

# Mashups

Pro Web 2.0

## 프로 웹 2.0 매시업

레이몬드 이 지음 / 박천구 옮김 / 유준석 · 박천구 특별부록



Apress®



위키 북스



프로  
웹 2.0  
매쉬업

## • 목 차 •

# 1부 프로그래밍하지 않고 정보를 조합하기

1장	특정 매쉬업으로부터 배우기	3
	매쉬업에서 패턴 찾아내기 .....	4
	하우징 맵.....	6
	무엇이 조합되는가?.....	6
	왜 구성요소들이 조합되는가? 풀어야 할 문제들은 어떤 것들이 있는가?..	7
	리믹스가 어디에서 이루어지는가?.....	7
	이러한 요소들이 어떻게 조합되는가? .....	7
	비교할 만한 매쉬업들 .....	9
	플리커 내의 구글 맵.....	9
	무엇이 조합되는가?.....	10
	왜 구성요소들이 조합되는가? 해결해야 할 문제는 어떤 것이 있는가?...	10
	어떻게 이런 요소들이 조합되는가? .....	15
	비교할 만한 매쉬업들 .....	16
	라이브러리룩업 북마크릿 .....	16
	라이브러리룩업 북마크릿 설정하기 .....	18
	라이브러리룩업 북마크릿 실행시키기 .....	19
	어떻게 이 매쉬업이 동작하는가? .....	20
	이 매쉬업을 어떻게 확장할 수 있는가? .....	20
	비교할 만한 매쉬업들 .....	21
	다른 매쉬업들 찾아보기 .....	22
	요약.....	22

## 2장      웹사이트 매쉬업 가능성 살펴보기      25

어떤 요소가 웹 사이트와 애플리케이션을 매쉬업 가능하도록 만드는가.	26
웹사이트의 주요 요소를 확인하기 .....	27
공개 API와 기존의 매쉬업 .....	28
Ajax 사용 .....	29
스크립트를 임베딩할 수 있는가 .....	30
브라우저 플러그인 .....	30
웹사이트에서 데이터를 가져오거나 외부로 내보내기 .....	31
사용자와 개발자 커뮤니티 .....	31
모바일과 대안 인터페이스 그리고 웹사이트 스킨의 변경이 자유로운가	32
문서화 .....	32
웹사이트가 오픈소스 정책을 지원하는가 .....	33
지적재산권, 재사용성, 그리고 크리에이티브 커먼즈 .....	33
태깅, 피드, 그리고 웹로깅(블로깅) .....	33
웹사이트의 URL 언어 .....	34
몇몇 매쉬업에 대해 간략하게 다시 보자 .....	35
본질적으로 매쉬업이 용이한 플리커 .....	36
플리커의 리소스 .....	37
사용자와 사진 .....	38
개별 사진과 연관된 데이터 .....	42
태그 .....	43
사용자의 저장소에서 사진을 일자별로 브라우징하기 .....	45
세트 .....	46
컬렉션 .....	47
좋아하는 사진 .....	47
사용자의 인기사진 .....	48
연락처 목록 .....	48
그룹 .....	49
계정관리 .....	50
플리커 브라우징하기 .....	51
검색 .....	52
위치정보가 태깅된 플리커 사진 .....	54
플리커의 사진, 비디오 관리 .....	56

최근 활동 .....	56
메일링 인터페이스 .....	56
블로그와 인터페이스하기 .....	57
RSS와 Atom 피드의 배포 .....	57
무선을 통한 접근 .....	57
서드파티 플리커 애플리케이션 .....	58
크리에이티브 커먼즈 라이선스의 적용 .....	58
카메라 .....	59
URL 템플릿과 임베딩 패턴에 의한 매쉬업 .....	60
구글 맵 .....	63
구글 맵의 URL 언어 .....	63
구글 맵의 KML 파일 보기 .....	65
야후!파이프와 구글 맵의 연결 .....	66
구글 맵의 URL 언어를 이용한 기타 애플리케이션들 .....	66
아마존 .....	68
아마존 아이템 .....	68
리스트 .....	70
태그 .....	71
주제별 분류 .....	71
딜리셔스 .....	73
스크린-스크레이핑과 봇 .....	74
요약 .....	77

### 3장      태깅과 포크소너미에 대한 이해      79

플리커에서의 태깅 .....	80
플리커의 태그 .....	81
태그가 실제로 어떻게 사용되는가 .....	83
태그 생성하기 .....	84
플리커에서의 태그 문법 .....	84
태그의 잠재적인 약점 .....	84
플리커에서 단수형과 복수형의 태그 .....	85
태깅 시스템의 해킹 : 위치정보 태깅과 머신 태그 .....	86

플리커 태그를 사용하는 흥미로운 애플리케이션들 .....	88
딜리셔스에서 태깅 .....	88
딜리셔스에서 태그를 추가하는 기법 .....	89
대소문자 구분과 여러 단어로 이루어진 문장의 처리 .....	90
보다 많은 정보 얻기 .....	91
테크노라티 태그를 통해 콘텐츠 수집하기 .....	93
태그를 사용하여 테크노라티에서 검색하기 .....	93
테크노라티는 어떻게 웹에서 태그를 찾아낼까 .....	94
테크노라티 태그에 있어 단어의 어형변화와 언어상의 제약 .....	94
태그를 사용하여 플리커와 딜리셔스를 매쉬업하기 .....	95
태깅을 사용하는 기타 시스템 .....	96
정규적인 분류 체계와 태그의 관계 .....	96
요약 .....	99

## 4장 피드, RSS, 그리고 Atom으로 작업하기 101

피드가 무엇이며 이것이 왜 중요한가? .....	102
RSS2.0 .....	103
RSS1.0 .....	104
Atom1.0 .....	107
RSS2.0과 Atom1.0에 대한 확장기능 .....	109
플리커의 피드 .....	111
플리커 피드의 파라미터 .....	112
플리커 피드에 대한 검토 .....	113
RSS와 Atom 외의 데이터 교환 포맷 .....	117
다른 웹사이트의 피드 .....	119
피드를 찾아내고 자동발견하기 .....	120
블로그로부터의 피드 .....	122
위키피디아 피드 .....	122
구글과 야후! 뉴스 .....	123
뉴스 수집기: 플리커 피드를 다른 곳에서 보여주기 .....	125
피드의 검증 .....	126

GUI 툴을 사용하여 피드를 스크레이핑하기 .....	127
피드를 피드버너와 리믹스하기 .....	128
야후!파이프와 피드를 리믹스하기 .....	129
야후!뉴스를 사용하는 첫 번째 파이프.....	131
구글 뉴스와 파이프를 리팩토링하기 .....	132
위키뉴스와 뉴욕 타임즈: 피드 필터링하기 .....	133
피드들을 함께 가져오기 .....	134
요약.....	135

## 5장      블로그와 통합하기 137

블로그 통합 시나리오 .....	138
플리커의 사진을 블로그로 전송하기 .....	139
블로그와 통합되도록 플리커를 설정하기 .....	140
플리커의 사진을 블로그하기 .....	144
어떻게 플리커와 블로그의 통합이 이루어지는가? .....	145
데스크톱 기반 블로깅 툴 .....	146
피드와 블로깅의 조합을 통한 리블로깅 .....	148
플락 : 블로그와 플리커를 함께 사용하기 .....	149
RSD: 블로그 API의 인지성(discoverability) .....	150
링크백.....	152
초기단계의 위키 통합.....	153
요약.....	154



## 2부 API를 사용하여 단일 웹 애플리케이션을 리믹스하기

### 6장 플리커를 통해 웹서비스 API에 대해 배워보자159

플리커 API 소개.....	160
이 XML 응답은 무엇을 의미하는가?.....	163
XML 응답을 가지고 무엇을 할 수 있는가?.....	166
API 문서, 커뮤니티, 그리고 정책.....	168
API의 사용약관.....	169
플리커 API 탐색기와 문서 사용하기.....	170
PHP에서 기본적인 플리커 API 메서드 호출하기.....	174
HTTP 클라이언트.....	175
HTTP에 대해 좀 더 자세히 알아보자.....	176
XML 처리.....	182
사진을 표시하는 간단한 HTML을 생성하기.....	187
이 코드가 의미하는 것은 무엇인가?.....	189
플리커 API에 대해 전반적으로 알아보자.....	189
flickr.reflection 메서드 사용하기.....	190
PHP에서 플리커의 리플렉션 메서드 사용하기.....	193
요청과 응답의 포맷.....	199
플리커의 권한부여.....	202
왜 비밀번호를 전송하는 방식이 효과적이지 않은가.....	203
웹 애플리케이션에서의 권한부여.....	203
플리커 API 키트 사용하기.....	213
PEAR::Flickr_API.....	214
phpFlickr.....	215
Phlickr.....	217
플리커 API의 한계.....	218
요약.....	220

## 7장      다른 웹 API에 대해 알아보자      221

XML-RPC.....	223
네트워크상에서는 어떤 일이 벌어지는가? .....	228
HTTP 메시지의 분석과 구성에 와이어샤크와 킬 사용하기.....	228
XML-RPC 트래픽을 파싱하기.....	230
SOAP.....	233
WSDL과 SOAP을 사용하면 웹서비스 호출이 쉬워진다는 것은 희망사항이다.....	234
geocoder.us .....	235
아마존 ECS.....	245
SOAP을 통해서 플리커 API를 사용하기 .....	250
특정 웹 API에 대해서 알아보자.....	250
Programmableweb.com .....	251
유튜브.....	254
GData와 블로거 API.....	255
블로거 API를 HTTP 메서드에 기반한 표준 인터페이스로 사용하기 ...	261
요약.....	261

## 8장      Ajax/자바스크립트 위젯과 위젯 API에 대해 배워 보자      263

어떤 것을 알아야 하는가.....	265
Ajax는 어떤 차이점을 만들어 내는가?.....	265
파이어버그, DOM 인스펙터, 그리고 자바스크립트 셸에 대해서 알아보자.....	267
DOM 인스펙터 사용하기 .....	267
파이어폭스에서 파이어버그 확장모듈 사용하기 .....	267
자바스크립트 셸 사용하기.....	269
자바스크립트 라이브러리 사용하기 .....	270
YUI 위젯.....	271
YUI 캘린더 사용하기.....	271
YUI를 여러분의 서버호스트에 설치하기 .....	273
구글 맵에 대해 알아보자.....	273

자바스크립트에서 플리커에 접근하기 .....	278
그리즈명키를 사용하여 뉴욕 타임즈가 제공하는 퍼머링크에 접근하기 .....	282
자바스크립트와 Ajax에 대해 좀 더 알아보기 .....	286
요약.....	286

## 3부 매쉬업 작성하기

### 9장 API와 리믹스에 사용되는 요소에 대한 내용에서 매쉬업 관련 내용까지 알아보기 291

ProgrammableWeb에 대한 소개.....	292
ProgrammableWeb에서 사용자가 생성한 데이터.....	293
매쉬업 관련 디렉터리 서비스가 유지될 수 있을까? .....	293
전반적인 매쉬업 현황에 대해 알아보기 .....	294
매쉬업 디렉터리 .....	295
매쉬업을 추적하기 위해 피드 사용하기 .....	296
매쉬업을 설명하기 위한 태그의 사용.....	296
API와 매쉬업 버티컬 .....	299
특정 매쉬업 프로필에 대해 알아보기 .....	300
특정 API를 사용하는 매쉬업 찾아내기 .....	301
매쉬업을 사용하여 해결할 수 있는 문제 .....	301
관심있는 책 추적하기 .....	302
언제 항공기 티켓을 구매해야 하는지 알아보기 .....	307
꿈에 그리던 집 찾기 .....	309
최신뉴스 맵핑하기 .....	310
요약.....	312

