의 단위가 페이지!
- 서버에서 페이지를 생성해서 응답함.
  - 페이지 단위의 요청과 응답의 문제점은 화면 일부분만 갱신하고 싶어도 페이지 전체를 HTTP 서버로부터 다시 받아와야 한다는 것을 의미
  - HTTP 서버는 브라우저의 요청이 있을 때마다 페이지 전체를 재생성하여 전송

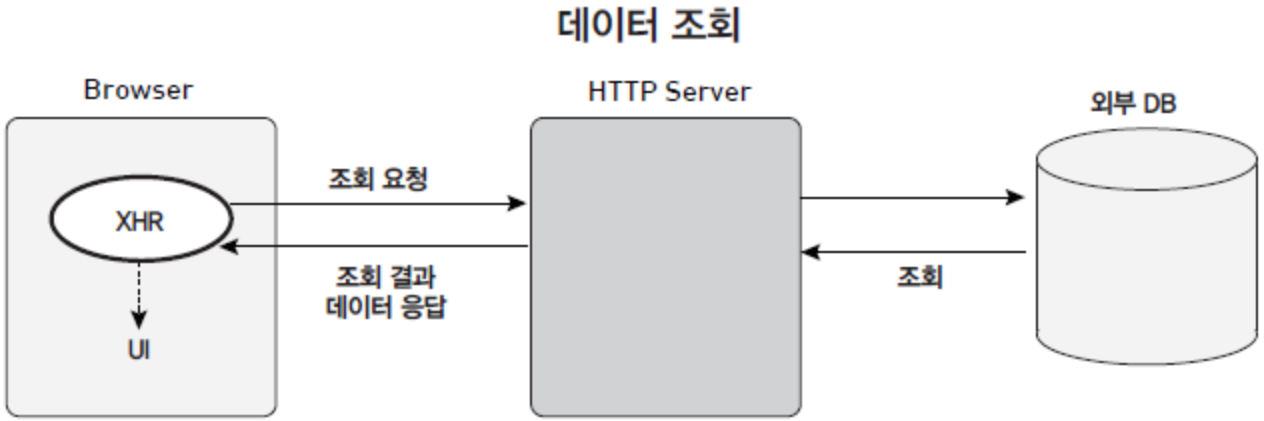


AJAX 개요(1)

- "Asynchronous Javascript And XML"
- Javascript And XML의 의미
  - jQuery는 기본적으로 브라우저에서 작동
    - 외부 데이터를 조회하고 싶거나 데이터를 외부 저장소에 저장하고 싶은 경우에는 jQuery만으로 해결할 수 없다.
    - 브라우저에서 외부 서버와 통신할 수 있는 기능이 필요
  - 브라우저 화면 일부분만 갱신할 수 있어야 한다
    - 전통적인 웹 애플리케이션은 서버에서 HTML 문서 즉 화면 UI를 모두 생성해서 응답하기 때문에 요청 단위가 페이지가 될 수 밖에 없었고, 이로 인해 브라우저 화면의 일부분만 갱신하는 것은 사실상 불가능했다.
  - 문제 해결을 위해 XHR(XMLHTTPRequest) 객체 이용
    - 작업한 내용을 외부의 데이터베이스에 안전하게 저장하기 위해서는 브라우저가 HTTP 서버와 통신할 수 있어야만 한다.
    - 서버와 통신하여 전송한 데이터는 서버측 기술(JSP, 서블릿, ASP, PHP 등)을 이용해 데이터베이스에 저장한다.

AJAX 개요(2)


데이터 저장
Browser
HTTP Server
외부 DB
XHR
데이터 전송
처리 결과 응답
저장


데이터 조회
Browser
HTTP Server
외부 DB
XHR
조회 요청
조회 결과
데이터 응답
조회
UI

AJAX 개요(3)

- XHR 객체를 이용한 요청 단위는 페이지가 아니라 데이터이다.
  - 화면 UI를 생성하는 것은 자바스크립트 코드를 이용한다.
- "Javascript And XML"
  - 자바스크립트 코드를 이용해서 서버와 데이터(XML)를 주고 받겠다는 것이다
  - 지금은 XML보다는 JSON!!

[**표 13-1** : JSON, XML 비교]

구분	JSON	XML
데이터 크기	상대적으로 작다	상대적으로 크다
가독성	좋다	좋다
웹앱 접근성	좋다	상대적으로 나쁘다
스키마 존재 여부	있지만 많이 쓰이지는 않는다. 표준화 작업이 진행 중이다.	존재하며 많이 쓰인다.

AJAX 개요(4)

Asynchronous의 의미

- 비동기적!!
- 예) 음식 먹기
  - 동기적으로 음식 먹기 : 대기시간-15분

    손오공은 짜장면 가게로 가서 짜장면을 주문하고 다른 일을 하지 못하고 5분동안 짜장면이 나오기를 기다린다.

    5분 후에 짜장면이 나오면 5초 동안 먹는다.

    1, 2단계를 반복적으로 수행하여 떡볶이와 초밥을 먹는다.

