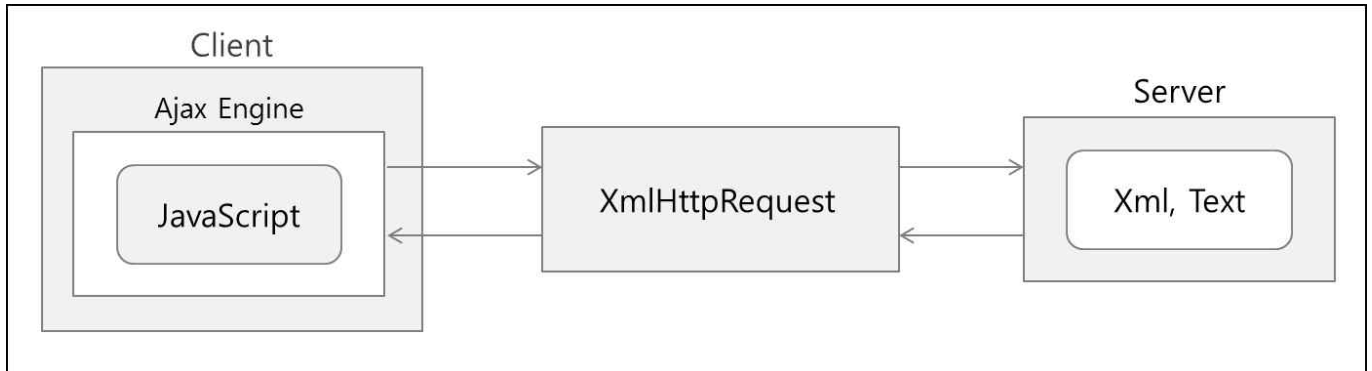
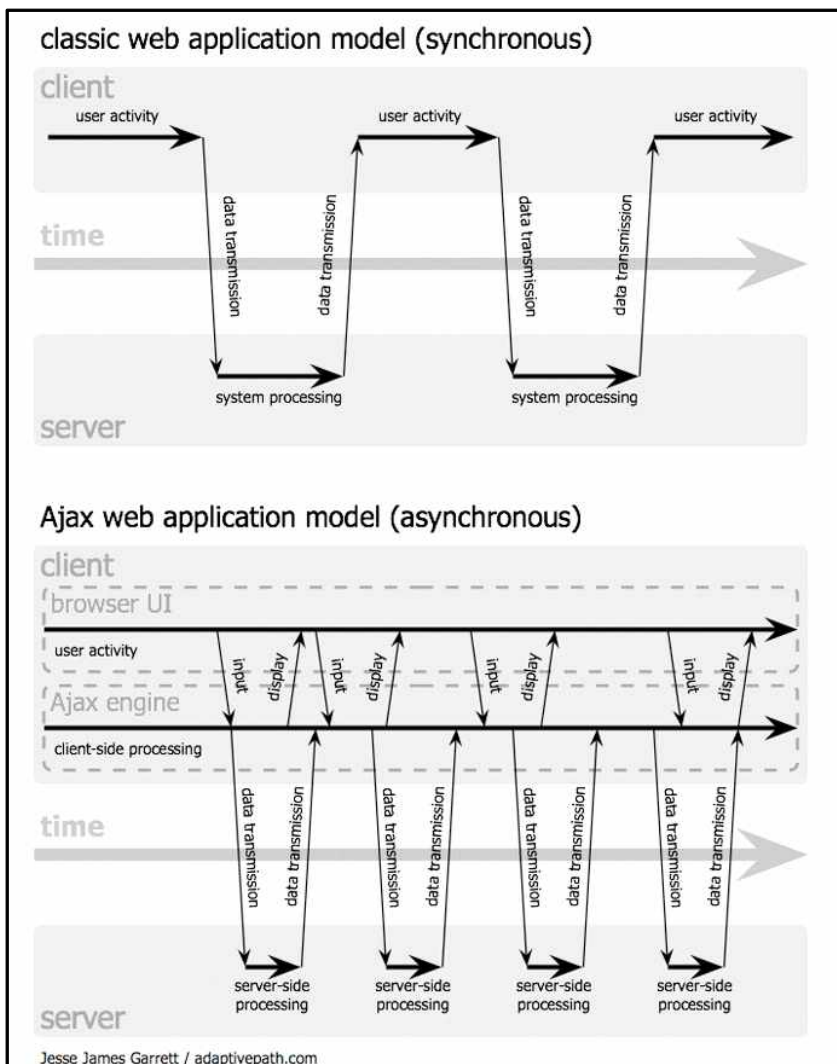


Ajax(Asynchronous Javascript and XML)