### 10장 다양한 서비스로부터 매쉬업 생성하기 313

설계.....	314
배경지식: 플리커에서의 위치정보 태깅 .....	316
배경지식: XMLHttpRequest와 라이브러리 포함하기 .....	320

XMLHttpRequest를 직접 사용하기 .....	321
야후 커넥션 매니저 사용하기 .....	323
서버 측 프록시 작성하기 .....	326
XHR을 사용하여 직접 API를 호출하면 어떤 일이 발생할까? .....	326
위치정보가 포함된 사진을 가져오는 서버 측 스크립트 작성하기 .....	328
간단한 클라이언트 측 프레임워크 생성하기 .....	331
엘리먼트를 읽고 작성하기 .....	331
간단한 이벤트를 입력 폼과 연결하고 폼 전송의 결과를 화면에 표시하기 .....	334
클라이언트 측 프레임워크를 플리커에 연결하기 .....	335
flickrgeo.php에 쿼리하는 URL 작성하기 .....	336
JSON 형태의 데이터를 읽기 위해 YUI 커넥션 매니저를 통해 XHR 사 용하기 .....	337
JSON을 HTML로 변환하기 .....	339
플리커와 구글 맵 API를 매쉬업하기 .....	341
기본 구글 맵 설정하기 .....	342
표시되는 영역이 변경될 때 지도가 반응하도록 설정하기 .....	343
플리커와 구글 맵의 코드 합치기 .....	345
구글 맵의 지도 표시 영역을 연결하기 .....	345
사진들을 지도상에 표시하기 .....	348
플리커 사진을 표시해주는 구글 맵플릿 .....	352
요약 .....	357

## 11장 툴을 사용하여 매쉬업 생성하기 359

매쉬업 툴이 해결해주는 문제 .....	360
이번 장에서 개발할 것은 무엇인가 .....	361
매쉬업의 생성 : 단계별로 예제를 개발해보자 .....	362
구글 매쉬업 에디터에 익숙해지기 .....	363
피드를 읽고 화면에 표시하기(심플 템플릿) .....	364
사용자 정의 템플릿 사용하기 .....	366
플리커 접근 시 야후! 파이프 사용하기 .....	368
<gm:map>를 사용하여 플리커 사진을 화면에 표시하기 .....	370
매쉬업에 자바스크립트 추가하기 .....	373

어떻게 피드를 영속화하고 탭을 사용할 것인가.....	378
저장된 항목을 지도상에 표시하는 최종 애플리케이션.....	383
구글 매쉬업 에디터와 야후! 파이프의 사용에 따르는 트레이드-오프 분석	388
기타 매쉬업 툴.....	390
요약.....	392

## 12장 사이트를 매쉬업 가능하도록 만들기 393

왜 웹사이트를 매쉬업 가능하도록 만들어야 하는가.....	394
API에 의존하지 않는 기술 사용하기 .....	395
일관성 있고 풍부한 URL 언어를 사용하라 .....	395
웹사이트 개발에 W3C 표준을 사용하라 .....	396
웹 접근성에 주의를 기울여라.....	396
사용자들이 콘텐츠에 태그를 추가할 수 있도록 고려하라 .....	397
피드를 제공하라 .....	397
여러분의 콘텐츠를 블로그와 다른 웹사이트에 쉽게 게시할 수 있도록 하라 .....	398
명확한 라이선스를 표시하여 콘텐츠의 공유를 장려하라 .....	399
사용자 콘텐츠에 대해 다양한 불러오기, 내보내기 옵션을 개발하라	400
사용자들이 여러분의 콘텐츠를 어떻게 리믹스하고 리믹스하기 쉽게 만드는지 연구하라 .....	400
매쉬업이 용이한 API 만들기 .....	400
다른 API로부터 배우고 모방하라.....	400
API의 사용자를 잊지 말라 .....	401
배우기 쉬운 API를 개발하라.....	401
API의 사용성을 테스트하라.....	402
느슨하게 결합된 아키텍처를 구성하여 API의 생성이 다른 사용자에게 도움이 되는 만큼 여러분에게도 도움이 되도록 하라.....	403
REST 방식을 채택하고 가능하다면 SOAP와 XML-RPC도 지원하라...404	
REST의 특정 예로서 Atom 출판 프로토콜(Atom Publishing Protocol)의 사용을 고려해보라 .....	404
서드파티 개발자나 내부 개발자의 API 키트 개발을 장려하라 .....	404
API에서 광범위한 에러 리포팅을 지원하라.....	405
출력과 입력을 위해 여러 가지 입출력 형식을 지원하라 .....	406

UI에서 제공되는 기능을 API에서도 지원하라 .....	406
여러분의 사이트를 위한 검색기능을 포함시켜라 .....	406
API의 버전을 관리하라 .....	406
개발자 커뮤니티를 육성하라 .....	407
API를 너무 통제하려 들지 마라 .....	407
서비스 레벨 보증서의 생성을 고려하라 .....	407
API 사용자가 여러분의 리소스를 현명하게 사용하도록 도와라 .....	408
여러분의 애플리케이션을 오픈소스화하는 것에 대해 고민하라 .....	408
쉽게 이해할 수 있는 데이터 표준 .....	409
요약 .....	410

## 4부 다른 매쉬업 관련 주제에 대해 알아보자

### 13장 온라인 지도와 3D 디지털 세계를 리믹스하기 415

온라인 지도 서비스의 범람 .....	417
지도 기반 매쉬업 예제들 .....	418
프로그래밍 없이 지도를 생성하기 .....	418
Mapbuilder.net .....	419
구글 My Maps .....	421
매쉬업 기회 : 야후! 로컬 컬렉션을 지도 서비스와 매쉬업하기 .....	422
야후! 로컬의 XML 파일을 Mapbuilder.net에서 사용 가능한 CSV 포맷 으로 변환하기 .....	425
마이크로소프트 라이브 서치 맵에서의 컬렉션 생성 .....	427
프로그래밍하지 않고 지도를 생성하는 방법 .....	429
데이터 교환 포맷 .....	429
CSV .....	429
HTML에서 사용되는 마이크로포맷과 메타태그 .....	430
GeoRSS .....	431
야후!의 GeoRSS 사용과 YMaps 확장기능 .....	433
KML .....	438
GeoRSS와 KML과 같은 포맷들 사이의 호환성 .....	439

API 프로그래밍으로 지도 생성하기 .....	439
구글 맵 API .....	440
야후! 맵 API .....	445
마이크로소프트의 라이브 서치 맵/버추얼 어스 .....	448
주소와 좌표 간의 변환 .....	451
야후! 맵 .....	451
Geocoder.us .....	452
구글 Geocoder .....	454
버추얼 어스 .....	457
미국 내 주소가 아닌 경우의 주소-좌표변환 .....	459
구글 어스와 KML .....	460
최종 사용자로서 KML을 화면에 표시하고 다루기 .....	461
KML .....	466
COM과 애플스크립트를 사용하여 구글 어스 프로그래밍하기 .....	472
맵스트랙션과 오픈레이어즈 .....	475
통합 예제: 플리커 사진을 구글 어스에서 보여주기 .....	476
KML 네트워킹크 .....	479
사진을 표시하는 KML 생성하기 .....	483
flickrgeo.php 코드 .....	484
요약 .....	494

## 14장 소셜 북마킹과 도서목록 시스템에 대한 탐구 497

소셜 북마킹 사이트의 시장현황 .....	499
Programmableweb.com을 사용하여 API의 인기를 알아보기 .....	499
딜리셔스 .....	500
딜리셔스 API 사용하기 .....	501
딜리셔스를 위한 서드파티 툴 .....	510
서드파티 API 키트 .....	510
야후! 북마크와 마이웹 .....	513
캐노티 .....	514
플리커와 딜리셔스의 매쉬업 .....	519
요약 .....	523

## 15장 온라인 캘린더와 이벤트 수집기 525

구글 캘린더.....	526
최종 사용자로서 구글 캘린더 설정하기.....	527
구글 캘린더가 제공하는 피드 포맷에 대해 알아보자.....	530
GData 기반의 캘린더 API를 직접 사용하기.....	536
구글 캘린더용 PHP API 키트 사용하기.....	547
구글 캘린더용 파이썬 API 키트 사용하기.....	550
30boxes.com.....	551
최종 사용자 튜토리얼.....	552
30boxes.com의 API.....	552
이벤트 수집기.....	556
Upcoming.yahoo.com.....	556
Eventful.com.....	568
iCalendar로 프로그래밍하기.....	575
파이썬과 iCalendar.....	575
PHP와 iCalendar.....	577
이벤트 캘린더를 iCalendar와 구글 캘린더로 내보내기.....	579
소스: UC 버클리 이벤트 캘린더.....	579
파이썬을 사용하여 '비평가의 선택'에서 iCalendar 피드 생성하기.....	580
구글 캘린더에 이벤트 추가하기.....	582
요약.....	590

## 16장 온라인 스토리지 서비스 사용하기 591

아마존 S3에 대한 소개.....	592
S3를 사용하는 근본적인 이유.....	592
아마존 S3의 개념적인 구조.....	593
파이어폭스 S3 확장기능을 사용하여 S3 시작하기 .....	595
S3의 REST 인터페이스 사용하기 .....	597
REST 인터페이스를 사용하여 버킷 목록 가져오기 .....	600
S3의 SOAP 인터페이스 사용하기 .....	601



아마존의 S3 API 키트.....	603
PHP .....	603
파이썬.....	605
요약.....	606

## 17장   데스크톱과 웹 기반 오피스 스위트 매쉬업하기 607

오피스 스위트의 매쉬업 시나리오 .....	608
문서 마크업의 세계 .....	609
OpenDocument 포맷 .....	610
기본적인 ODF 태그에 대해 배워보자 .....	620
ODF 요소에 어떠한 서식도 적용하지 않은 ODF 텍스트 문서 생성하기 .....	622
문단의 텍스트를 text-body로 설정하기 .....	627
정렬된 목록과 정렬되지 않은 목록을 구분하기 위해 포매팅하기 .....	628
<text:span> 엘리먼트에 볼드체, 이탤릭체, 폰트 변경, 그리고 색상 변경 적용하기 .....	630
ODF를 다루는 API 키트 .....	631
Odfpy .....	632
OpenDocumentPHP .....	641
오픈오피스를 사용하여 ODF 생성하기 .....	644
ECMA 표준인 Office Open XML(OOXML) .....	645
OOXML 뷰어와 검증기 .....	649
ODF와 OOXML의 비교 .....	649
온라인 오피스 스위트 .....	650
프로그래밍 가능한 온라인 스프레드시트의 사용 시나리오 .....	650
구글 스프레드시트 API .....	651
파이썬 API 키트 .....	651
아마존 위시리스트와 구글 스프레드시트의 매쉬업 .....	656
구글 스프레드시트용 젠드 PHP API 키트 .....	662
마지막 변경: COM을 이용하여 아마존 위시리스트를 마이크로소프트 엑셀로 전달하기 .....	664
조호 API .....	665
요약 .....	666