  - 비동기적으로 음식 먹기 : 대기시간-0분, 주문한 이후에 다른 작업 가능

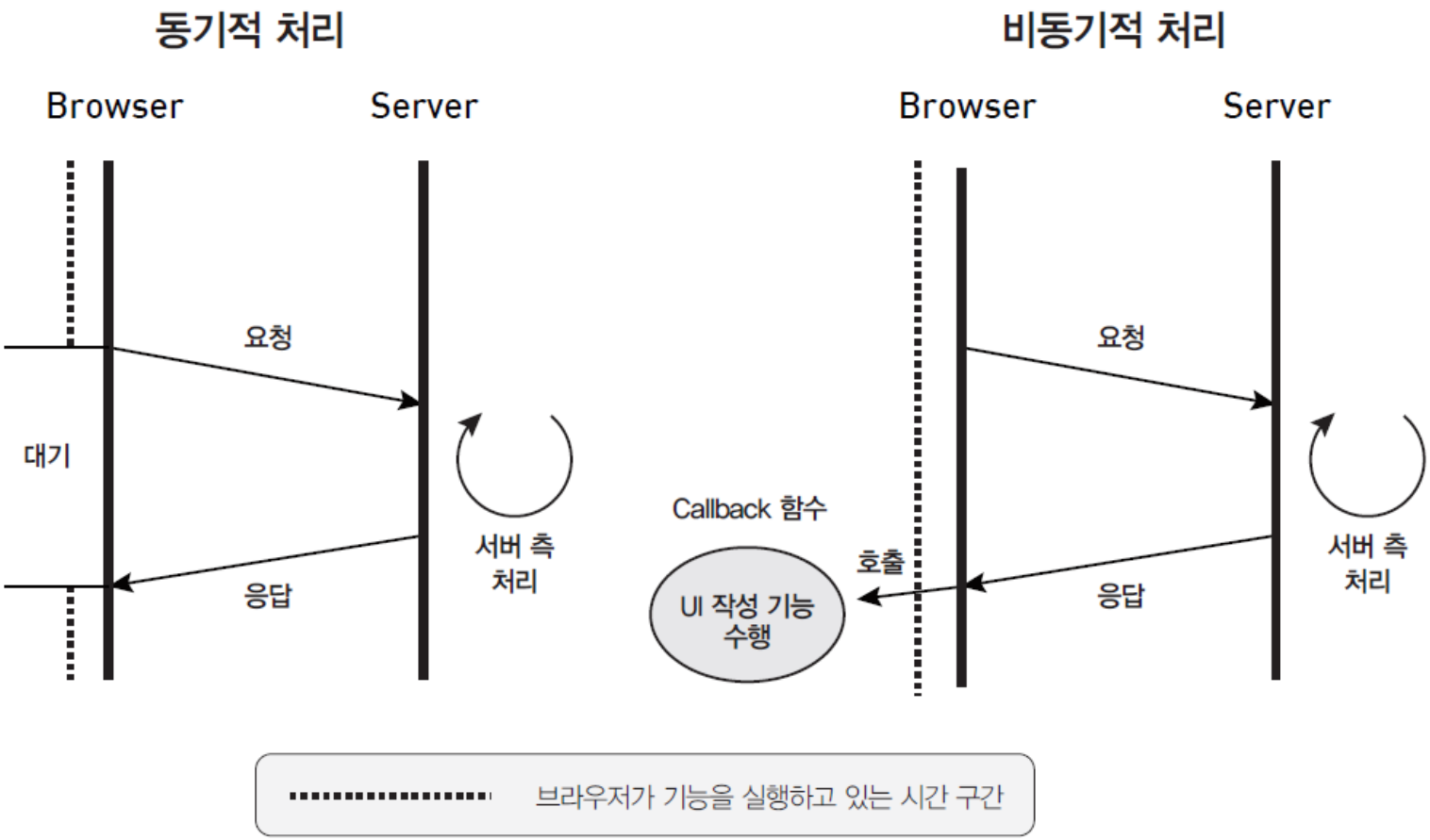
    손오공은 짜장면, 떡볶이, 초밥 가게를 다니면서 주문만 하고 알림벨을 받아온다.

    여유있게 5분동안 자리에 앉아 다른 일(예를 들면 독서, 게임 등)을 한다.

    알림벨이 울리면 울린 순서대로 음식을 받아와 5초 동안 먹는다.

AJAX 개요(5)

- 동기적 vs 비동기적



데이터 서비스 작성(1)

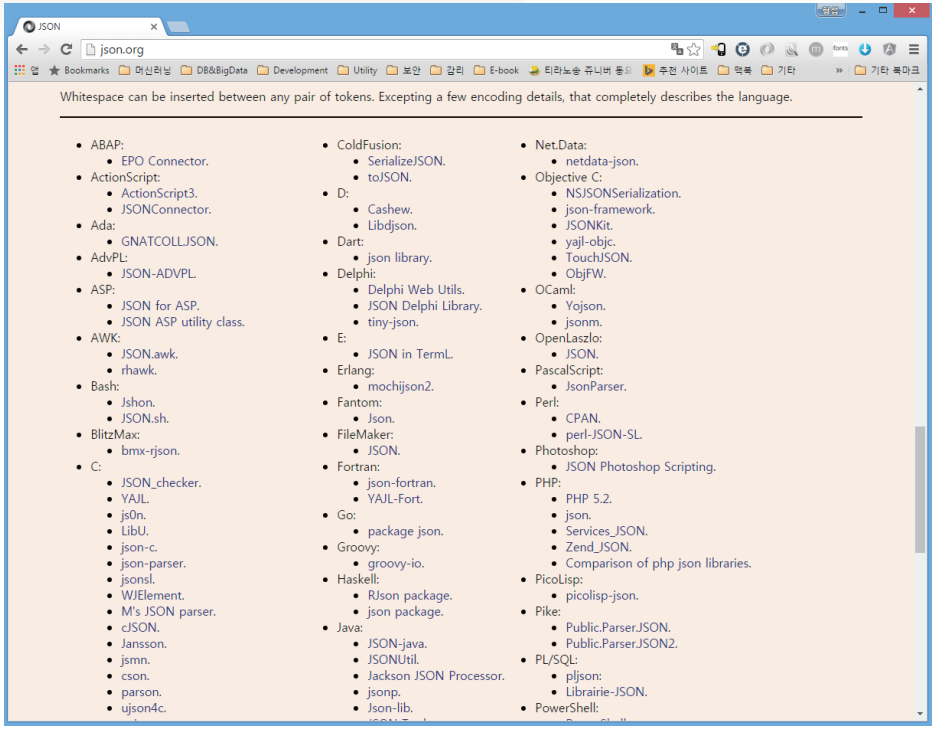
간단한 서비스 작성

- 서비스는 어떤 플랫폼이나 어떤 개발 언어를 사용해도 된다.
  - php,asp, jsp, python+Django, node.js, ASP.NET, ASP.NET MVC 등
  - 이책에서는 JSP를 사용
  - 우리나라 기업에서 자바를 가장 많이 이용하기 때문이다.
  - 하지만 자바, JSP에 대한 자세한 설명은 생략
- jQuery 측에서는 서비스에 대한 입/출력 정보와 정보 형식을 정확하게 파악하는 것이 중요함.
  - 이 책의 예제가 아니라도 자신이 사용할 수 있는 서버 측 스크립트 기술이나 HTTP 서비스 기술이 있다면 그것을 사용해도 좋다.
  - 서비스 작성을 하지 않고 완성된 코드를 복사한 후에 서비스 기능만을 테스트해보고 다음 절로 넘어가도 좋다.