- Web에서 화면을 갱신하지 않고 Server로부터 Data를 가져오는 방법



- 자바스크립트를 이용한 비동기 통신

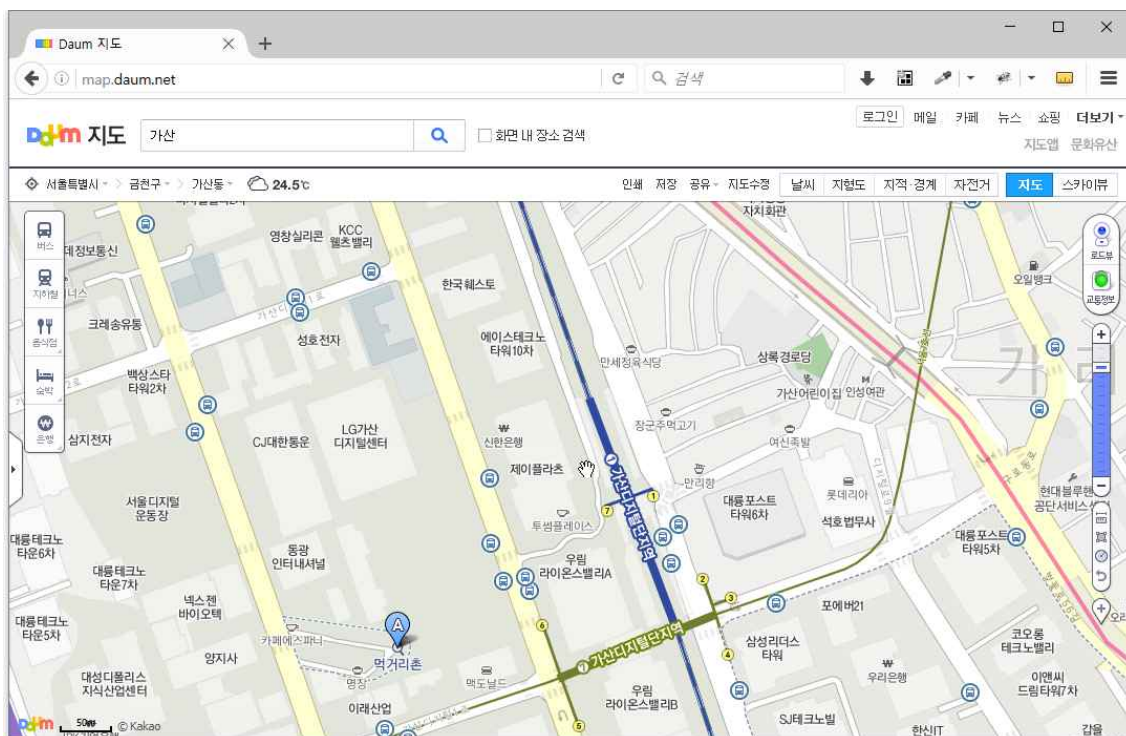


■ 전통적인 웹 접근 방법

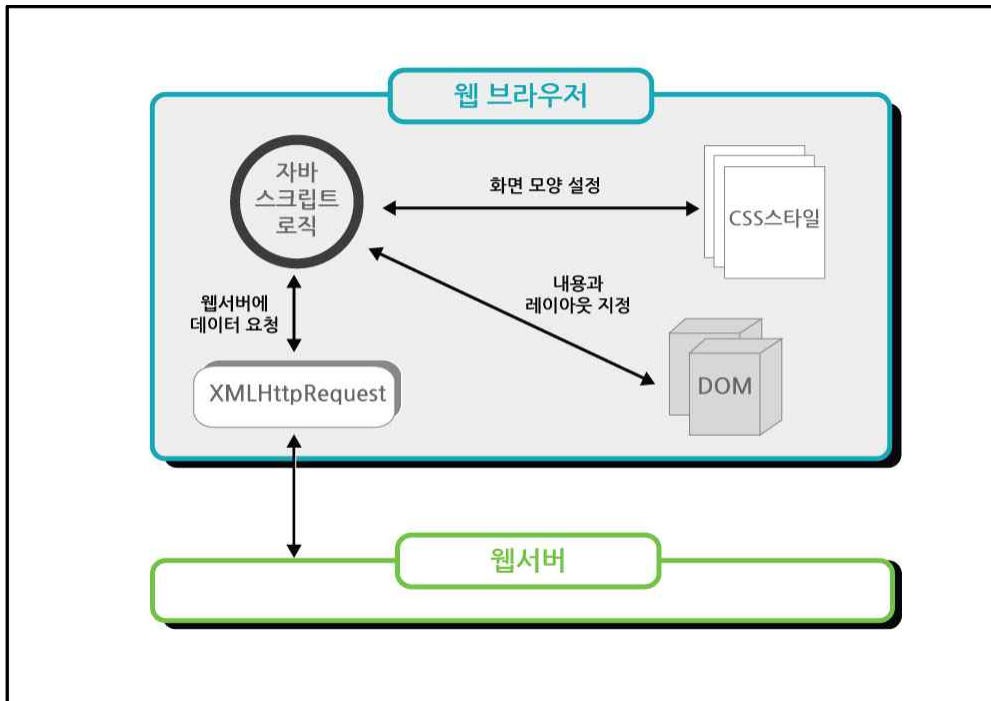
- 요청 / 응답 방식
- 요청에 대한 응답이 서버로부터 돌아오지 않으면 다른 일 수행 불가
- 응답이 돌아오면 다음 페이지로 리로딩 됨