## 18장 가능한 데이터 포맷으로 마이크로포맷과 RDFa 사용하기 667

Operator를 사용하여 마이크로포맷에 대해 학습하기 .....	668
adr(Addresses) .....	671
hCard(Contacts) .....	672
hCalendar(Events) .....	673
geo(Locations) .....	674
tag(Tagspaces) .....	675
마이크로포맷에 대한 정의와 설계목표 .....	675
마이크로포맷 디자인 패턴 .....	677
rel-design-pattern .....	677
class-design-pattern .....	678
abbr-design-pattern .....	679
include-pattern .....	679
마이크로포맷 예제 .....	680
rel-license .....	680
rel-tag .....	681
xfn .....	681
xFolk .....	682
geo .....	683
hCard와 adr .....	683
hCalendar .....	685
다른 마이크로포맷들 .....	685
실전에서의 마이크로포맷 .....	686
마이크로포맷으로 프로그래밍하기 .....	686
특정 언어용 라이브러리 .....	687
Operator 스크립트 작성하기 .....	688
튜토리얼 스크립트에 대한 이해 .....	688
주소-좌표 변환 스크립트 작성하기 .....	690
마이크로포맷을 보완해줄 RDFa .....	692
향후 연구에 참고할 만한 문서 .....	693
요약 .....	694

## 19장 검색 통합하기 695

구글 Ajax 검색.....	695
검색결과 다루기 .....	696
야후! 검색 .....	698
야후! 이미지 .....	700
마이크로소프트 Live.com 검색 .....	701
OpenSearch.....	706
구글 데스크톱 HTTP/XML 게이트웨이 .....	709
요약.....	710

## 부록 A 아이폰 매쉬업 사이트 만들기 711

오픈 API로 작업하기 .....	711
네이버 오픈 API 키 발급받기.....	711
야후! 거기 오픈 API 이용하기 .....	713
구글 오픈 API 이용하기 .....	715
Rails Project 시작하기 .....	716
rails 의 webrick 서버 구동하기 .....	717
Rails의 MVC (Model, View, Controller).....	718
Model 생성하기 .....	718
Controller생성하기 .....	720
Controller과 View를 함께 생성하기 .....	721
본격적인 Mashupmap 지도 서비스 만들기 .....	723
Model 객체에 테이블 작성 .....	723
Map Controller.....	724
location view.....	728
검색 결과 보여주기 .....	732
맺음말.....	738

## 부록 B 트위터 기반 매쉬업 739

트위터란? .....	739
트위터 기반 매쉬업(트윗봇) .....	745
사용 API 소개 .....	746
구글 애플리케이션 엔진 개발 환경 .....	756
예제 코드 작성 .....	762
맺음말 .....	770

## 부록 C 크리에이티브 커먼즈 771

## 인덱스 777



## 역 자 서 문



## 저 자 소 개



## 감 수 자 소 개

감 사 의 말



# 소 개

- 하우스맵(housingmaps.com)은 크레이그리스트(craigslist.com)의 주택매매/임대관련 목록과 구글맵을 연동한다. 이 서비스가 구글이나 크레이그리스트에 의해 개발된 것이 아니라 개인 프로그래머인 폴 레이드마허Paul Rademacher에 의해 개발되었다는 것

---

<sup>1</sup> [http://en.wikipedia.org/wiki/Mashup\\_\(web\\_application\\_hybrid\)](http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid)), [http://en.wikipedia.org/w/index.php?title=Mashup\\_%28web\\_application\\_hybrid%29&oldid=98002063](http://en.wikipedia.org/w/index.php?title=Mashup_%28web_application_hybrid%29&oldid=98002063)

에 주목하라. 하우스맵은 크레이그리스트가 제공하는 특정 항목의 위치를 지도상에서 보여주는 기능을 추가하였지만 해당 지역의 모든 임대 정보나 매매 정보를 표시하지는 않는다.<sup>2</sup>[지은이]

- 플리커 내의 구글맵(Google Maps in Flickr, GMiF)은 그리즈멍키(Greasemonkey)를 통해 플리커 사진, 구글맵, 구글 어스, 그리고 파이어폭스 브라우저를 통합한다.<sup>3</sup>
- 라이브러리 룩업 북마크(Library Lookup bookmark)는 자바스크립트 기반 북마크릿으로 아마존과 여러분 지역 도서관의 도서목록을 연결한다.<sup>4</sup>

- 아마존에서 찾아낸 책을 즉시 지역 도서관에서 찾아낸다
- 많은 뉴스 관련 소스로부터 야후! 파이프를 사용하여 하나의 새로운 피드를 생성한다.
- 버튼 클릭 한 번으로 플리커의 사진을 블로그로 포스팅한다.
- 사진을 구글맵과 구글 어스에 표시한다.
- 구글맵을 어떻게 프로그래밍하는지 배우기 위해 파이어폭스의 특별한 확장기능을 사용한다.
- 그리즈멍키를 사용하여 웹 페이지 내에 추가적인 정보를 포함시킨다.
- 선호하는 뉴스 소스(뉴욕타임즈와 같은)의 기사를 지도상에 표시한다.
- 웹사이트가 리믹스에 용이하도록 개발하여 다른 사용자가 여러분의 콘텐츠를 사용하여 매쉬업할 수 있도록 한다.
- 사용 중인 이벤트 캘린더로부터 구글 캘린더를 생성한다.
- 파일들을 온라인 스토리지 서비스(아마존의 S3)에 저장하거나 가져온다.

---

2 <http://housingmaps.com/>

3 <http://webdev.yuan.cc/gmif/>

4 <http://weblog.infoworld.com/udell/stories/2002/12/11/librarylookup.html>

- 아마존의 위시리스트로부터 온라인 스프레드시트를 생성한다.
- 웹페이지 내에 포함된 데이터를 인식하고 조작한다.
- 웹에 게시된 이벤트를 캘린더에 추가하고 마우스 클릭 한 번으로 다른 사람에게 메일로 보낸다.
- 웹 애플리케이션에 웹 검색기능을 개발하여 추가한다.
- 웹사이트에 맞게 포매팅된 워드 문서를 재출판한다.

## 웹2.0 운동

- 비즈니스용, 개인용, 그리고 소비자용 애플리케이션(오피스 스위트를 포함)에서 XML 데이터 소스와 XML 데이터 포맷을 사용할 수 있는 가능성의 증가
- 광범위한 XML 웹서비스의 사용
- 데이터 리믹스나 매쉬업에 대한 폭넓은 관심
- Ajax의 활용, 그리고 자바스크립트 기반 위젯(widget), 마이크로 애플리케이션의 유용성

- 확장(예를 들어, 파이어폭스 확장기능과 그리즈명키 스크립트) 가능한 방식으로 진화하는 웹브라우저
- '사용자 생성 콘텐츠' 또는 '사용자 주도 혁신' 부문의 폭발적인 성장
- 인터넷을 플랫폼으로 보는 개념의 보급 ("웹 2.0")
- 광대역 통신망의 보급

# 책의 전반적인 흐름

## 책의 구성

- 1부, ‘프로그래밍하지 않고 정보를 조합하기’는 프로그래밍 기술을 요하지 않는 매쉬업을 소개하고 리믹스가 가능한지를 확인하기 위해 애플리케이션을 분해하고 분석하는 기술을 가르친다.
  - 1장, ‘특정 매쉬업으로부터 배우기’는 매쉬업과 이 책의 전반에 걸쳐 지속적으로 다루게 될 일반적인 주제에 대한 방향을 잡을 수 있도록, 하우스맵(housingmaps.com), 플리커 내에 포함된 구글맵(GMiF), 그리고 라이브러리룩업(LibraryLookup) 북마크릿과 같은 선별된 매쉬업에 대해 자세히 분석한다.
  - 2장, ‘웹 사이트의 매쉬업 가능성에 대해 살펴보기’는 플리커를 주요 예제로 다루며, 특정 애플리케이션의 리믹스 방법과 리믹스에 유용한 기능의 사용방법을 배우는데 있어서 어떤 요소 때문에 플리커가 최적의 매쉬업 플랫폼인지 분석한다. 우리는 플리커(Flickr)를 다른 리믹스 플랫폼인 딜리셔스(del.icio.us), 구글맵(Google Maps), 그리고 아마존(Amazon.com)과 비교해볼 것이다.
  - 3장, ‘태깅Tagging과 포크소너미Folksonomies에 대한 이해’는 태깅에 대해서 다룬다. 사진, 웹사이트뿐만 아니라 웹의 거의 모든 구성요소에 추가할 수 있는 태깅은 웹사이트 내에서와 서로 다른 웹사이트에 걸쳐 많은 것을 함께 묶어주는 역할을 한다. 이번 장은 플리커, 딜리셔스, 그리고 테크노라티(Technorati)에서 태깅이 어떻게 사용되는지 설명하고 어떻게 태그 기반의 재미있는 매쉬업을 생성하는지, 어떻게 사용자들이 태깅 시스템을 ‘해킹’하여 특별한 데이터베이스를 생성하는지, 그리고 어떻게 태그들이 여타 분류 시스템과 연관되는지에 대해 논의할 것이다.



- 4장, ‘피드Feeds, RSS, 그리고 Atom으로 작업하기’는 그 자체로 리믹스를 위한 잠재력 있는 기술이며 XML의 학습에 보다 일반적으로 사용될 수 있는 유형인 RSS와 Atom을 제시한다. 다양한 RSS/Atom 포맷을 다른 절과 정보 리믹스에 있어 이들의 중요성을 설명하는 절은 반드시 눈여겨 보아야 할 부분이다. 이 장은 피드를 필터링하고 합성하는 데 야후! 파이프를 사용하는 방법을 포함한다.
  - 5장, ‘블로그와 통합하기’는 플리커와 블로그를 통합해보는 것으로 시작하여 블로그, 위키, 그리고 이 시스템들이 프로그래밍 가능한지에 대해 알아본다. 블로그와의 통합이 중요한 이유는 지금까지 논의한 태그와 RSS를 통한 데이터 지향적인 리믹스와는 다르게, 블로그가 리믹스 유형을 이야기 식으로 표현하기 때문이다. 이 장은 위키와의 통합에 대한 논의를 간략하게 다루는 것으로 끝맺는다.
- 2부, ‘API를 사용하여 단일 웹 애플리케이션을 리믹스하기’는 각 계층의 표본을 학습함으로써 폭넓은 계층의 웹기반 API를 설명하는 것에 초점을 맞춘다.
- 6장, ‘플리커를 통해 웹서비스 API에 대해 배우기’는 플리커에 대해 자세하게 알아본다. 플리커는 1부에서 프로그래밍을 필요로 하지 않는 다양한 리믹스 기술에 대한 표본일 뿐만 아니라, XML 웹서비스를 배우는 예제로도 아주 훌륭하다. 이 장은 플리커 API를 어떻게 사용하는지 보여준다. 먼저 API를 간단히 호출을 하는 방법을 살펴보는 것으로 시작하여 어떻게 전체 API를 이해하는지, 그리고 어떻게 사용자 인증에 대한 처리를 일반화시키는지 설명한다.
  - 7장, ‘다른 웹 API에 대해 알아보기’는 다양한 API 제공자들, 특히 플리커와 다른 시스템들 간의 공통점과 차이점을 설명하며 사용 가능한 서비스의 유형과 API의 범위에 대해서 어떻게 생각할지 살펴본다. 여러분은 REST, XML-RPC, SOAP 기반 서비스를 어떻게 호출하는지 배우게 될 것이다. 이 장은 다양한 API를 문서화하고 있는 Programmableweb.com과 같은 사이트와 이렇게 방대한 API를 문서화하는 과정에서 발생하는 어려움에 대해 알아본다.
  - 8장, ‘Ajax/자바스크립트 위젯과 위젯 API에 대해 배우기’는 다른 종류의 방대한 애플리케이션, 즉 Ajax를 사용한 자바스크립트 기반 위젯의 리믹스 가능성에 대해서 설명한다. 이 장은 플리커와 다른 애플리케이션을 예로 들어 Ajax를 사용한 애플리케이션과 오래된 스타일의 웹 애플리케이션을 비교하며 어떻게 위젯 프로그래밍을 하는지 설명하기 위해 자바스크립트 위젯 라이브러리인 야후! UI 라이브러리를 소개한다. 또한 자바스크립트를 학습하는 데 파이어버그Firebug라는 파이어폭스 확장

기능과 자바스크립트 셸JavaScript Shell을 어떻게 사용하는지 배우게 될 것이며 이 장은 그리즈멍키Greasemonkey에 대한 소개로 끝을 맺는다.