데이터 서비스 작성(2)

JSON 직렬화 라이브러리

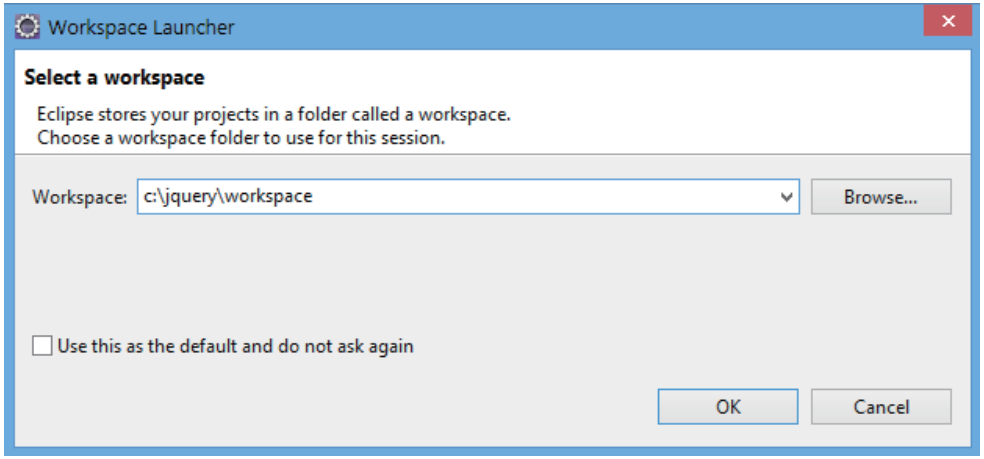
- JSP에서 JSON을 생성하기 위해서 직접 문자열을 이어 붙이는 것은 비효율적이다
- 이 책에서는 google-gson을 사용함.
- http://json.org 에서 각 언어별 라이브러리 확인 가능



데이터 서비스 작성(3)

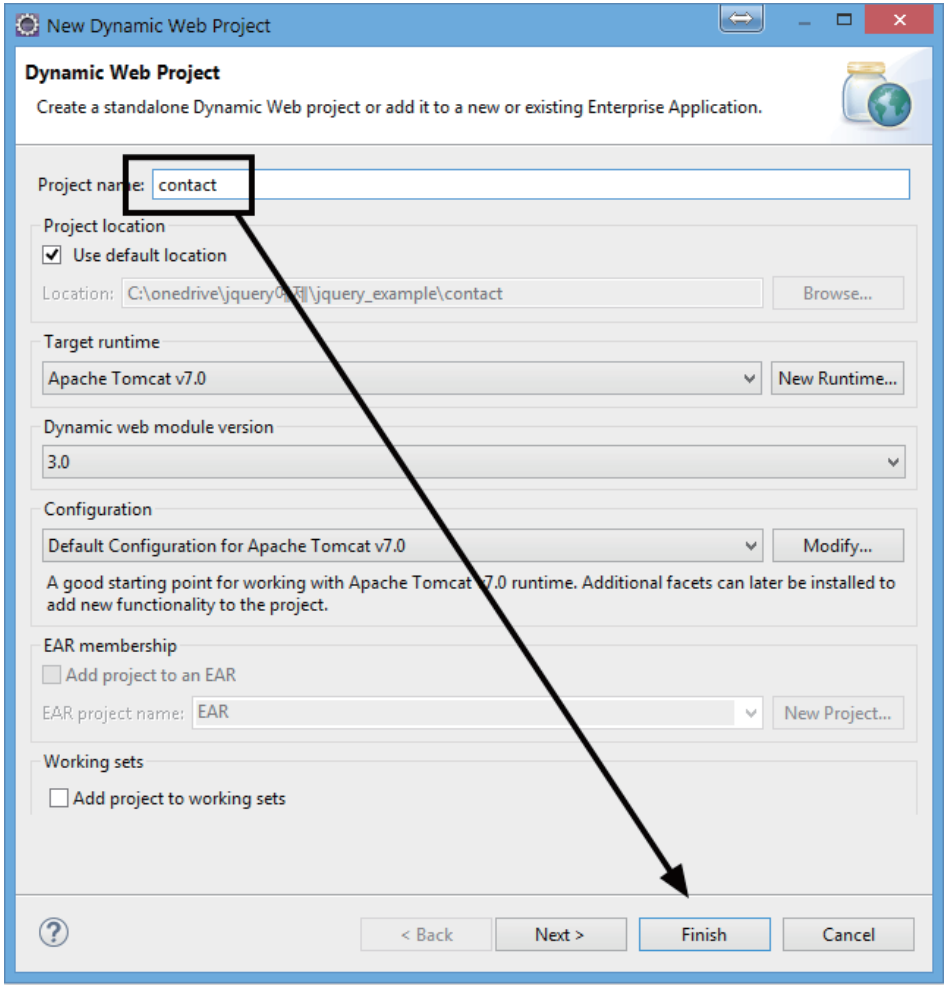
프로젝트 생성 및 초기화

- 개발환경 설정은 1장을 다시 참조
- 이클립스 실행
  - Workspace 지정 : 적절한 디렉토리 지정



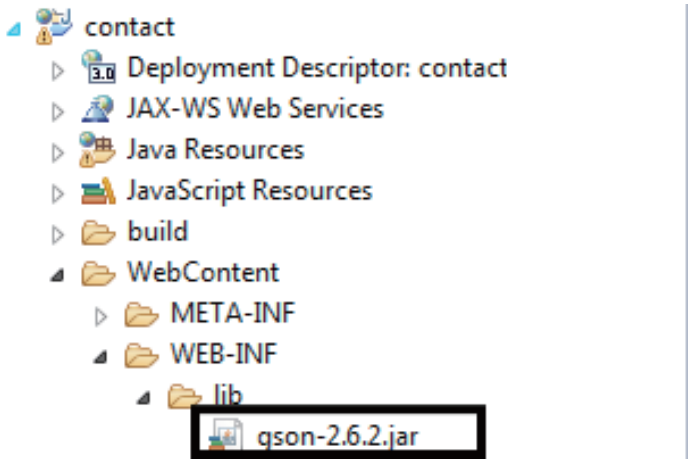
데이터 서비스 작성(4)

- 연락처 서비스 작성
  - Dynamic Web Project 생성
  - 데이터베이스를 이용하지 않고 메모리에 데이터를 저장하는 방법을 사용함.