■ Ajax를 이용한 웹 접근 방법

- 요청 / 응답 방식 동일
- 응답은 자바스크립트의 XMLHttpRequest객체가 받음
- 사용자는 하나의 페이지에서 다른일을 계속 수행할 수 있음
- 응답이 오면 자바스크립트의 함수를 통해 일을 수행
- 하나의 페이지에서 여러 가지 데이터를 서버로부터 실시간으로 갱신 가능
- 페이지 리로딩없이 계속 서버와 통신 가능

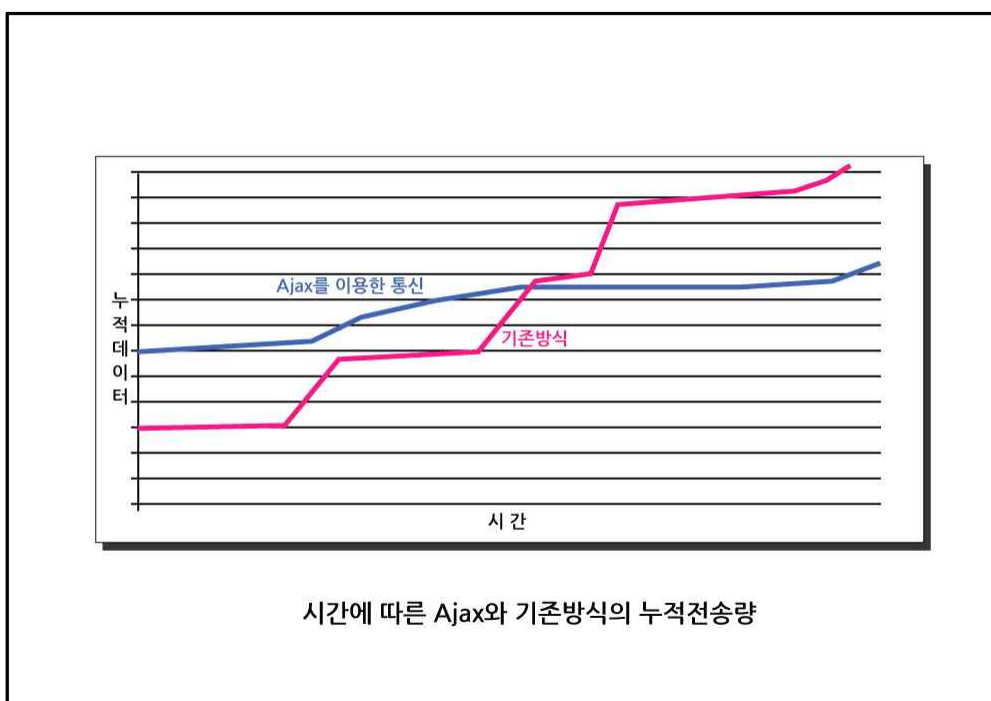


■ Ajax는 하나의 기술을 의미하는 것이 아님



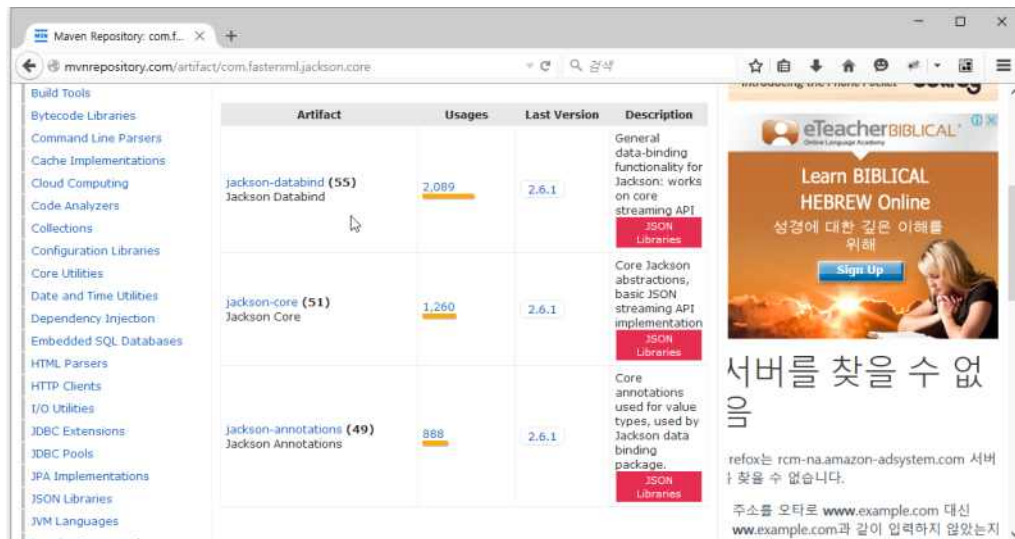
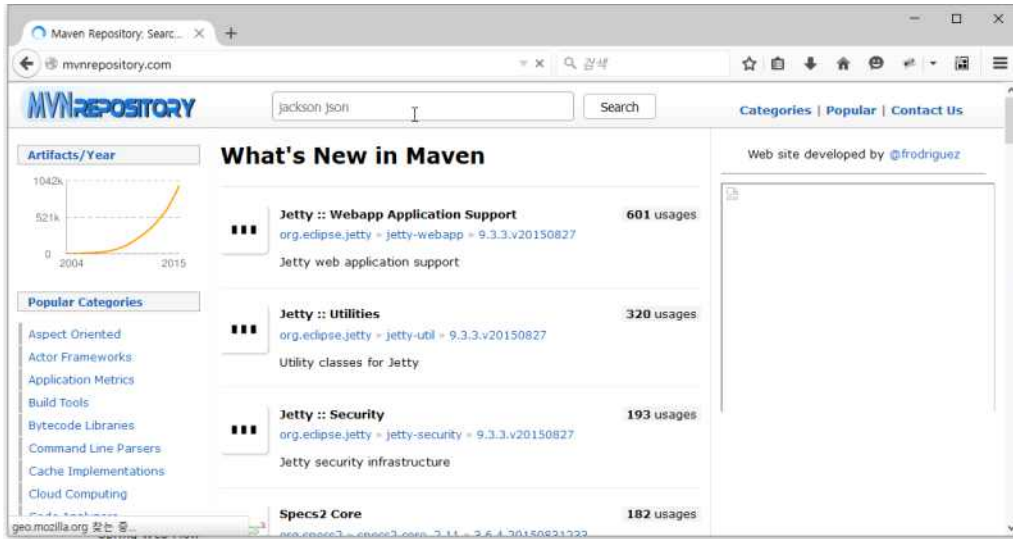
■ Ajax의 장점