■ 3부, ‘매쉬업 작성하기’는 이 책의 가장 핵심적인 내용, 즉 여러분이 1부와 2부에서 배운 내용을 활용하여 어떻게 매쉬업을 생성하는가에 대한 내용을 다룬다.

- 9장, ‘API와 리믹스에 사용되는 요소에 대한 내용에서 매쉬업 관련 내용까지 알아 보기’는 매쉬업이 유용한 해결책을 제시할 수 있는 일련의 특정 문제를 연구함으로써 매쉬업에 대한 내용과 매쉬업이 API와 어떤 관계가 있는지 분석한다. 이 장은 여러분이 어떻게 다양한 API를 조합하여 책, 부동산, 항공요금, 그리고 현재 이벤트를 추적할 수 있는지 살펴본다. 여러분은 이러한 문제들을 분석하기 위해 어떻게 Programmableweb.com을 사용하는지 배우게 될 것이다.
- 10장, ‘다양한 서비스로부터 매쉬업 생성하기’는 매쉬업 개발을 초기단계부터 경험 해볼 수 있는 상세한 예제, 즉 처음에는 구글맵 API를 사용하고 그 후에는 구글 맵 플릿Google Mapplets API를 사용하여 위치정보가 태깅된 플리커 사진과 구글맵의 매쉬업을 생성하는 예제를 제공함으로써 매쉬업의 작성 방법에 대해 알아본다.
- 11장, ‘툴을 사용하여 매쉬업 생성하기’는 전통적인 웹 프로그래밍 기술을 사용하는 것에 비해 매쉬업의 생성을 쉽게 해 줄 수 있는 툴에 대해 논의한다. 이 장은 이러한 툴 중의 하나인 구글 매쉬업 에디터Google Mashup Editor의 사용방법을 설명하며 다른 툴들에 대해서도 간략하게 정리한다.
- 12장, ‘매쉬업 가능하도록 사이트 만들기’는 이 책의 초점을 데이터와 API를 ‘사용’하는 것에서 ‘생성’하는 것으로 변경한다. 이 장은 매쉬업이 용이한 사이트를 개발 하려는 콘텐츠 생산자를 위한 지침을 제공한다. 즉, 이번 장은 여러분이 콘텐츠 생산자로서 어떻게 여러분의 디지털 콘텐츠를 사용자나 개발자가 효과적으로 리믹스 하고 매쉬업할 수 있도록 만들 것인가 하는 질문에 답을 제공한다.

■ 4부, ‘다른 매쉬업 관련 주제에 대해 알아보기’는 논의를 이끌어나가기 위해 1부에서 3부에 걸쳐 다뤘던 개념적인 프레임워크를 사용하여 특정 부류의 애플리케이션을 어떻게 리믹스하고 통합하는지에 관해 다룬다.

- 13장, ‘온라인 지도와 3D 디지털 세계를 리믹스하기’는 지도기반 매쉬업에 대한 예제들을 제공하며 인기 있는 온라인 지도 서비스와 가상의 지구virtual globe를 구현한 서비스에 대해 다룬다. 여러분은 프로그래밍과 데이터 교환 포맷(GeoRSS와

KML)을 사용하지 않고 지도를 만드는 방법에 대해 배우게 될 것이며 그 후에 구글 맵, 야후!맵, 그리고 마이크로소프트 맵 등 다양한 API에 대해 배우게 될 것이다. 또한 미국 내 주소의 주소-좌표변환과 미국 내 주소가 아닌 경우의 주소-좌표변환에 대해 다룰 것이다. 이 장은 구글 어스Google Earth, 구글 어스와 KML의 관계, 그리고 플리커 사진을 KML을 통해 어떻게 표시하는지에 대한 논의로 끝을 맺는다.

- 14장, '소셜 북마킹과 도서관 목록 시스템에 대한 탐구'는 "웹에서 한번 찾아낸 것을 지속적으로 관리하는 방법은 없는가?" 라는 근본적인 문제를 어떻게 소셜 북마킹이 해결하는지 다룬다. 이러한 해결방법이 기초적인 수준에서는 URL을 통해 수행되지만 여러분은 이미지와 데이터 세트와 같은 다른 디지털 콘텐츠에 대해서 배우게 될 것이다. 소셜 북마킹은 이 시스템들에 내장된 확장성, 리믹스의 용이성 때문만이 아니라 다른 시스템들에 대한 통찰력을 제공해주기 때문에 흥미롭다. 이 장은 선별된 소셜 북마킹 시스템들간의 상호운용상의 어려움에 대한 논의뿐만 아니라 이들 시스템과 API에 대해서 살펴볼 것이다. 이 장은 플리커와 딜리셔스의 매쉬업을 어떻게 생성하는지 보여줄 것이다.
- 15장, '온라인 캘린더와 이벤트 수집기에 접근하기'는 프로그래밍하지 않고 (iCalendar와 XML 피드를 사용하여) 어떤 데이터를 캘린더로부터 가져오고 추가할 수 있는지, 어떻게 각각의 캘린더를 프로그래밍하는지(구글 캘린더와 30boxes를 사용하여), 그리고 어떻게 각각의 이벤트 수집기 API를 프로그래밍하는지(upcoming.yahoo.com과 eventful.com을 사용하여)를 보여준다. 이 장은 공개 이벤트 캘린더와 구글 캘린더의 매쉬업을 설명하는 것으로 결론을 맺는다.
- 16장, '온라인 스토리지 서비스 사용하기'는 지속적으로 성장하고 있으며 앞으로는 더 중요하게 생각될 시장인 온라인 스토리지 솔루션에 대해 조사하고 아마존의 S3를 어떻게 사용하는지 대한 기본적인 내용을 다룬다.
- 17장, '데스크톱과 웹 기반 오피스 스위트Office Suite 매쉬업하기'는 ODF와 OpenXML을 어떻게 파싱하는지 간단한 예를 보여주고 ODF와 OpenXML 포맷을 사용하여 간단한 문서를 어떻게 생성하는지, 그리고 마이크로소프트 오피스와 오픈오피스의 스크립팅에 대해서 간단한 예제를 설명한 뒤 구글 스프레드시트와 아마존 웹 서비스의 매쉬업을 설명하는 것으로 결론을 맺는다.
- 18장, '포함 가능한 데이터 포맷으로 마이크로포맷과 RDFa 사용하기'는 어떻게 웹 페이지에 사람과 컴퓨터 프로그램 모두가 이해하기 쉬운 방법으로 정보를 포함시킬

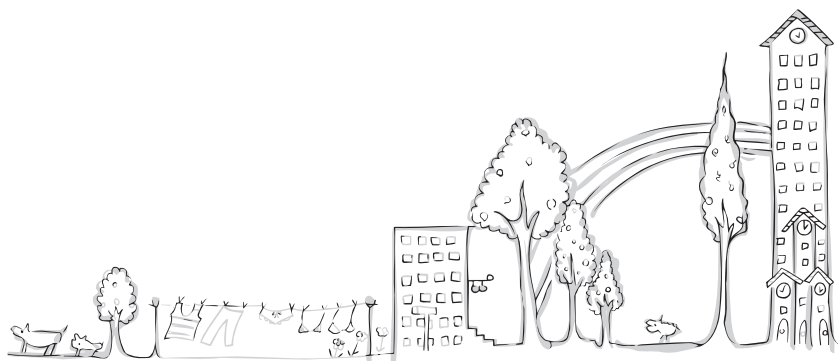
것인가 하는 문제에 대한 두 가지 해답인 마이크로포맷microformats과 RDFa에 대해 연구한다. 여러분은 마이크로포맷을 인식하고 조작하기 위해 어떻게 Operator라는 파이어폭스 확장기능을 사용하고 프로그래밍하는지 배우게 될 것이다.

- 19장, ‘검색 통합하기’는 구글 Ajax 검색 API, 야후! 검색 API, 그리고 마이크로소프트 Live.com 검색을 어떻게 사용하는지 설명하며 OpenSearch와 구글 데스크톱 HTTP/XML 게이트웨이에 대해 소개한다.

## 대상 독자

업데이트

관련 웹사이트 URL 링크



1부

---

# 프로그래밍하지 않고 정보를 조합하기

---





# 01

---

## 특정 매쉬업으로부터 배우기

---

- 무엇이 조합되는가?
  - 왜 이러한 요소들이 조합되는가?
  - 어느 부분에서 조합과 통합이 이루어지는가?
  - 다양한 요소들이 어떻게 조합되는가?(즉, 사용자 인터페이스뿐만 아니라 그 뒤에 숨어 있는 기술적인 장치에 있어서)
  - 어떻게 매쉬업이 확장될 수 있는가?
- 
- 하우징맵(Housingmaps.com)
  - GMiF 그리즈명키 스트립트
  - 존 우델Jon Udell의 라이브러리룩업 북마크릿bookmarklet

# 매쉬업에서 패턴 찾아내기

1. 데이터는 소스Source 웹사이트에서 추출된다.
2. 이렇게 추출된 데이터가 타겟Target 웹 사이트에 활용할 수 있는 형태로 변형된다.
3. 재가공된 데이터가 타겟 웹사이트로 전달된다.

## 용어에 대한 이해

이 책에서는 지적이고 창조적인 작업물을 재사용하여 파생적인 작업물을 만드는 과정에서 나타나는 각종 양상들을 설명하기 위해 매쉬업, 리믹스remix, 재결합recombine, 데이터 그리고 서비스와 같은 다양한 용어들을 사용한다. 물론 재사용 - 예술적인 사용, 학구적인 속성, 문학적인 인용, 암시, 모방 또는 풍자 중에 어떤 형태를 띠든 간에 - 은 인간의 지적, 창조적, 상업적인 시도에 걸쳐 오랜 역사를 갖는다. 재사용과 같은 용어들은 소프트웨어의 재사용 또는 코드의 재사용과 같은 형태로 아주 빈번하게 사용되고 있다. 리믹스, 매쉬업과 같은 용어들은 웹2.0에 관련된 최근의 논의과정에서 만들어졌으며 종종 XML을 이용하거나 XML 기반 웹서비스를 통해 구현되는 전혀 다른 소스가 제공하는 데이터들 간의 결합을 설명하는 데 사용된다. 어떤 면에서 보면, 리믹스라는 용어가 여전히 포괄적인 용도로 미디어를 재사용하는 것에 대한 일반적인 내용을 의미하는 반면(remix culture라는 용어에서처럼), 매쉬업은 콘텐츠를 결합하여 새로운 것을 만들어내는 웹 인터페이스와 애플리케이션을 가리키는 용어로서 제 구실을 해내고 있다.

하지만 매쉬업과 리믹스의 경계는 다소 모호하다. 매쉬업과 리믹스라는 용어는 대중가요에서 유래된 말이다.\*1 개략적으로 말하면, 리믹스는 원본 음악의 변형된 버전인 반면 매쉬업은 두 개나 그 이상의 음악요소를 합친 것을 말한다. 매쉬업이라는 용어는 최근에는 여러 개의 비디오 소스를 조합하여 새로운 비디오를 생성하는 것을 설명하는 용어로까지 확장되어 사용된다.\*2 이러한 점에서, 만약 대중음악과의 유사점을 근거로 디지털 애플리케이션을 설명한다면 다른 콘텐츠와의 결합 없이 데이터를 재사용하거나 다시 패키징하는 시나리오(예를 들어, 플리커의 이미지들만을 포함하는 웹 페이지를 만들기 위해 플리커 API를 사용하는 것과 같은)를 설명하는 데에는 리믹스를 사용할 것이며 매쉬업이라는 용어는 다양한 소스로부터 얻어진 데이터를 결합하는 시나리오(예를 들어, 플리커의 사진들을 피카사의 사진들과 결합하는 경우와 같은)를 설명하는 데 사용할 것이다. 그러나 이 둘 간의 경계가 모호하기 때문에 지나치게 구분하려고 애쓸 필요는 없다고 생각한다.