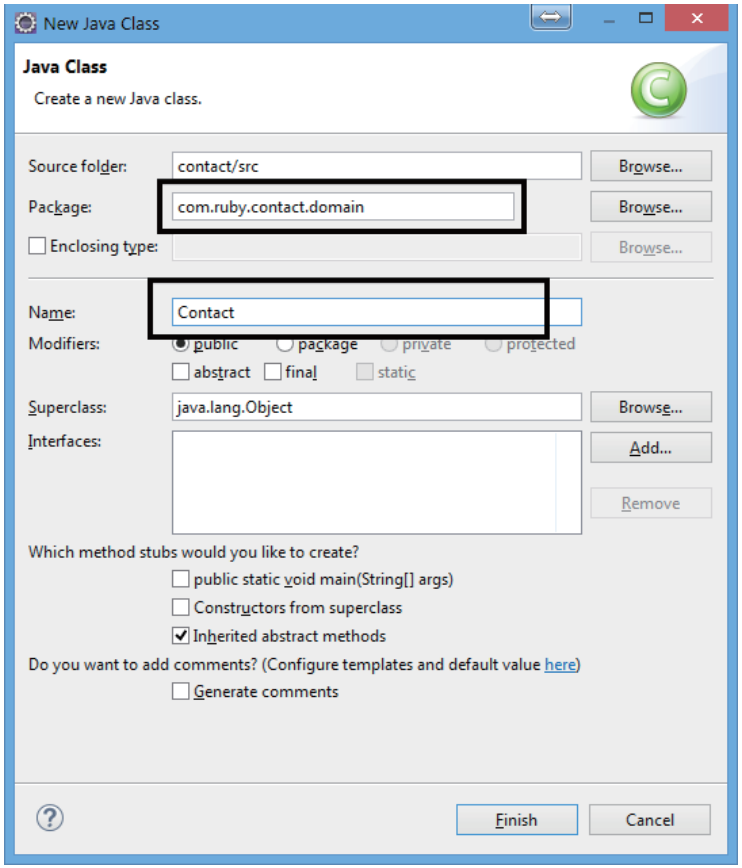


데이터 서비스 작성(5)

- gson 라이브러리를 다운로드하여 프로젝트에 참조



- VO 클래스 준비

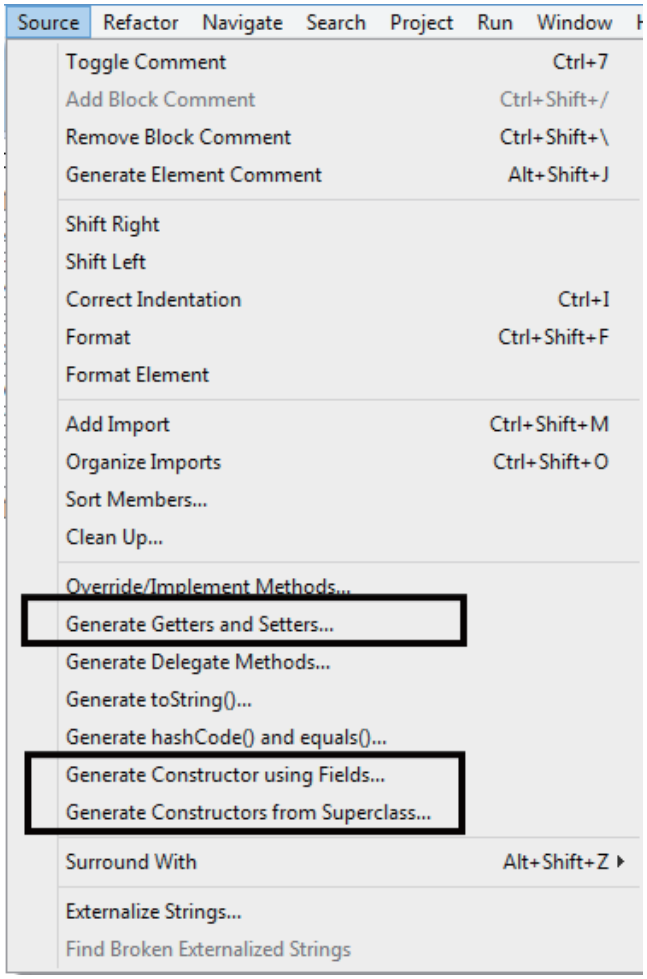


데이터 서비스 작성(6)

- VO 클래스 준비(이어서)