- 서버 처리를 기다리지 않고, 비동기 요청이 가능
- 수신하는 데이터의 양을 줄일 수 있음
- 서버의 부하를 줄이고, 성능좋은 클라이언트를 사용함

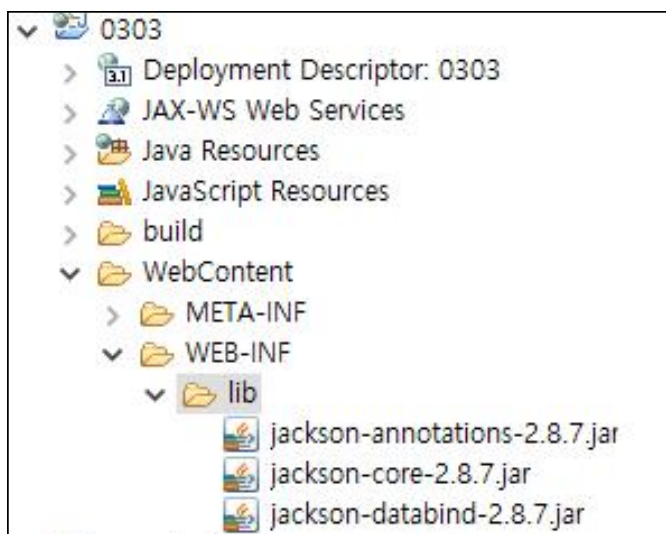


□ Jackson Json을 이용한 JSON 변환

■ <http://mvnrepository.com/> 에서 jackson json 검색



- 3개 파일 다운로드 및 lib에



■ Member(V0)클래스 생성

```
public class Member {
    private String id,password,name,phone;
    public Member() {}
    public Member(String id, String password, String name, String phone) {
        this.id = id;
        this.password = password;
        this.name = name;
        this.phone = phone;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getPhone() {
        return phone;
    }

    public void setPhone(String phone) {
        this.phone = phone;
    }
}
```