대체로 이 책에서는 소프트웨어에서의 매쉬업에 초점을 맞출 것이며 대부분의 경우 데이터와 서비스들을 리믹스하는 웹 매쉬업에 대해 초점을 맞출 것이다. 데이터라는 용어를 사용하는 경우는 그것이 컴퓨터 네트워크상에 존재하던 여러분의 컴퓨터 또는 다른 디바이스에 존재하던 디지털 콘텐츠를 의미한다. 서비스라는 용어는 대략적으로 '서비스 지향 아키텍처(SOA)'와 '서비스로서의 소프트웨어(SaaS)' 개념에서 사용되는 재사용가능한 웹서비스나 애플리케이션을 지칭하는 의미로 사용할 것이다.

매쉬업이란 용어는 웹2.0과 밀접하게 연관되어 있지만 복합 애플리케이션 composite application 과 같은 이름으로 엔터프라이즈 컴퓨팅과 서비스 지향 아키텍처 부문에서도 매쉬업과 유사한 형태의 개발이 진행되고 있다. 복합 애플리케이션 역시 데이터와 서비스를 한데 묶어내는 것과 관련이 있지만 구글과 아마존의 공개 API 대신 보통 방화벽 내에 위치한 기업내부의 데이터와 공급망을 통합한다. 비록 매쉬업과 복합 애플리케이션이 공통적인 기술들을 공유한다고 해도 이들은 상당히 다른 문화적인 요소에서 출발하였다.

이 책은 엔터프라이즈 관점에서의 정보 재사용보다는 개인적인 정보의 재사용에 초점을 맞춘다. 개인적인 정보는 그 자체의 이질성, 개인 정보 관리와의 연관성, 대중적인 커스터마이징에 대한 요구, 그리고 하드웨어, 소프트웨어, 개인들의 독특한 욕구로부터 기인한 데이터의 많은 변경 때문에 구별된다. 그럼에도 불구하고, 여기서 엔터프라이즈2.0과의 시너지를 이끌어낼 수 있는 기회가 있다면 그렇게 할 것이다.

\*1 [http://en.wikipedia.org/wiki/Mashup\\_%28music%29](http://en.wikipedia.org/wiki/Mashup_%28music%29) 와 <http://en.wikipedia.org/wiki/Remix>

\*2 [http://en.wikipedia.org/wiki/Mashup\\_%28video%29](http://en.wikipedia.org/wiki/Mashup_%28video%29)

**For Rent**   **For Sale**   **Rooms**   **Sublets**

City: **SF - East Bay**   Price: **\$1500 - \$2000**   [Show Filters](#)   [Refresh](#)   [Link](#)

Powered by [craigslist](#) and [Google Maps](#)

**\$1,500 - 2bd  
Charming West Berkeley  
Duplex**  
8th St At Bancroft  
Berkeley Ca  
510-295-0933

왜 구성요소들이 조합되는가? 풀어야 할 문제들은 어떤 것들이 있는가?

리믹스가 어디에서 이루어지는가?

이러한 요소들이 어떻게 조합되는가?

- 어떻게 하우스징맵이 크레이그리스트로부터 주택과 임대 관련 데이터를 가져오는가?
- 하우스징맵이 이렇게 획득한 데이터를 가지고 어떻게 구글 맵을 생성하는가?

**\* NOTE!** 공개 API와 RSS 피드를 사용하는 것이 일반적으로 웹사이트의 화면에 표시되는 데이터를 분석하는 것보다 나은 방법이다. 화면의 데이터를 분석하는 코드가 잘못 구현되면 데이터 소스에 지나친 부하를 주게 되므로 항상 데이터 소스의 서비스 약관을 준수하고 있는지 확인하자.

우리가 2월초에 사이트를 런칭한 직후부터 ‘해커’들이 구글 맵을 해킹하려고 하는 것을 보면서 이런 저런 이유로 우리는 흥분해 있었다. 정말 몇 일만에, 그들의 블로

---

1 <http://www.google.com/apis/maps/>

2 <http://benmetcalfe.com/blog/index.php/2005/06/29/google-make-map-api-available-finally>

3 Google Maps Hacks. 리치 깁슨(Rich Gibson), 얼리 슈일러(Erle Schuyler) 공저. (오라일리 미디어, 2006년)

그에는 우리가 작성한 설계문서보다 구글 맵의 내부적인 동작에 대해 더 잘 설명한 글들이 게재되었고 곧 놀랄 만한 '해킹'들이 눈에 띄기 시작했다. 필립 린드세이<sup>Philp Lindsay</sup>의 독립실행모드<sup>stand alone</sup> 형태의 구글 맵, 폴 레이드마허<sup>Paul Rademacher</sup>의 하우스맵, 그리고 크리스 스모크<sup>Chris Smoak</sup>의 버스몬스터<sup>Busmonster</sup> 등 많은 것들이 등장했다.

## 비교할 만한 매쉬업들

- 에이드리안 홀로버티<sup>Adrian Holovaty</sup>의 시카고 범죄 지도(<http://chicagocrime.org>). 시카고에서 보고된 범죄 데이터베이스와 구글 맵으로 사용자 인터페이스를 매쉬업하였다.
- 웨더 봉크(<http://www.weatherbonk.com/weather/about.jsp>). 날씨 데이터를 구글 맵과 매쉬업하였다.

## 플리커 내의 구글 맵

---

4 <http://googlemapsmania.blogspot.com/>을 보면 매일 매일 등장하는 많은 수의 구글 맵에 기반한 새로운 매쉬업을 볼 수 있다.

## 무엇이 조합되는가?

- 플리커(<http://flickr.com>)는 잘 알려진 사진 공유 사이트이다.
- 구글 맵(<http://maps.google.com/>)은 온라인 지도 시스템이다.
- 구글 어스(<http://earth.google.com/>)는 지구를 확대하거나 상하좌우로 움직여가면서 볼 수 있게 해주는 훌륭한 데스크톱 애플리케이션이다.
- 파이어폭스 웹브라우저(<http://www.mozilla.com/firefox/>)는 오픈소스 웹브라우저이다. 파이어폭스의 기능들 가운데 주목할 만한 것으로는 모듈을 추가하여 확장할 수 있는 아키텍처를 들 수 있으며 이 기능을 사용하여 개발자들은 브라우저에 새 기능을 추가할 수 있다.
- 그리즈명키(<http://www.greasespot.net/>)는 파이어폭스 확장모듈이며 사용자들이 스크립트를 설치하여 특정 웹 페이지를 런타임에 변경할 수 있도록 해준다. 그리즈명키 스크립트는 영구적으로 적용되기 때문에, 해당 페이지가 열릴 때마다 웹 페이지에 변경된 내용이 실행되어 사용자가 변경내용을 영구적으로 사용할 수 있도록 해준다.<sup>5</sup> 그리즈명키 스크립트는 웹사이트의 작성자가 아닌 사용자가 웹 브라우저 안에서 해당 사이트를 커스터마이징할 수 있게 해준다.

## 왜 구성요소들이 조합되는가? 해결해야 할 문제는 어떤 것이 있는가?

<sup>5</sup> 이전 버전의 <http://en.wikipedia.org/w/index.php?title=Greasemonkey&oldid=97588087> 이 2007년 1월 1일부터 <http://en.wikipedia.org/wiki/Greasemonkey> 로 수정되었다.



## GMiF 스크립트 설치방법

GMiF 그리즈멍키 스크립트를 실행하기 위해서는 반드시 파이어폭스 웹 브라우저와 그리즈멍키 확장모듈, GMiF 스크립트를 함께 사용해야 한다. 아래에 GMiF를 설치하는 방법에 대해 설명해 두었다.

1. 만약 컴퓨터에 파이어폭스가 설치되어 있지 않다면, <http://getfirefox.com>에 접속하여 Download Firefox 버튼을 클릭한 뒤 이어지는 설명에 따라 설치하도록 하자.
2. 이제 파이어폭스용 그리즈멍키 확장모듈을 설치해야하므로 <https://addons.mozilla.org/en-US/firefox/addon/748>(한국어버전은 <https://addons.mozilla.org/ko/firefox/addon/748>)에 접속하자.
3. 'firefox에 추가'버튼을 클릭하여 설치한 뒤 브라우저를 재시작하여 그리즈멍키 확장모듈을 활성화시키도록 하자.
4. 이제 GMiF 그리즈멍키 스크립트를 설치할 차례이다. <http://webdev.yuan.cc/gmif/>에 접속하도록 하자.
5. "Download GM user script:flickr.user.js(latest version)" 링크를 클릭하자. 스크립트를 설치하겠느냐고 묻는 화면이 나타나면 설치를 클릭하자.

- 이 사진은 연관된 위치정보(예를 들어, geo:lat=37.8721, geo:lon=-122.257704, 그리고 geotagged라는 태그정보)를 가지고 있다.
- 사진 우측 상단에 위치한 GMap 버튼을 주의 깊게 보도록 하자. 이 버튼은 GMiF 스크립트가 실행되어 삽입된 버튼으로 여러분의 컴퓨터에 GMiF 그리즈메키 스크립트가 설치되어 있지 않다면 이 버튼을 볼 수 없다.

### Campanile in fog

ADD NOTE
SEND TO GROUP
ADD TO SET
BLOG THIS
ALL SIZES
ORDER PRINTS
ROTATE
DELETE
Gmap

Uploaded on June 9, 2005  
by [Raymond Yee](#)

**Raymond Yee's photostream**

21,023 photos  
[View as slideshow](#)

[+ home](#)
[+ browse](#)
[+ more](#)

**Tags**

UcBerkeley [x]  
campanile [x]  
geotagged [x]

▼ Hide machine tags (2)

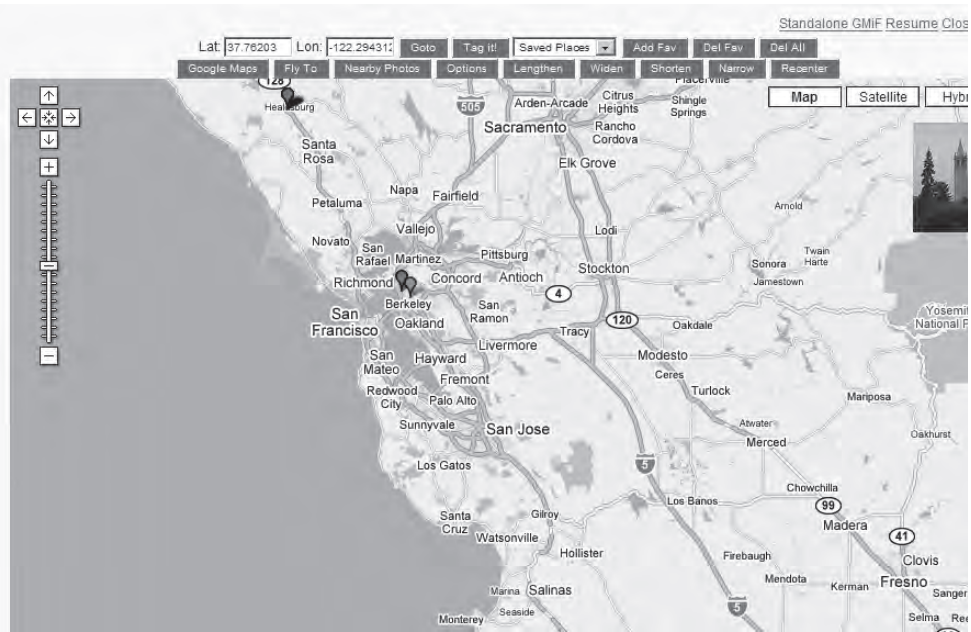
geo:lon=-122.257704 [x]  
geo:lat=37.8721 [x]

