



군산대학교
KUNSAN NATIONAL UNIVERSITY

Open API and JSON

모바일 시스템 SW

남 광 우

Location and MAP

❖ 위치 획득 방법

✓ GPS

- Global Positioning System (GPS)
 - 미국의 위성기반 측위 시스템(러시아는 GLONAS, 유럽은 GALIEO 등등)
 - 2009년 현재 기본적으로 10~25m 정도의 오차를 가짐
 - 내비게이션은 자체 알고리즘으로 이를 보정함
- 2000년 5월 이전에는 미국의 군사적 보안을 이유로, 민간용은 일부러 100m 오차를 가지게 했었음 (Selective Availability)

✓ Cell ID : 인근 휴대폰 기지국

✓ WIFI

✓ IP Addresses

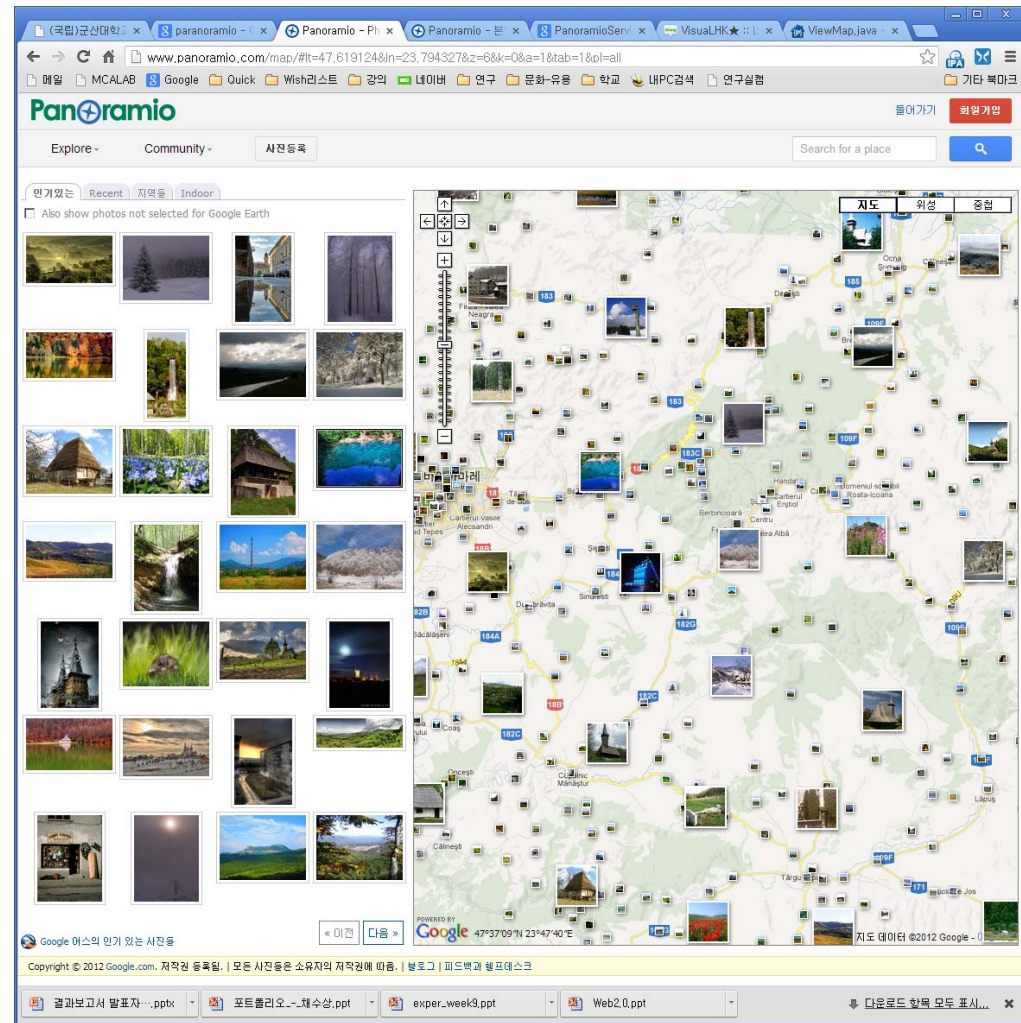
Panoramio API

❖ Panoramio

✓ Google Photo System

<http://www.panoramio.com>

<http://www.panoramio.com/map/#lt=47.619124&ln=23.794327&z=6&k=0&a=1&tab=1&pl=all>



Panoramio API

❖ Data API

- ✓ <http://www.panoramio.com/api/data/api.html>
- ✓ REST API GET
 - http://www.panoramio.com/map/get_panoramas.php?set=public&from=0&to=20&minx=-180&miny=-90&maxx=180&maxy=90&size=medium&mapfilter=true



결과는 JSON

Panoramio API

❖ Panoramio Result

```
{
  "count": 773840, "photos": [
    {
      "photo_id": 532693,
      "photo_title": "Wheatfield in afternoon light",
      "photo_url": "http://www.panoramio.com/photo/532693",
      "photo_file_url": "http://static2.bareka.com/photos/medium/532693.jpg",
      "longitude": 11.280727,
      "latitude": 59.643198,
      "width": 500,
      "height": 333,
      "upload_date": "22 January 2007",
      "owner_id": 39160,
      "owner_name": "Snemann",
      "owner_url": "http://www.panoramio.com/user/39160",
    },
    {
      "photo_id": 505229,
      "photo_title": "Etangs près de Dijon",
      "photo_url": "http://www.panoramio.com/photo/505229",
      "photo_file_url": "http://static2.bareka.com/photos/medium/505229.jpg",
      "longitude": 5.168552,
      "latitude": 47.312642,
      "width": 350,
      "height": 500,
      "upload_date": "20 January 2007",
      "owner_id": 78506,
      "owner_name": "Philippe Stoop",
      "owner_url": "http://www.panoramio.com/user/78506"
    }, ...
  ]
}
```

JSON

❖ JSON이란?