■ jackson.jsp 작성

```
<%@page import="org.codehaus.jackson.map.ObjectMapper"%>
<%@page import="java.util.ArrayList"%>
<%@page import="java.util.List"%>
<%@page import="vo.Member"%>
<% page language="java" contentType="application/json; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%

    List<Member> list = new ArrayList();
    list.add(new Member("test", "1234", "김연아", "010-222-3333"));
    list.add(new Member("test2", "2222", "류현진", "010-777-1234"));
    list.add(new Member("test3", "5555", "박지성", "010-888-2345"));
    list.add(new Member("test4", "adfb", "손흥민", "010-153-1111"));

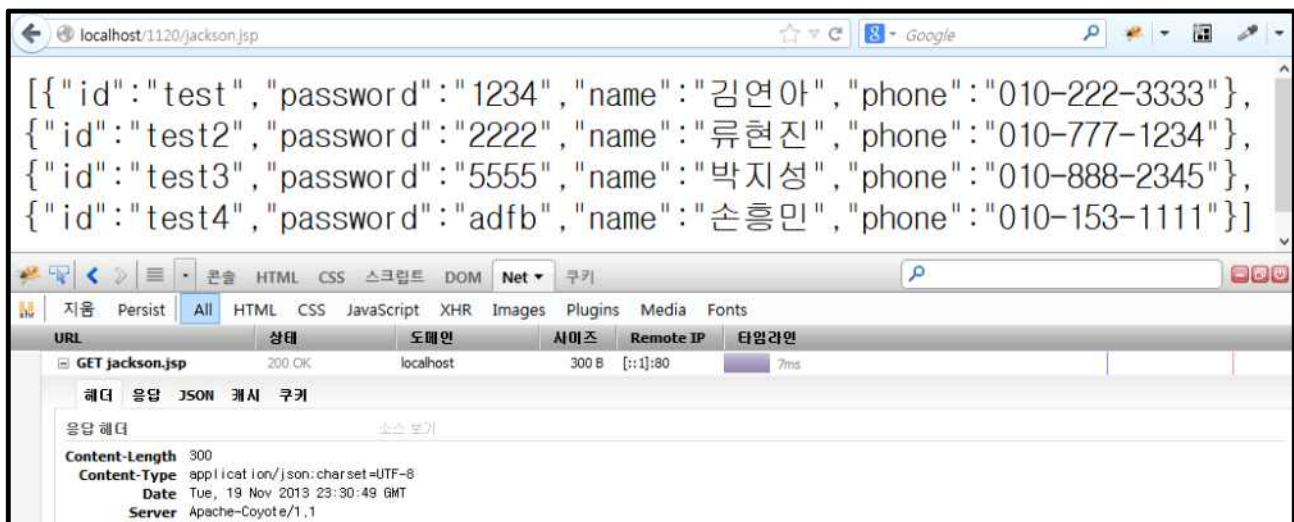
    ObjectMapper om = new ObjectMapper();

    String json = om.writeValueAsString(list);

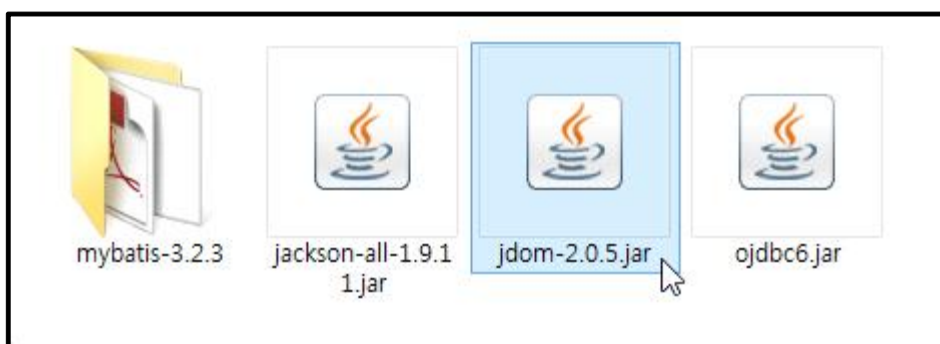
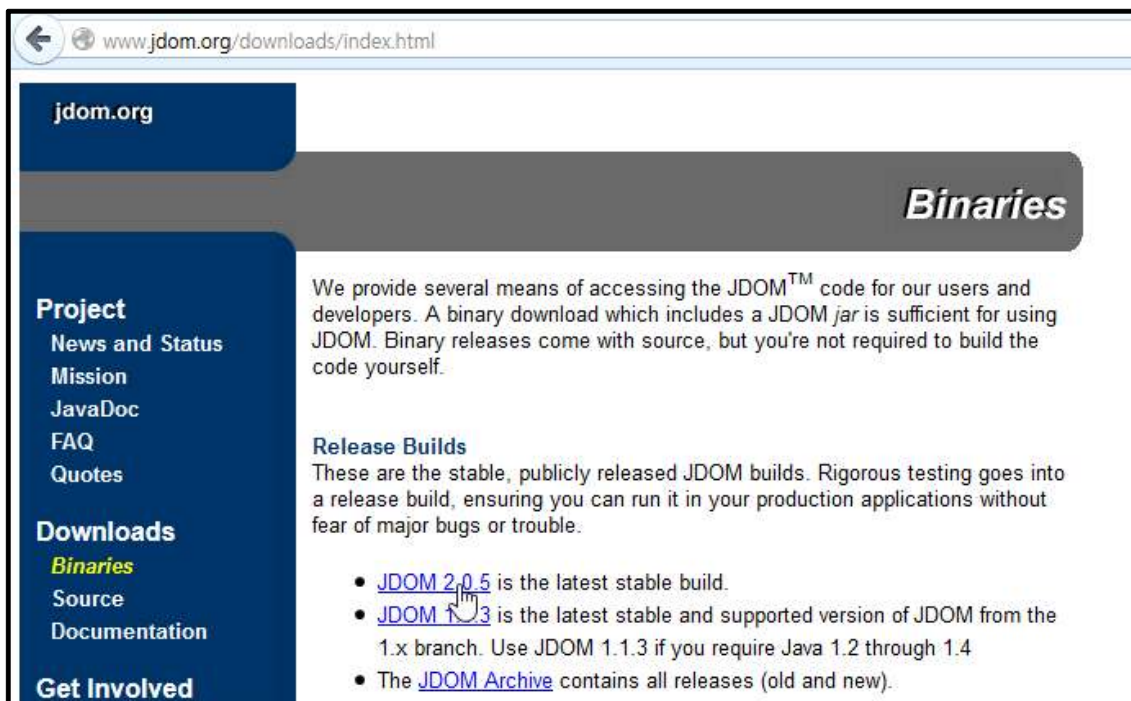
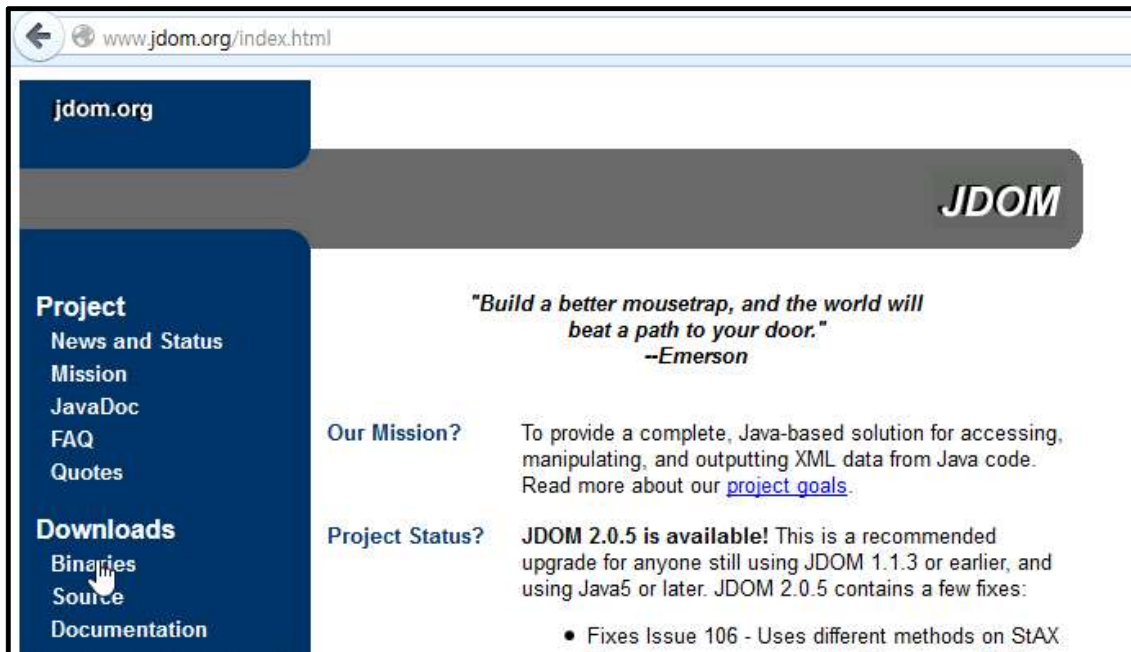
    out.write(json);

%>
```