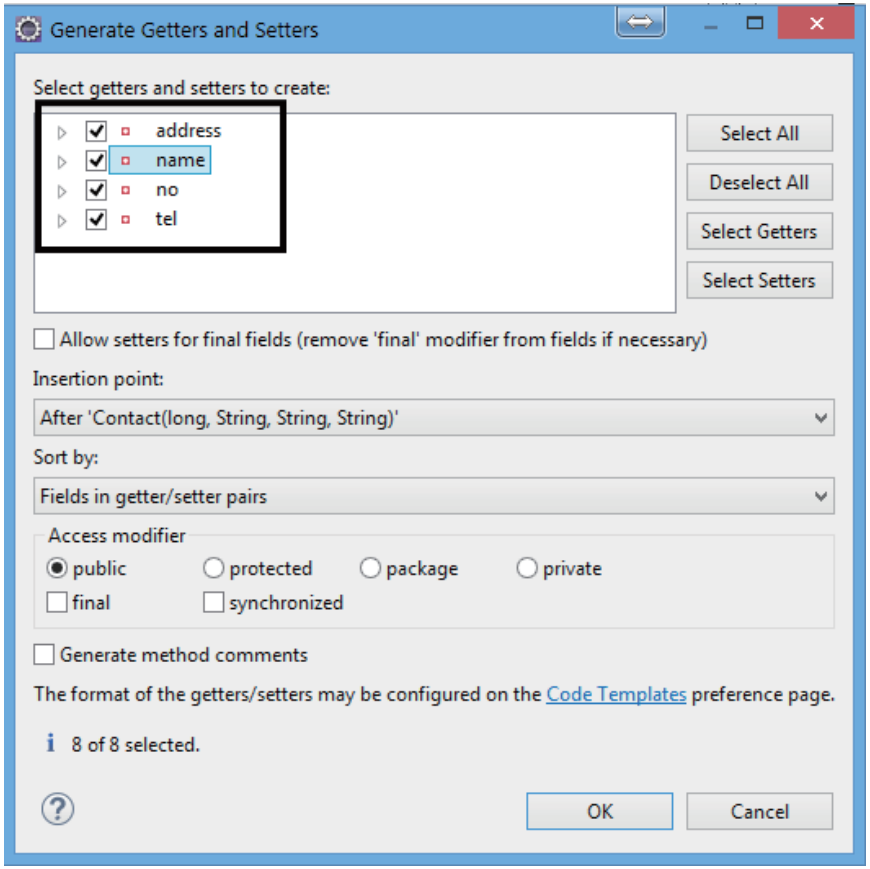
[예제 13-01 : Contact.java]

```
01: package com.ruby.contact.domain;
02:
03: public class Contact {
04:   private long no;
05:   private String name;
06:   private String tel;
07:   private String address;
08:
09:
10: }
```



데이터 서비스 작성(7)

- Getter & Setter 생성



데이터 서비스 작성(8)

- Contact 객체가 생성하는 JSON

```
{ "no" : 1, "name" : "홍길동", tel : "010-1212-3232", "address" : "서울시" }
```

- Contact 여러건 : List<Contact> 컬렉션

```
[
   { "no" : 1, "name" : "홍길동", tel : "010-1212-3232", "address" : "서울시" },
   { "no" : 2, "name" : "이몽룡", tel : "010-1212-3233", "address" : "경기도" },
   { "no" : 3, "name" : "성춘향", tel : "010-1212-3234", "address" : "제주도" },
   ......
]
```

- 추가적인 정보까지 포함하려면? 다음 슬라이드로

데이터 서비스 작성(9)

- ContactList 클래스

```
{
   "pageNo" : 3,
   "pageSize" : 5,
   "totalCount" : 34,
   "contacts" : [
      { "no" : 1, "name" : "홍길동", tel : "010-1212-3232", "address" : "서울시" },
      { "no" : 2, "name" : "이몽룡", tel : "010-1212-3233", "address" : "경기도" },
      { "no" : 3, "name" : "성춘향", tel : "010-1212-3234", "address" : "제주도" },
      ......
   ]
}
```

- 생성자와 Getter, Setter까지 작성

[예제 13-03 : ContactList.java]

```
01: package com.ruby.contact.domain;
02:
03: import java.util.List;
04:
05: public class ContactList {
06:   private int pageNo;
07:   private int pageSize;
08:   private int totalCount;
09:   private List<Contact> contacts;
10:
11:
12: }
```

데이터 서비스 작성(10)

Util 클래스 작성

- Converter 클래스
  - JSON 문자열 <--> 자바 객체

[예제 13-04 : Converter.java]

```
01: package com.ruby.contact.util;
02:
03: import java.io.BufferedReader;
04: import java.io.InputStream;
05: import java.io.InputStreamReader;
06: import java.io.Reader;
07: import java.io.UnsupportedEncodingException;
08:
09: import com.google.gson.Gson;
10:
11: public class Converter {
12:   private static Gson gson;
13:   static {
14:      gson = new Gson();
15:   }
16:
17:   public static String convertToJson(Object obj) {
18:      return gson.toJson(obj);
19:   }
20:
21:   public static <T> T convertFromJson(String json, Class<T> type) {
22:      return gson.fromJson(json, type);
23:   }
24:
25:   public static <T> T convertFromJSONStream(InputStream is, Class<T> type) {
26:      try {
27:         Reader reader = new BufferedReader(
28:            new InputStreamReader(is, "UTF-8"));
29:         return gson.fromJson(reader, type);
30:      } catch (UnsupportedEncodingException e) {
31:         e.printStackTrace();
32:         return null;
33:      }
34:   }
35: }
```

데이터 서비스 작성(11)

- Converter 클래스 (이어서)

converToJson( )	자바 객체를 파라미터로 전달하여 호출하면 JSON 문자열을 생성하여 리턴한다.
convertFromJson( )	JSON 문자열과 클래스 정보를 전달하면 전달한 클래스 형식의 객체로 변환하여 리턴한다.
convertFromJSONStream( )	입력 스트림 객체와 클래스 정보를 전달하면 전달한 클래스 형식의 객체를 생성하여 리턴한다.

데이터 서비스 작성(12)

- SampleDAO 클래스
  - 코드 생략
  - 메서드에 대한 간략한 설명

addContact( ) Contact 객체를 파라미터로 전달하여 새로운 연락처 정보를 추가한다. 이때 일련 번호는 자동으로 부여될 수 있도록 전역 변수 nextNo를 1씩 증가시켜 사용한다.

deleteContact( ) 연락처의 일련 번호를 파라미터로 전달하여 연락처 정보를 삭제한다.

updateContact( ) 연락처 정보를 수정한다.

updateBatch( ) ContactList 객체를 파라미터로 전달하여 한 번에 여러 건의 연락처 정보를 수정한다.

getContacts( ) 모든 연락처 리스트 정보를 ContactList 객체에 담아 리턴한다.

getContacts(pageno, pagesize) 특정 페이지의 연락처 리스트 정보를 ContactList 객체에 담아 리턴한다.