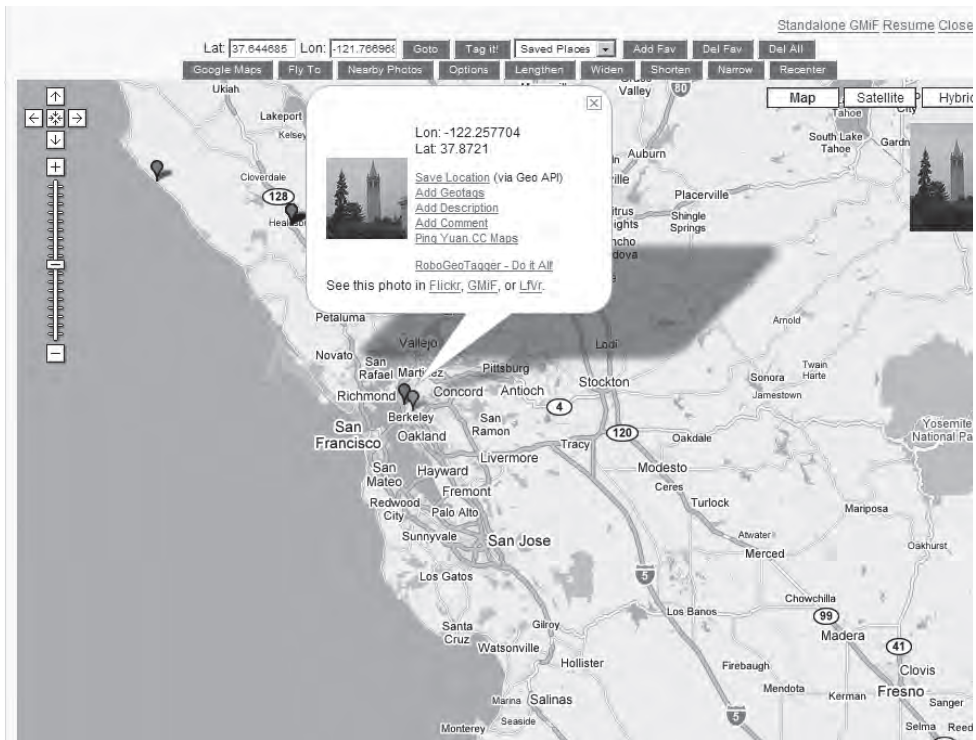
Add a tag

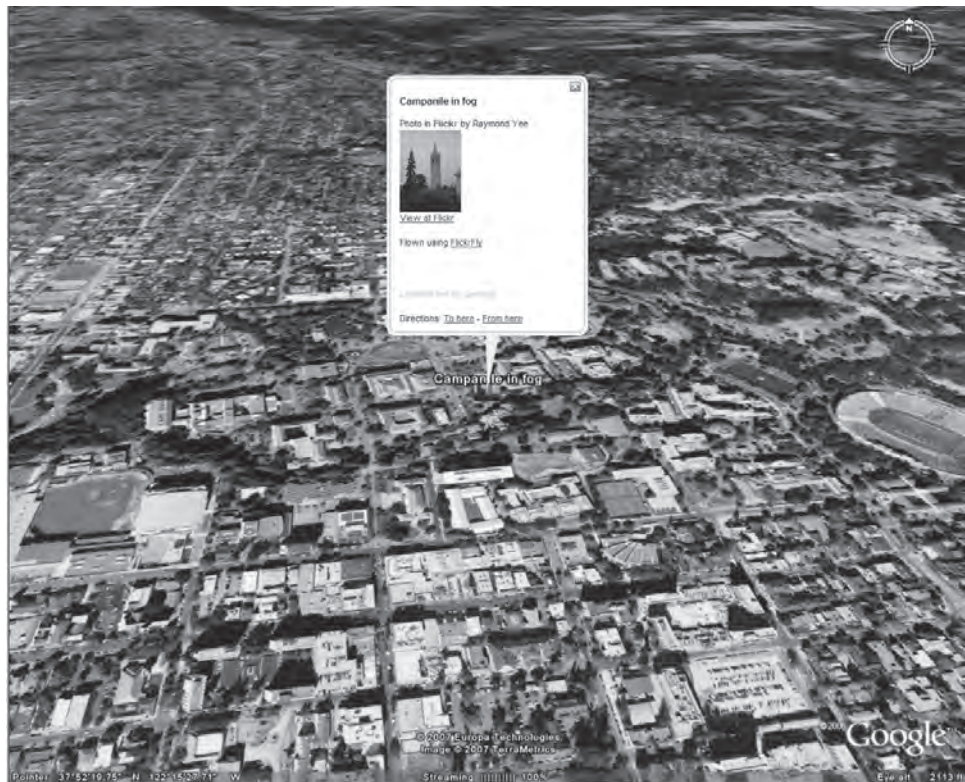
**Additional Information**

Some rights reserved. (edit)

- ◊ Taken in Oakland, California (map)
- ◊ Taken with a Pentax Optio S5i
- ◊ More properties
- ◊ Taken on June 9, 2005 (edit)
- ◊ See different sizes
- ◊ Viewed 165 times (Not including you)
- ◊ Edit title, description, and tags
- ◊ Replace this photo







어떻게 이런 요소들이 조합되는가?

**\* NOTE!** 물론 모든 그리즈명키 스크립트의 경우와 마찬가지로 GMiF에는 잠재적인 함정들이 있어서 그리즈명키 스크립트가 기반을 두고 동작하는 웹 페이지의 HTML과 자바스크립트가 변경되면 작동하지 않을 수 있다. 또, 사용자들이 그리즈명키 스크립트를 사용하게 되기 때문에 플리커에 대한 공통된 경험을 갖는 대신에 인터페이스를 서로 다르게 이해할 수 있다. 여러분만의 버전으로 인터페이스를 변경하여 웹 사이트와의 상호작용을 커스터마이징할 수 있다는 이점들을 고려하면 이 정도의 위험요소쯤은 감수할 만한 가치가 있다고 생각한다.

비교할 만한 매쉬업들

라이브러리룩업 북마크릿

### 공식적인 API 와 비공식적인 API, 그리고 통합 메커니즘

이전 절에서 API나 XML 피드와 같은 공식적인 통합 메커니즘의 확산이 어떻게 보다 많은 매쉬업과 리믹스에 대한 가능성을 낳게 되는지에 대해 언급하였다. 매쉬업이 이러한 공식적인 메커니즘뿐만 아니라 비공식적인 메커니즘에도 의존한다는 사실을 아는 것이 중요하다. 따라서 이 책에서는 공식적인 메커니즘과 비공식적인 메커니즘 모두를 어떻게 찾아내는지 논의할 것이다. 여기서 예로 설명하는 매쉬업들은 위의 두 가지 모두를 사용한다. 이런 비공식적인 방법들을 어떻게 찾아내는지 잘 전달되면 좋겠다.

1. 선택한 도서관에 맞게 북마크릿을 설정하라.
2. 도서관에서 찾고 싶은 책을 웹 페이지에서 찾았을 때 북마크릿을 호출하라.

7 <http://weblog.infoworld.com/udell/stories/2002/12/11/librarylookup.html>

8 이전 버전의 <http://en.wikipedia.org/w/index.php?title=Bookmarklet&oldid=96304211> 이 <http://en.wikipedia.org/wiki/Bookmarklets>로 수정되었다.

## 라이브리룩업 북마크릿 설정하기

<http://weblog.infoworld.com/udell/stories/2002/12/11/librarylookupGenerator.html>

The screenshot shows a web browser window with the address bar displaying the URL: <http://weblog.infoworld.com/udell/stories/2002/12/11/librarylookupGenerator.html>. The browser's tab bar shows several open tabs, including "Blogs and wilds", "bookmark systems", "shell", "jaenv", "Getting Started", "my link", "Latest Headlines", "ecopro", "Message...", "Worldwide Stan...", "LibraryThing: Barnes...", "LibraryThing: Lib...", "Open WorldCat pr...", and "Amazon.com: Pa...".

The main content of the page is titled "The LibraryLookup Bookmarklet Generator". Below the title, there is a message: "Can't find your library on one of the [lists](#)? Try building your own bookmarklet.".

The form consists of the following fields and options:

- Base URL:** A text input field containing "http://www.berkeley-public.org" and an example "(Example: http://kscitlib.keene.edu)".
- Library Name:** A text input field containing "Berkeley Public Library" and an example "(Example: Keene Public Library)".
- Vendor:** A list of radio button options for selecting a library system:
  - ☒ Innovative (example)
  - ☐ Voyager (example)
  - ☐ iPac<sup>1</sup> (example) ISBN
  - ☐ Sirsi/DRA (example)
  - ☐ Sirsi (WebCat)<sup>2</sup> (example)
  - ☐ Sirsi/Web2 (example)
  - ☐ Talls (example)
  - ☐ DS (UK) (example)
  - ☐ Pica (example)
  - ☐ Gaylord (Polaris)<sup>3</sup> (example)
  - ☐ Gaylord (Polaris 3.0) (example)
  - ☐ Koha (example)
  - ☐ EOS Q Series/EOS.Web (example)<sup>4</sup> WEBOPAC
  - ☐ EOS GLAS (example)<sup>5</sup> GLASOPAC
  - ☐ Aleph (older) (example)<sup>6</sup> 7 8
  - ☐ Aleph (newer) (example)
  - ☐ TLC (example)<sup>9</sup>
  - ☐ Winnebago (example)
  - ☐ Athens (example)<sup>10</sup>
  - ☐ Sagebrush (example)

At the bottom of the form, there is a button labeled "Build your new LibraryLookup bookmarklet!".



**\* NOTE!** 만약 도서관이 우텔이 제시한 벤더의 목록에 없더라도, 이 템플릿을 이용하면 약간 다른 시스템으로 운영되는 도서관들에 적용하는 것이 그리 어렵지 않다

## 라이브리룩업 북마크릿 실행시키기

Customers Who Bought This Item Also Bought

## 어떻게 이 매쉬업이 동작하는가?

1. 아마존의 해당 도서 URL로부터 ISBN을 추출해낸다.
2. 지정한 도서관의 목록에서 책을 찾아내기 위해 ISBN을 새로운 URL로 다시 가공한다.