■ 결과



■ JDOM을 이용한 xml 입 / 출력



■ xml 읽기

```
<%@page import="vo.Member"%>
<%@page import="java.util.List"%>
<%@page import="org.jdom2.Element"%>
<%@page import="org.jdom2.Document"%>
<%@page import="org.jdom2.input.SAXBuilder"%>
<%@page import="java.io.BufferedReader"%>
<%@page import="java.io.InputStreamReader"%>
<%@page import="java.io.InputStream"%>
<%@page import="java.net.URL"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%

    URL url = new URL("http://localhost/1120/data/members.xml");

    InputStream in = url.openStream();

    //읽어올때는 SAX로
    SAXBuilder sb = new SAXBuilder();

    //xml문서 생성
    Document xml = sb.build(in);

    //root Element(idols)를 얻어옵니다.
    Element members = xml.getRootElement();

    List<Element> memberList = members.getChildren();

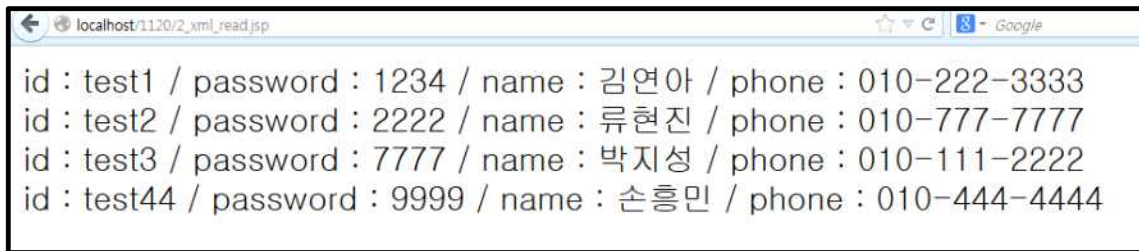
    for(Element member : memberList) {

        String id = member.getChildText("id");
        String password = member.getChildText("password");
        String name = member.getChildText("name");
        String phone = member.getChildText("phone");

        System.out.println(id);
        System.out.println(password);
        System.out.println(name);
        System.out.println(phone);

        Member mem = new Member(id,password,name,phone);
        out.print(mem+"<br/>");
    }
%>
```


■ 결과



```
id : test1 / password : 1234 / name : 김연아 / phone : 010-222-3333
id : test2 / password : 2222 / name : 류현진 / phone : 010-777-7777
id : test3 / password : 7777 / name : 박지성 / phone : 010-111-2222
id : test44 / password : 9999 / name : 손흥민 / phone : 010-444-4444
```