데이터 서비스 작성(13)

각 페이지 작성

조회 페이지

- list.jsp : 예제 13-06
- 응답 Content Type을 application/json 으로 지정

```
list.jsp

list.jsp?pageno=2&pagesize=5
```

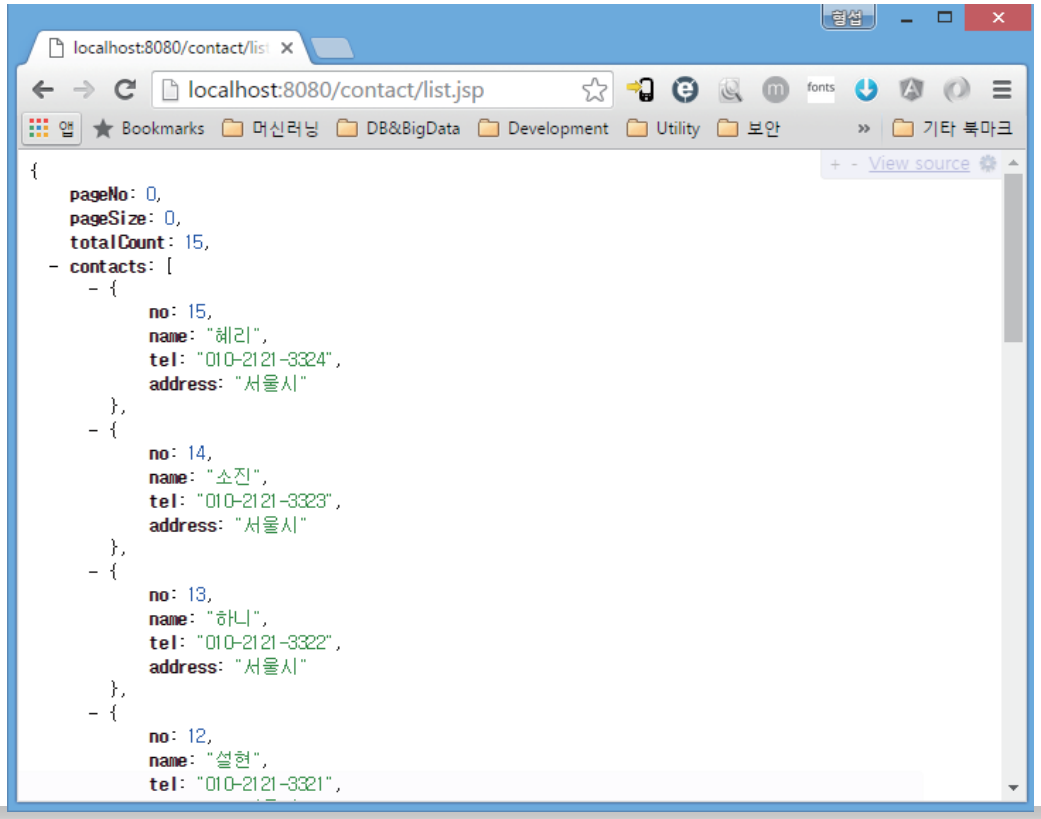
- 쿼리 문자열이 없는 경우 전체 데이터 조회
- pageno, pagesize 쿼리 문자열이 전달된 경우 특정 페이지 데이터 조회
- 받아낸 데이터를 Converter 클래스의 convertToJson() 메서드를 호출해 JSON으로 변환한 다음 응답함

데이터 서비스 작성(14)

- 연락처 추가 페이지 작성
  - 예제 13-07
  - POST 메서드로만 실행
  - name, tel, address 값을 읽어들여 Contact 객체를 만든 다음 SampleDAO의 addContact() 메서드를 호출하여 추가
  - 추가 작업의 성공/실패 여부는 JSON 문자열로 응답
- 수정, 삭제 페이지 작성
  - 예제 13-08~09
  - 추가 페이지와 작동 방식 유사함
  - 일련번호 값을 이용해 수정/삭제함.
- 배치 업데이트 페이지 작성
  - 예제 13-10
  - application/x-www-form-urlencoded 형식 대신에 application/json 형식의 데이터를 수신하도록 설정
  - 한번에 여러건의 데이터를 처리하기에 적합함.