## 이 매쉬업을 어떻게 확장할 수 있는가?

<http://library.berkeley-public.org/search/i=006019667X>

- 첫째는 아마존의 해당 책에 관한 페이지에서 동작하는 그리즈명키 스크립트이다. 스크립트는 먼저 아마존에서 검색한 책과 동일한 ISBN을 가진 책이 우텔의 지역 도서관에 존재하는지를 확인한다. 만약 존재하지 않는다면, 스크립트는 xISBN 서비스에 의해 생성된 해당 책과 관련된 ISBN의 목록을 사용하여 지역 도서관을 검색한다.<sup>10</sup>
- 두 번째 확장모듈은 그리즈명키 스크립트(파이어폭스에 의존적이다)를 인터넷 익스플로러 내에서 동작하도록 포팅한 것이다.<sup>11</sup>

## 비교할 만한 매쉬업들

오늘 나는 xISBN에 대한 해답으로 라이브러리쌍의 thingISBN을 릴리즈한다. 하부구조를 살펴보면, xISBN은 FRBR(Functional Requirements for Bibliographic Records)의 테스트 결과로 도서관 사서들이 도서목록의 연관관계를 모델링할 수 있도록 잘 개발되고 면밀히 고안된 방법이다. 이와는 대조적으로, thingISBN은 “모든 사람이 도서관 사서이다”라는 도서목록 모델링에 대한 라이브러리쌍의 아이디

9 <http://www.worldcat.org/affiliate/webservices/xisbn/app.jsp>

10 <http://weblog.infoworld.com/udell/2006/01/30.html>

11 <http://blog.jonudell.net/2007/04/23/greasemonkeying-with-ie/>

12 <http://elmcity.info/services>

13 <http://bookburro.org/>

14 <http://www.librarything.com/about.php>

15 [http://www.librarything.com/thingology/2006/06/introducing-thingisbn\\_14.php](http://www.librarything.com/thingology/2006/06/introducing-thingisbn_14.php)

어에 기초를 두고 있다. 사용자들은 그들이 적합하다고 생각하는 작품들을 '조합'한다. 만약 사용자들이 실수를 한다면, 다른 사용자들이 '분리'할 수 있다. 이 방법은 약간 혼란스럽지만 몇몇 유용한 결과를 제공한다.

## 다른 매쉬업들 찾아보기

- Programmableweb.com은 매쉬업과 API 세계의 최근 개발 동향에 대한 이야기를 다루고 있다.<sup>17</sup>
- 매쉬업 대시보드는 2007년 8월을 기점으로 Programmableweb.com 데이터베이스 내의 2,220개가 넘는 매쉬업에 대한 개괄적인 내용을 제공한다.<sup>18</sup>

## 요약

---

16 <http://www.frbr.org/categories/librarything/>

17 <http://blog.programmableweb.com>

18 <http://www.programmableweb.com/mashups>

1. 데이터는 소스 웹사이트에서 추출된다.
2. 이 데이터는 타겟 웹사이트에 의미 있는 형태로 변환된다.
3. 재가공된 데이터는 타겟 웹사이트로 전달된다.



## 02

---

### 웹사이트의 매쉬업 가능성 살펴보기

---

## 어떤 요소가 웹 사이트와 애플리케이션을 매쉬업 가능하도록 만드는가

### 용어에 대한 이해

여러분은 해킹이라는 용어가 사용되는 서로 다른 경우를 조심스럽게 구별해야 한다. 해킹 가능한 사이트를 원한다고 말하는 경우, 사람들이 웹사이트의 보안요소들을 쉽게 깨뜨릴 수 있어야 한다는 것을 의미하지는 않는다. 그러한 행위들은 해킹이 아니라 크래킹이다. 해킹 가능한 사이트를 설계한다는 것은, 사이트가 확장 가능하도록 설계하는 것이며 심지어는 고도의 변형적인 방법으로 확장될 수 있도록 설계하는 것이다. 에릭 레이몬드Eric Raymond가 Jargon File 웹사이트에 정리한 해커와 관련된 정의(<http://www.catb.org/jargon/html/H/hacker.html>)에서 특히 다음 부분을 보도록 하자.



- “프로그래밍 가능한 시스템의 세부사항과 이 시스템의 기능을 향상시키는 방법의 탐구를 즐기는 사람”
- “창조적으로 한계를 극복하거나 우회하는 지적인 도전을 즐기는 사람”

다음 정의가 이제 해커를 나타내는 의미로 사용되지 않는다는 것에 주의하도록 하자.

“이리저리 찢어보며 민감한 정보를 발견하려고 시도하는 악의적인 해방꾼으로서 패스워드 해커, 네트워크 해커와 같은 사람을 지칭하는 경우의 올바른 용어는 크래커이다.”

몇몇 사람들은 ‘해킹이 용이한 설계’(http://www.brianoberkirch.com/2007/04/03/designing-for-hackability/)에 관해 논의하고 있다.

웹사이트와 웹사이트의 기능 그리고 웹사이트가 어떻게 구성되었는지에 대해 주의 깊게 연구하는 행위를 리버스 엔지니어링(reverse engineering)이라고 지칭할 것이다. 몇 가지 기술에 대해서는 개략적으로 설명할 것이지만 실제로는 여기서 언급하는 기술보다 더 많은 기술이 존재한다. 리버스 엔지니어링은 이 사회에서 오랫동안 사용된 전통과 같은 것이지만 리버스 엔지니어링을 사용함에 있어 법적, 윤리적인 이슈들에 대해 알고 있어야 한다. 이것에 관한 정보는 <http://www.chillingeffects.org/reverse/faq.cgi> 를 참조하도록 하자(물론 여기서 언급하는 내용은 리버스 엔지니어링에 대한 법적인 조언은 아니다).

## 웹사이트의 주요 요소를 확인하기

명세서에서는 어떤 것이 리소스가 될 수 있는지에 대해 제한을 두지 않고 있다. 오히려 URI에 의해 식별되기만 한다면 일반적인 의미에서 ‘리소스’라는 용어가 사용된다. 리소스라는 용어를 사용하는 익숙한 예로는 전자문서, 이미지, 일관된 목적을 가진 정보(예를 들어 ‘로스엔젤리스의 오늘의 날씨’), 서비스(예를 들어, HTTP와

1 <http://tools.ietf.org/html/rfc3986#section-1.1>

SMS를 연결하는 게이트웨이), 그리고 다른 리소스의 집합 등이 있다. 리소스가 반드시 인터넷을 통해 접근 가능해야 할 필요는 없다. 이를 떼면 사람, 회사, 그리고 도서관에 보관된 제본된 책들 또한 리소스가 될 수 있다. 마찬가지로, 수학의 방정식에 있어 연산자와 피연산자, 관계의 형태(예를 들어 '부모'나 '종업원'), 또는 수치적인 값들(예를 들어 0, 1, 그리고 무한대 값)과 같이 추상적인 개념들도 리소스가 될 수 있다.

<http://www.flickr.com/photos/raymondye/508341822/>

공개 API와 기존의 매쉬업

## Ajax 사용

## 스크립트를 임베딩할 수 있는가

- 아이구글(<http://www.google.com/ig>)을 확장하기 위한 구글가젯(<http://www.google.com/ig/directory>)과 구글 맵(<http://maps.google.com/maps/mm?mapprev=1>)을 확장하기 위한 구글 맵플릿(<http://www.google.com/apis/maps/documentation/mapplets/>)
- 윈도우 라이브(<http://live.com>)를 확장하기 위한 마이크로소프트 웹가젯(<http://gallery.live.com/>)
- 워드프레스 플러그인(<http://codex.wordpress.org/Plugins>)
- 미디어위키 확장모듈([http://www.mediawiki.org/wiki/MediaWiki\\_extensions](http://www.mediawiki.org/wiki/MediaWiki_extensions))
- 마이크로소프트 오피스의 매크로와 애드인(<http://msdn2.microsoft.com/en-us/office/default.aspx>)
- 오픈오피스의 매크로와 애드인, 애드온(<http://wiki.services.openoffice.org/wiki/Extensions>)
- 야후! 위젯(<http://widgets.yahoo.com/>)
- 스케치업 루비(<http://www.sketchup.com/?sid=79>)

## 브라우저 플러그인

- 베타 플리커 파이어폭스 익스텐션(<http://lifelacker.com/software/lifelacker-code/upgrade-flickr-with-the-better-flickr-firefox-extension-263985.php>)

- 딜리셔스 파이어폭스 익스텐션(<http://del.icio.us/help/firefox/extension>)
- S3폭스 아마존 S3 파이어폭스 오거나이저(<https://addons.mozilla.org/en-US/firefox/addon/3247>)

## 웹사이트에서 데이터를 가져오거나 외부로 내보내기

## 사용자와 개발자 커뮤니티

모바일과 대안 인터페이스 그리고 웹사이트 스킨의 변경이 자유로운가

문서화

웹사이트가 오픈소스 정책을 지원하는가

지적재산권, 재사용성, 그리고 크리에이티브 커먼즈

<http://creativecommons.org>

태깅, 피드, 그리고 웹로깅(블로깅)

## 웹사이트의 URL 언어

“주소를 통한 접근성은 여러분이 운영하는 서비스의 모든 흥미로운 요소들이 외부에서 즉시 접근 가능함을 의미한다. 서비스의 흥미로운 요소들은 모두 URI(컴퓨터를 사용하는 모든 사람들에게 친숙한 포맷인 유일한 식별자)를 가지고 있다. 주소를 통한 접근성은 다른 사람들이 여러분의 서비스를 매쉬업할 수 있게 해주어 여러분이 전혀 상상하지 않았던 방식으로 서비스가 사용될 수 있게 해준다.”

---

4 URI를 주소로 통한 접근성, 세밀함, 투명성 그리고 지속성의 관점에서 분석하는 아이디어는 RESTful Web Services(오라일리 미디어, 2007. 레오나드 리차드슨, 샘 루비 공저)에서 인용하였다.



## URI, URL 그리고 URN 사이의 관계

URI(Universal Resource Identifier)는 특별한 형태의 식별자이며 URI는 URL(Universal Resource Locator)과 URN(Universal Resource Name)의 두 종류로 나뉜다. 여러분은 전자에 더 익숙할 것이다. 후자의 예로는 이 책을 가리키는 `urn:isbn:159059858X`을 들 수 있다. 이 책에서 URL이라고 언급한 것은 일반적으로 URI에 해당한다.

RFC3986은 URI, URL 그리고 URN의 관계를 다음과 같이 정리하고 있다(<https://tools.ietf.org/html/rfc3986#section-1.1.3>).

URI는 나아가 위치식별자, 이름, 또는 이 두 가지 모두로 분류될 수 있다. 'URL(Uniform Resource Locator)'이란 용어는 URI의 서브셋이며 리소스를 식별하는 것에 더하여 주요 접근 메커니즘(예를 들어 그것의 네트워크상에서의 '위치')을 기술함으로써 리소스의 위치를 찾는 방법을 제공한다. URN(Uniform Resource Name)이란 용어는 'urn' 체계하에서[RFC2141] 리소스가 존재하지 않거나 사용 불가능한 상태가 되더라도 전역적으로 유일하고 지속상태로 남아 있도록 요구되는 URI와 이름이라는 속성을 갖는 URI 모드를 참조하는 데에 오랫동안 사용되어 왔다.

각 체계가 '이름'이나 '위치식별자' 중 하나로 구분될 필요는 없다. 주어진 체계에 있어 URI는 이름이나 위치식별자 또는 두 가지 모두의 특징을 가질 수도 있으며, 종종 체계의 품질이 아니라 이름을 부여하는 기관이 할당한 식별자의 지속성과 관리에 의존한다. 앞으로 발표될 명세서와 관련 문서들은 제한적인 용어인 'URL'과 'URN'보다는 좀 더 일반적인 용어인 'URI'를 사용해야만 한다.[RFC3305]

## 몇몇 매쉬업에 대해 간략하게 다시 보자

- 하우징맵은 크레이그리스트가 공개 API를 제공하지 않기 때문에 크레이그리스트의 웹 페이지를 스크린-스크레이핑한 것과 크레이그리스트의 RSS 피드를 조합한 결과에 의존한다(9장에서 하우징맵의 숨겨진 로직에 대해서 보다 자세히 분석해볼 것이다.)
- 플리커 내에 구글 맵이 로딩되도록 하고(GMiF) 플리커 인터페이스에 아이콘을 붙여 넣는 그리즈명키 스크립트는 플리커가 제공하는 공개 API에 대한 이해와 더불어 사용자 커뮤니티가 어떻게 사진의 위치정보를 태깅하는지, HTML 페이지 내에 포함된 정보를 어떻게 스크린-스크레이핑하는지와 같은 플리커 UI에 대한 이해에 기초를 두고 있다.
- 라이브러리룩업 북마크릿의 생성은 주로 도서 정보의 소스(즉, 아마존)와 쿼리를 수행할 대상(즉, 도서관의 도서목록)의 URL을 이해하는 것에 의존적이다.