■ xml 쓰기

```
—<idols>
  —<idol>
    <no>11</no>
    <name>허가윤</name>
    <groupName>포미닛</groupName>
    <age>40</age>
    <agent>큐브</agent>
  </idol>
</idols>
```

```
<%@page import="org.jdom2.output.XMLOutputter"%><%@page import="org.jdom2.Element"%><%@page
import="org.jdom2.Document"%><%@ page language="java" contentType="text/xml; charset=UTF-8"
pageEncoding="UTF-8"%><%
    //문서 생성
    Document xml = new Document();

    //루트 엘리먼트 생성
    Element idols = new Element("idols");

    //허가윤
    Element idol = new Element("idol");

    //no
    Element no = new Element("no");

    no.setText("11");
```

```

//name
Element name = new Element("name");

name.setText("허가윤");

Element groupName = new Element("groupName");

groupName.setText("포미닛");

Element age = new Element("age");

age.setText("40");

Element agent = new Element("agent");

agent.setText("큐브");

idol.addContent(no);
idol.addContent(name);
idol.addContent(groupName);
idol.addContent(age);
idol.addContent(agent);

idols.addContent(idol);

//xml문서에 루트엘리먼트를 지정
xml.setRootElement(idols);

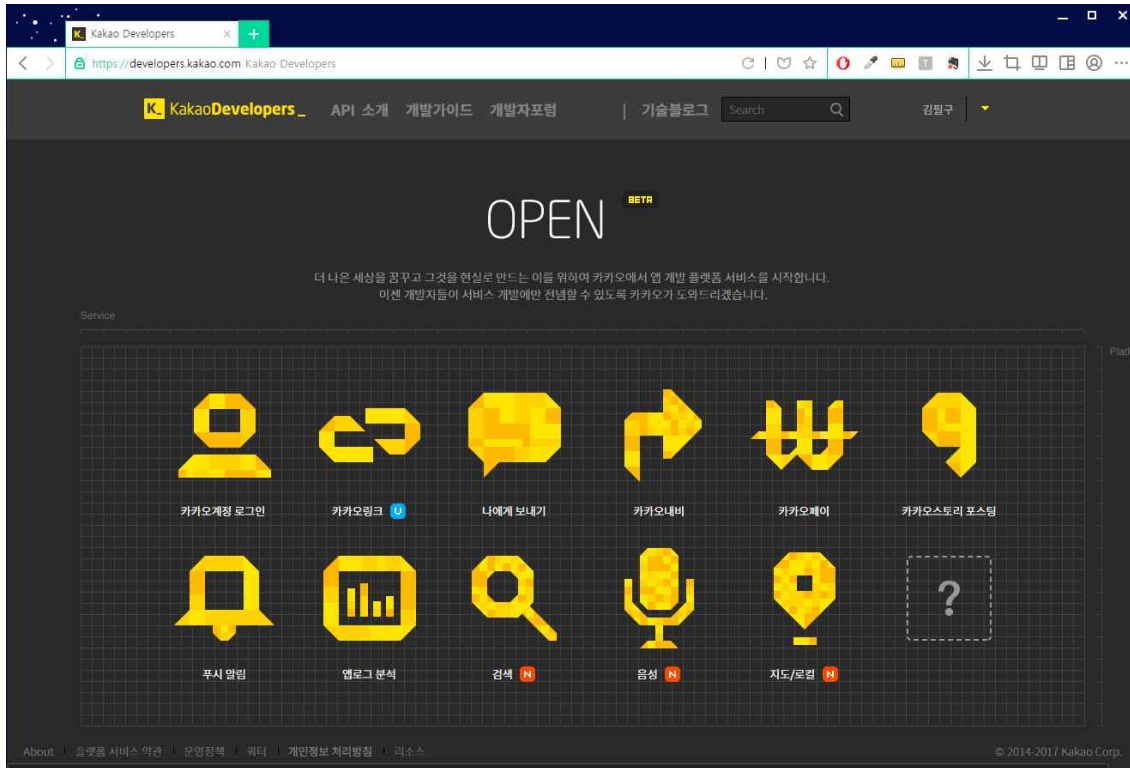
new XMLOutputter().output(xml, out);

```

%>

■ 카카오(다음)맵 API 사용

- <https://developers.kakao.com>



- 로그인 후 내 애플리케이션으로 이동



- 앱만들기



아이콘



Upload

데스크탑에서는 아이콘 이미지 파일을 끌어서 넣어도 됩니다.

이름

jbm

앱 만들기

앱 아이콘 확인

별도의 앱 아이콘을 등록하지 않을 경우 카카오톡 링크 사용 시 아래 예제와 같이 기본 앱 아이콘이 표시됩니다.

서비스 릴리즈를 계획중인 경우 서비스에 맞는 앱 아이콘을 등록하시면 유저에게 좀 더 나은 경험을 줄 수 있습니다.