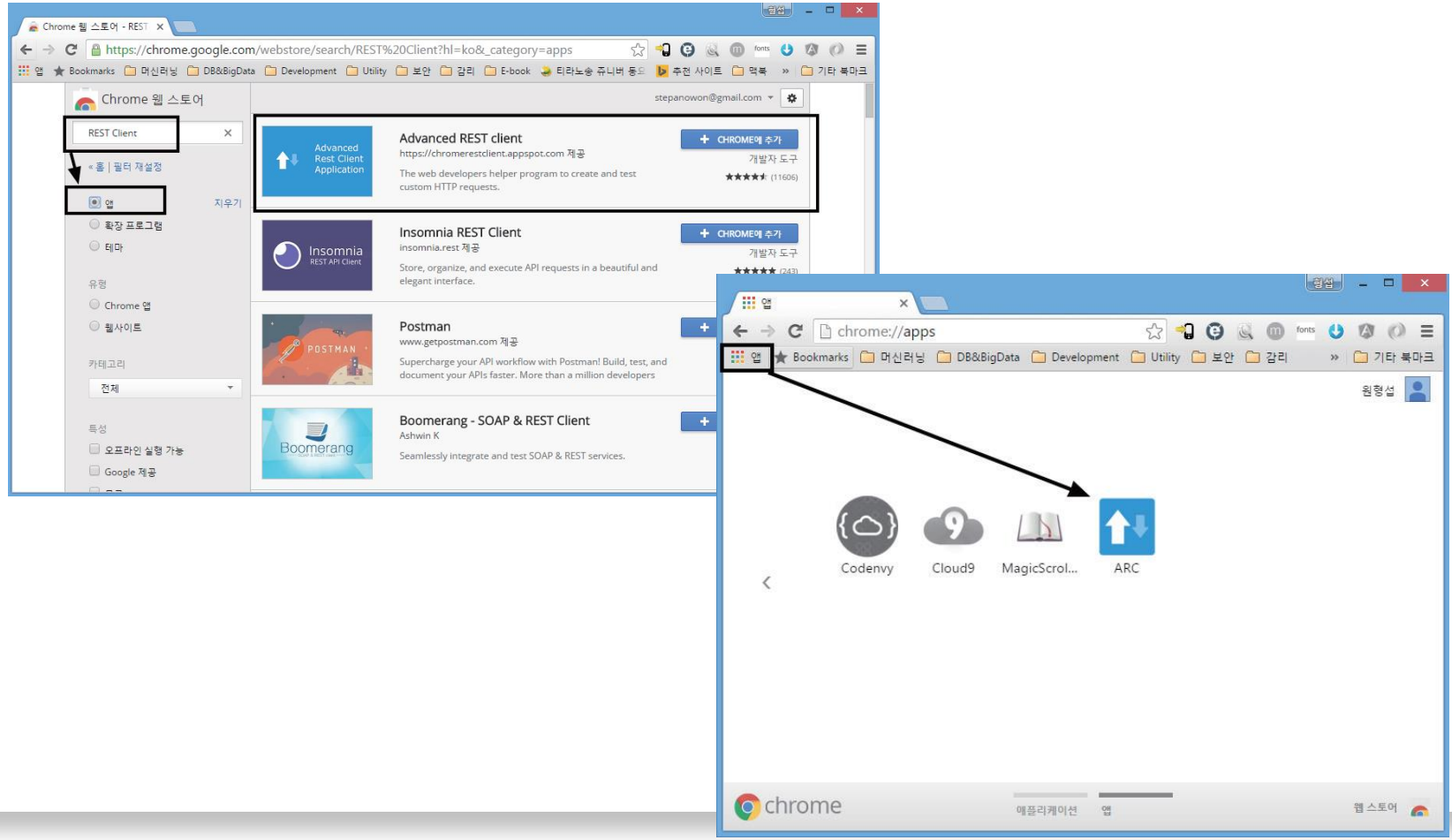
연락처 서비스 테스트(1)

- 서비스 기능의 테스트는 반드시 필요
  - 테스트를 수행하기 위한 크롬 확장 프로그램 설치
  - JSONView



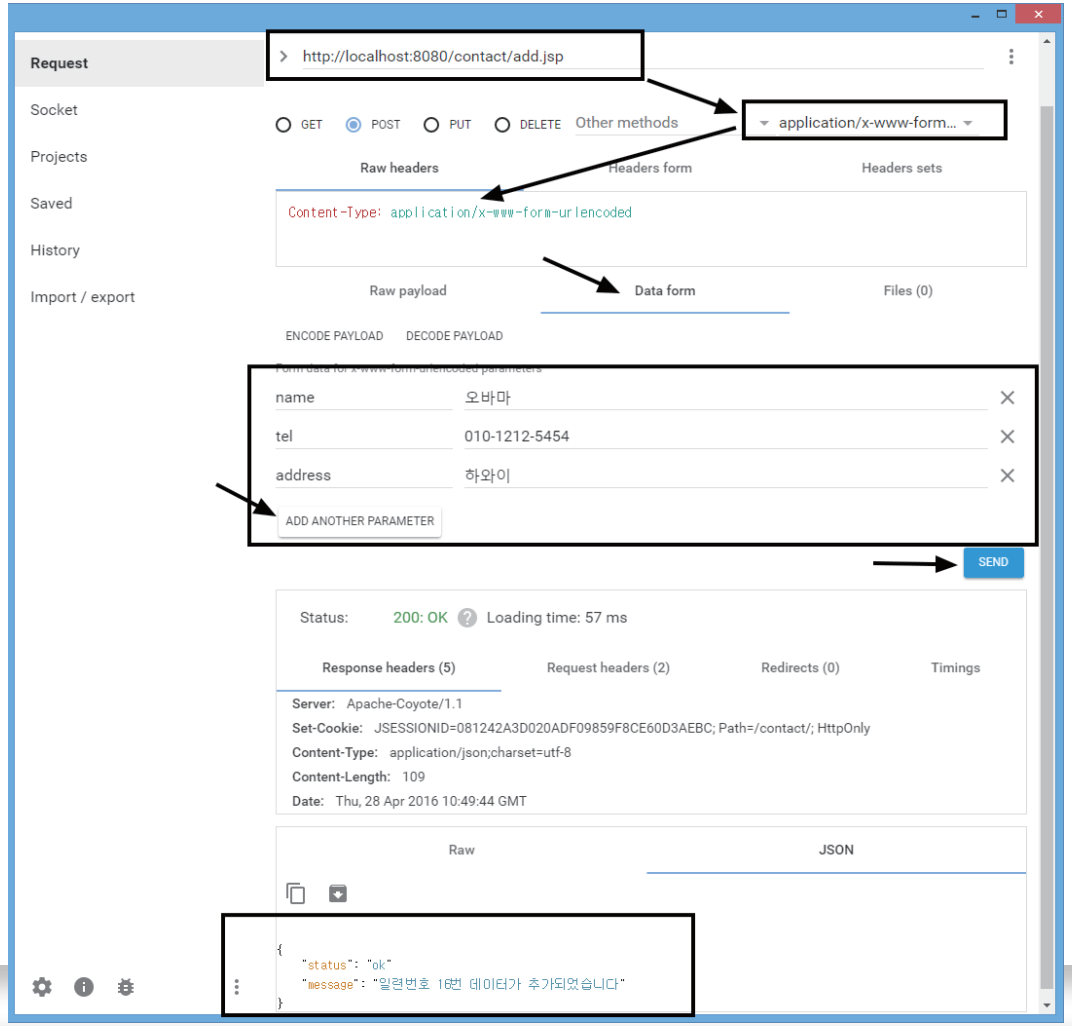
연락처 서비스 테스트(2)