- ✓ JSON : JavaScript Object Notation
- ✓ <http://www.json.org>

❖ JSON의 정의

- ✓ 경량의 데이터 교환 포맷
- ✓ 인간이 읽고 쓰기가 용이하고, 기계에 있어서도 간단하게 파싱이나 생성을 행할 수 있는 형식
- ✓ XML 대비 장점

❖ 형식

- ✓ Key/Value 형식

JSON

❖ 데이터의 형식

데이터 타입	표현 방법	예
number	Integer 또는 float	"number" : 1
string	큰 따옴표로 묶음	"name" : "황상규"
boolean	true 또는 false	"isResult" : true
object	여러 개의 key/value를 입력하여 중괄호로 묶음	{"name" : "황상규", "gender" : "남자"}
array	여러 개의 object를 대괄호로 묶음	{ "employees" : [{"name" : "황상규", "gender" : "남자"}, {"name" : "싸이", "gender" : "남자"}, {"name" : "김태희", "gender" : "여자"}] }

<http://urajilator.tistory.com/1105>

JSON

❖ JSON과 XML 포맷의 비교

```
1: {  
2:   "book": {  
3:     "type": "textbook",  
4:     "pages": "256",  
5:     "title": "Programming Pearls 2nd Edition",  
6:     "description": "The first edition of Programming",  
7:     "rating": "4.5",  
8:     "coverType": "paperback",  
9:     "genre": "Computer Science",  
10:    "author": "Jon Bentley",  
11:    "publisher": "Addison-Wesley Professional",  
12:    "copyright": "1999"  
13:  }  
14: }
```

```
1: <book>  
2:   <type>textbook</type>  
3:   <pages>256</pages>  
4:   <title>Programming Pearls 2nd Edition</title>  
5:   <description>The first edition of Programming</description>  
6:   <rating>4.5</rating>  
7:   <coverType>paperback</coverType>  
8:   <genre>Computer Science</genre>  
9:   <author>Jon Bentley</author>  
10:  <publisher>Addison-Wesley Professional</publisher>  
11:  <copyright>1999</copyright>  
12: </book>
```


JSON - Java

❖ JSON 라이브러리

- ✓ - Json-lib (<http://json-lib.sourceforge.net/>)
- ✓ - org.json.simple(<http://www.json.org/java/simple.txt>,
 - 다운로드: http://www.JSON.org/java/json_simple.zip

❖ JSON Java 라이브러리

- ✓ JSONObject
- ✓ JSONArray

JSON - Java

❖ JSONObject의 생성

```
JSONObject result = new JSONObject(map);  
JSONObject result = new JSONObject(string);
```

❖ http response로 부터 생성 예

```
URL url = new URL(urlString);  
InputStream inStream = url.openStream();  
String str = streamToString(inStream); // 직접 구현  
JSONObject json = new JSONObject(str);
```

JSON - Java

❖ JSON 생성

```
// JSON 생성
Map jsonMap = new HashMap();
jsonMap.put("name", "황상규");

JSONObject jo = new JSONObject();
jo.put("gender", "남자");
jo.put("person", jsonMap);

// JSON 파싱
jo.toString();    // {"gender":"남자", "person":{"name":"황상규"}}
jo.get("gender"); // "남자"
```



```
{
  "gender":"남자",
  "person":
    {
      "name":"황상규"
    }
}
```

JSON - Java

❖ JSONObject 에서 데이터 읽기

```
{
  "count": 773840, "photos": [
    {
      "photo_id": 532693,
      "photo_title": "Wheatfield in afternoon light",
      "photo_url": "http://www.panoramio.com/photo/532693",
      "photo_file_url": "http://static2.bareka.com/photos/medium/532693.jpg"
      "longitude": 11.280727,
      "latitude": 59.643198,
      "width": 500,
      "height": 333,
      "upload_date": "22 January 2007",
      "owner_id": 39160,
      "owner_name": "Snemann",
      "owner_url": "http://www.panoramio.com/user/39160",
    },
  ],
}
```

```
protected void parseSearchResult(JSONObject json, PanoramioList list) {
    try {
        list.count_ = (int) json.getLong("count");
        list.hasMore_ = json.getBoolean("has_more");

        JSONArray jphotos = json.getJSONArray("photos");
        for (int i = 0; i < jphotos.length(); i++) {
            JSONObject jphoto = jphotos.getJSONObject(i);
            Photo photo = new Photo();

            photo.id_ = String.valueOf(jphoto.getLong("photo_id"));
            photo.title_ = jphoto.getString("photo_title");

            photo.photoUrl_ = jphoto.getString("photo_url");

            photo.longitude_ = jphoto.getDouble("longitude");
            photo.latitude_ = jphoto.getDouble("latitude");

            photo.width_ = (int) jphoto.getLong("width");
            photo.height_ = (int) jphoto.getLong("height");

            photo.owner_ = jphoto.getString("owner_name");
            photo.ownerId_ = String.valueOf(jphoto.getLong("owner_id"));
            photo.ownerUrl_ = jphoto.getString("owner_url");

            // photo.uploadDate_ = new Date();

            photo.setFileUrls();

            list.list_.add(photo);
        }
    } catch (Exception e) {

    }
}
```

Panoramio API 활용하기

❖ Panoramio API 데모 소스

- ✓ <http://www.java2s.com/Open-Source/Android/UnTagged/photo-around/com/tagearth/panoramio/PanoramioServices.java.htm>