앱 아이콘 수정 후 진행

계속 진행

앱 만들기

jbm

애플리케이션을 생성하였습니다

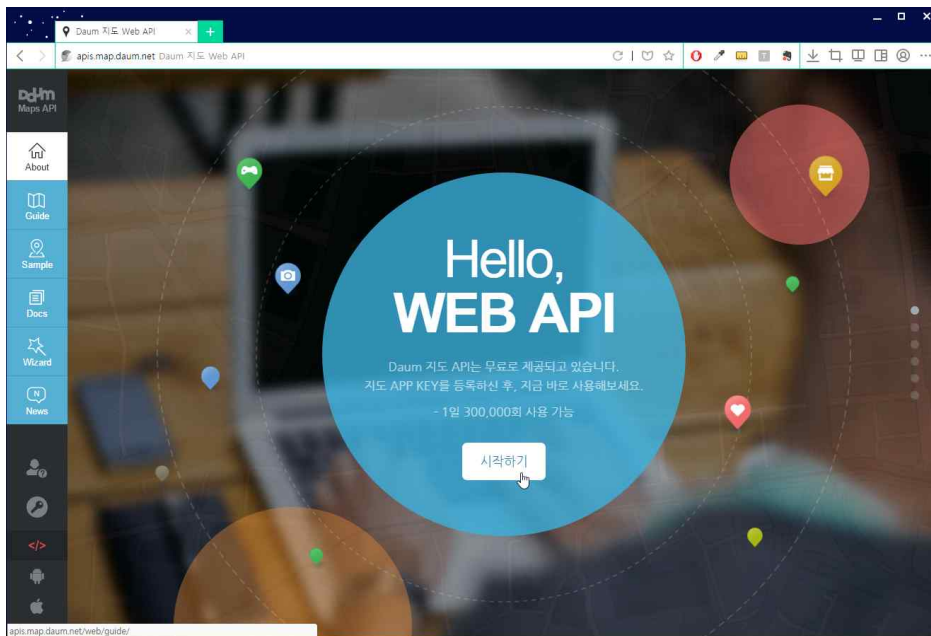
- 플랫폼 추가



- 우리는 자바스크립트키를 이용할 예정

■ 카카오(다음)맵 API의 사용

<http://apis.map.daum.net/web/>



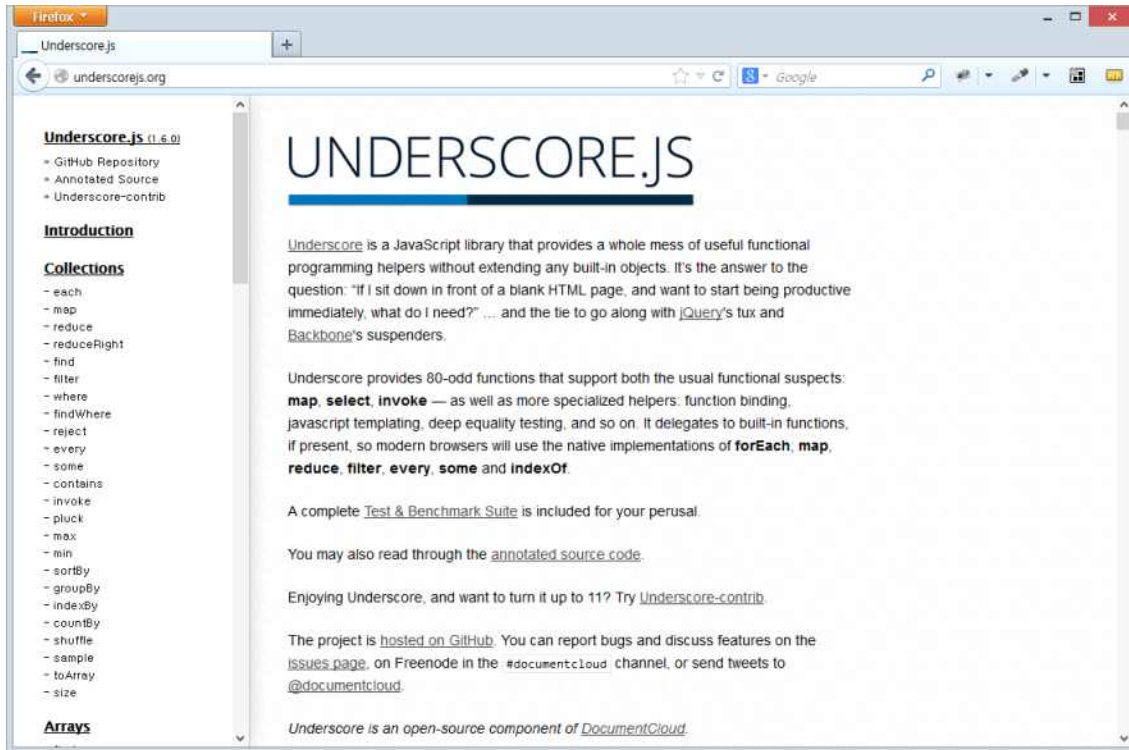
- Hello World예제로 이동

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>지도 생성하기</title>
</head>
<body>
<!-- 지도를 표시할 div 입니다 -->
<div id="map" style="width:100%;height:350px;"></div>
<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=발급받은 APP KEY를 사용하세요"></script>
<script>
var mapContainer = document.getElementById('map'), // 지도를 표시할 div
    mapOption = {
      center: new daum.maps.LatLng(33.450701, 126.570667), // 지도의 중심좌표
      level: 3 // 지도의 확대 레벨
    };

// 지도를 표시할 div와 지도 옵션으로 지도를 생성합니다
var map = new daum.maps.Map(mapContainer, mapOption);
</script>
</body>
</html>
```

- DAUM_MAPS_DEMO_APIKEY 자리에 발급받은 키 입력

- 템플릿관련 underscore.js 사용



<http://underscorejs.org/>