- ARC(Advanced REST Client) 설치
  - 크롬웹스토어에서 Advanced REST Client 설치



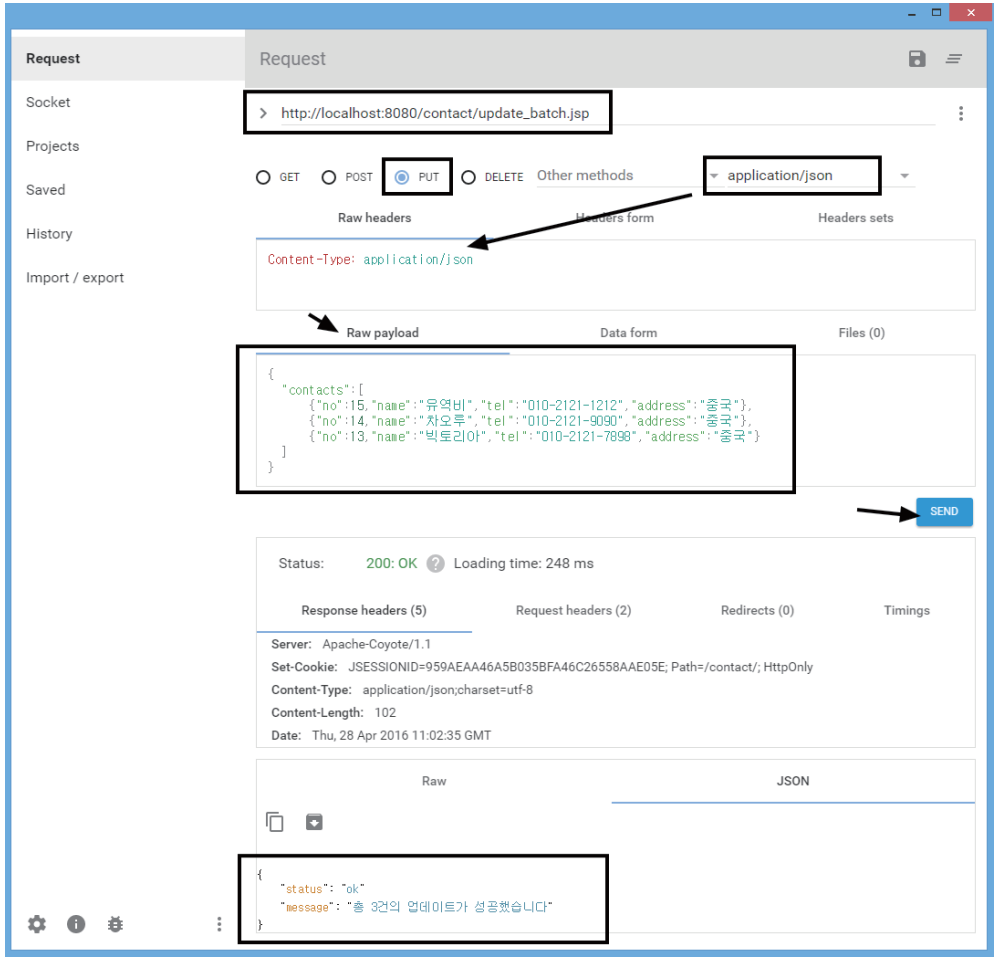
연락처 서비스 테스트(3)

- 추가 기능 테스트



연락처 서비스 테스트(4)

- 배치 업데이트 테스트
  - 배치 업데이트를 수행하는 update_batch.jsp를 테스트하는 방법은 Content-Type과 입력하는 데이터가 달라진다.
  - POST 메서드(PUT 아님)
  - Content Type 을 application/json으로 지정
  - Raw Payload에 JSON 문자열을 직접 전달함.



연락처 서비스 테스트(5)

- jQuery AJAX 애플리케이션을 개발하는 개발자는 REST 클라이언트를 사용할 수 있어야 함.
  - 웹 애플리케이션을 개발하기 전에 서비스의 기능을 테스트하고 요청, 응답 전송 데이터의 형식을 파악해야 하기 때문이다.
- 파악한 요청, 응답 전송 데이터의 형식은 문서화하는 것이 바람직함.
  - 문서화 예

[표 13-02 : 연락처 조회 기능 요청 정보 형식]

파라미터	타입	설명	기본값(Default Value)
pageno	Number	페이지 번호	0 0인 경우 전체 데이터 조회 시도
pagesize	Number	한 페이지 크기	3

[표 13-03 : 연락처 조회 기능 응답 정보 형식]

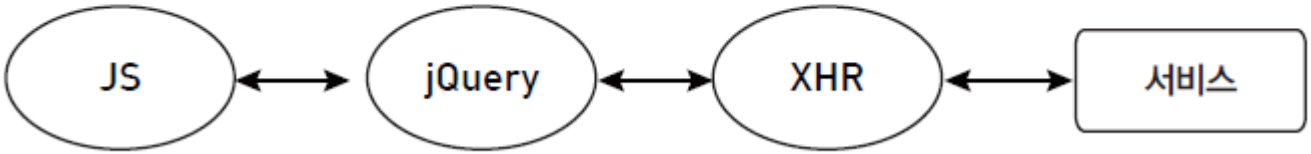
출력 필드	타입	설명	부모 요소
pageNo	Number	페이지 번호 0인 경우 전체 조회를 의미함	-
pageSize	Number	한 페이지 크기 0인 경우 전체 조회를 의미함	-
contacts	Array	연락처 리스트	-
no	Number	일련번호	contacts
name	String	이름	contacts
tel	String	전화번호	contacts
address	String	주소	contacts


```
{
    pageNo: 3,
    pageSize: 2,
    totalCount: 18,
  - contacts: [
      - {
            no: 14,
            name: "소진",
            tel: "010-2121-3323",
            address: "서울시"
        },
      - {
            no: 13,
            name: "하니",
            tel: "010-2121-3322",
            address: "서울시"
        }
    ]
}
```

XMLHttpRequest

- XHR(XMLHttpRequest) 객체를 직접 이용하는 것은 비효율적이다.
  - jQuery와 같은 라이브러리를 이용하는 것이 훨씬 간결하고 편리하다.

XHR 객체 직접 이용



jQuery AJAX 이용

  - 이 책에서는 jQuery를 이용한 AJAX 처리 방식만을 다룬다.