## 본질적으로 매쉬업이 용이한 플리커

“이러한 일을 해내기 위해서, 우리는 웹을 통해서, 무선 디바이스를 통해서, 사용자의 집에 있는 컴퓨터를 통해서, 그리고 사용자들이 사진을 관리하기 위해 어떤 소프트웨어를 사용하든 간에 가능한 한 많은 방법으로 시스템에서 사진을 가져오고 내보낼 수 있기를 원한다. 그리고 플리커의 웹사이트에서, RSS 피드로, 이메일로, 외부의 블로그에 포스팅하는 것을 통해서 또는 아직 생각해내지 못한 방법을 통해서 사진들을 가능한 한 많은 방법으로 전달할 수 있기를 원한다. 이 외에 또 어떤 방법으로 저장된 사진을 사용할 수 있을까?”

**! CAUTION!** 이어지는 플리커에 대한 분석에서, 이번 장에서 제기했던 의문점들에 대해 전부는 아니지만 많은 답을 제시할 것이며 몇몇 질문에 대해서는 이 책의 후반부에 답을 제시할 것이다. 또한, 플리커가 꾸준히 진화하는 웹사이트이므로 여기에서 다룬 플리커 관련 내용들이 변경될 수도 있다는 것을 이해하는 것이 중요하다. 바라는 것이 있다면 웹사이트에 대해 생각할 수 있는 충분한 방법을 제시하여 여러분이 이러한 변화에 적응할 수 있도록 하는 것이다.

## 플리커의 리소스

- 플리커를 사용하면서 내가 사용할 수 있는 각 기능을 살펴보았다. 각 기능에 있어서, 동작에 관련된 ‘명사’ 또는 요소들을 식별해내었고 관련된 URI와 애플리케이션의 변화에 따라 URL이 어떻게 변경되는지 주목하였다.
- 플리커 UI, UI관련 문서, 그리고 API에 대한 문서(<http://www.flickr.com/services/>

api/)에서 공통으로 사용되는 용어를 추출하였다. API의 구조는 종종 웹사이트 내의 중요한 요소를 가리킨다.

**! CAUTION!** “플리커 API는 사용자, 사진, 사진세트 그리고 유일하게 식별 가능한 다른 객체에 대해서 식별자를 사용할 수 있게 해준다. 이러한 식별자 ID들은 항상 특정한 형태의 정수가 아닌 정형화되지 않은 문자열로 다루어져야 한다. 왜냐하면, ID의 포맷은 시간이 흐르면 변경될 수 있으므로 현재 포맷에 의존하면 나중에 문제를 일으킬 수도 있기 때문이다.”<sup>5</sup> 이와 같이 플리커에서 유일한 식별자가 변경될 수 있다는 경고에 주의하도록 하자.

## 사용자와 사진

<http://www.flickr.com>

**\* NOTE!** URL 언어를 표현할 때 URI 템플릿([http://bitworking.org/news/URI\\_Templates](http://bitworking.org/news/URI_Templates))을 사용한다. 이것은 전체 URI을 생성하기 위해 추후에 치환될 변수를 추가해놓은 문자열이다. 치환될 변수들은 URI의 일부가 아닌 {}로 구분되어 있다. URI 템플릿이 현재 IETF 드래프트 상태에 있으며 여기서 사용하는 표기법은 단지 포함된 변수를 {}로 나타내는 것뿐이라는 것에 주의하자. 치환된 변수들은 적절하게 URL 인코딩(<http://en.wikipedia.org/wiki/Percent-encoding>)되어야 한다.

<http://www.flickr.com/people/{user-id}/>

---

<sup>5</sup> <http://www.flickr.com/services/api/misc.overview.html>

- 사용자가 계정을 생성할 때 플리커가 생성한 NSID(@문자를 포함하는 유일한 식별자)로서 48600101146@N01 과 같은 형태를 갖는다.
- 사용자가 설정한 '영구적인 별칭' 이나 사용자 지정 URL 핸들로 [http://www.flickr.com/profile\\_url.gne](http://www.flickr.com/profile_url.gne) 에서 설정할 수 있으며 raymondye와 같은 형태를 갖는다.

<http://www.flickr.com/people/48600101146@N01/>

<http://www.flickr.com/people/raymondye/>

<http://www.flickr.com/photos/upload/>

<http://www.flickr.com/photos/{user-id}/>

## 사진의 표시

<http://www.flickr.com/photos/{user-id}/{photo-id}/>

<http://www.flickr.com/photos/raymondye/508341822/>

<http://www.flickr.com/services/api/misc.urls.html>



- 
- 사진의 컨텍스트 페이지(사진을 포함하는 페이지)에 대한 URL
  - 다양한 사이즈의 사진 자체에 대한 URL

`http://www.flickr.com/photo_zoom.gne?id={photo-id}&size={context-type}`

- farm-id
- server-id
- photo-secret
- original-secret
- file-suffix

- 원본사진의 경우, 다음과 같은 형태를 보이며 file-suffix는 jpg, gif 또는 png이다.

`http://farm{farm-id}.static.flickr.com/{server-id}/{photo-id}_{o-secret}_o.{file-suffix}`

- 원본사진에서 생성된 중간 크기의 사진을 제외한 모든 크기의 사진들의 경우, 다음과 같은 URL을 갖는다.

`http://farm{farm-id}.static.flickr.com/{server-id}/{photo-id}_{photo-secret}_{photo-size}.jpg`

- 중간 크기의 사진의 경우는 다음과 같은 URL을 갖는다.

`http://farm{farm-id}.static.flickr.com/{server-id}/{photo-id}_{photo-secret}.jpg`



---

▶ **TIP!** 설명하는 문서를 자주 확인하기를 바란다. 그리고 플리커가 확장됨에 따라 계속 URL이 변경될 것이라 추측된다. 이전 형태로 생성한 URL에 대해서는 걱정할 필요가 없으며 플리커는 이전 형태의 URL이 계속 동작하도록 보장할 것이다(하지만 가능하다면 소프트웨어에서 가장 최근의 URL 구조를 사용하게 하는 것이 좋다)

## 개별 사진과 연관된 데이터

- 제목
- 설명
- 태그
- 사진이 촬영된 위치를 나타내는 머신 태그Machine tag
- 일자(사용 가능하다면 사진이 촬영된 일자와 업로드된 일자)
- EXIF 데이터
- 사진의 소유주
- 사진이 속한 사진 세트
- 사진이 속한 그룹
- 코멘트



- 노트
- 권한에 따른 노출수준

## 속성을 수정하는 여러 가지 방법

[http://flickr.com/photo\\_edit.gne?id={photo-id}](http://flickr.com/photo_edit.gne?id={photo-id})

[http://www.flickr.com/photo\\_exif.gne?id={photo-id}](http://www.flickr.com/photo_exif.gne?id={photo-id})

[http://www.flickr.com/photo\\_exif.gne?id=688436870](http://www.flickr.com/photo_exif.gne?id=688436870)

[http://www.flickr.com/photo\\_date\\_taken.gne?id={photo-id}](http://www.flickr.com/photo_date_taken.gne?id={photo-id})

## 태그

<http://www.flickr.com/photos/tags/>

<http://www.flickr.com/photos/tags/{tag}/>

<http://www.flickr.com/photos/tags/flower/>

<http://www.flickr.com/photos/tags/flower/?page={page-number}>

<http://www.flickr.com/photos/tags/{tag}/interesting/>

<http://www.flickr.com/photos/tags/{tag}/clusters/>

<http://www.flickr.com/photos/tags/flower/clusters/>

<http://www.flickr.com/photos/{user-id}/tags/>

<http://www.flickr.com/photos/{user-id}/alltags/>

<http://www.flickr.com/photos/{user-id}/tags/{tag}/>

<http://www.flickr.com/photos/{user-id}/tags/{tag}/edit/>

<http://www.flickr.com/photos/{user-id}/tags/{tag}/delete/>

<http://www.flickr.com/photos/{user-id}/tags/{tag}/show/>

## 사용자의 저장소에서 사진을 일자별로 브라우징하기

<http://www.flickr.com/photos/{user-id}/archives/>

<http://www.flickr.com/photos/raymondye/archives/>

<http://www.flickr.com/photos/{user-id}/archives/{date-taken-or-posted}/>

<http://www.flickr.com/photos/{user-id}/archives/{date-taken-or-posted}/{archive-view}>

<http://www.flickr.com/photos/{user-id}/archives/{date-taken-or-posted}/{year}>

<http://www.flickr.com/photos/{user-id}/archives/{date-taken-or-posted}/{year}/{month}>

<http://www.flickr.com/photos/{user-id}/archives/{date-taken-or-posted}/{year}/{month}/{day}>

<http://www.flickr.com/photos/{user-id}/archives/{date-taken-or-posted}/{year}>

<http://www.flickr.com/photos/{user-id}/archives/{date-taken-or-posted}/{year}/{archive-view}>

<http://www.flickr.com/photos/{user-id}/archives/{date-taken-or-posted}/{year}/{month}/{archive-view}>

<http://www.flickr.com/photos/{user-id}/archives/{date-taken-or-posted}/{year}/{month}/{day}/{archive-view-except-calendar}>

<http://www.flickr.com/photos/raymondye/archives/date-taken/2007/>

<http://www.flickr.com/photos/raymondye/archives/date-taken/2007/06/22/>

<http://www.flickr.com/photos/raymondye/archives/date-posted/2007/calendar/>

## 세트

<http://www.flickr.com/photos/{user-id}/sets/>

<http://www.flickr.com/photos/{user-id}/sets/{set-id}/>

<http://www.flickr.com/photos/{user-id}/sets/{set-id}/{set-view}>

<http://www.flickr.com/photos/raymondye/sets/72157600434284985/>

<http://www.flickr.com/photos/raymondye/sets/72157600434284985/detail/>

<http://www.flickr.com/photos/{user-id}/{photo-id}/in/set-{set-id}/>

<http://www.flickr.com/photos/raymondye/591991800/in/set-72157600434284985/>

## 컬렉션

<http://www.flickr.com/photos/{user-id}/collections/>

<http://www.flickr.com/photos/{user-id}/collections/{collection-id}>

<http://www.flickr.com/photos/raymondye/collections/72157600592620295/>

## 좋아하는 사진

<http://www.flickr.com/photos/{user-id}/favorites/>

<http://www.flickr.com/search/?w=faves&q={search-term}>

## 사용자의 인기사진

<http://www.flickr.com/photos/{user-id}/{popular-mode}/>

## 연락처 목록

<http://www.flickr.com/people/{user-id}/contacts/>

<http://www.flickr.com/people/{user-id}/contacts/?see={contact-type}>

<http://www.flickr.com/people/{user-id}/contacts/ignore/>

<http://www.flickr.com/people/{user-id}/contacts/rev/>

<http://www.flickr.com/invite.gne>

## 그룹

<http://www.flickr.com/groups/>

<http://www.flickr.com/groups/{group-id}/>

[http://www.flickr.com/groups\\_url.gne?id={group-nsid}](http://www.flickr.com/groups_url.gne?id={group-nsid})

<http://www.flickr.com/groups/34427469792@N01/>

<http://www.flickr.com/groups/central/>

<http://www.flickr.com/groups/{group-id}/discuss/page{page-number}/>

[http://www.flickr.com/groups\\_newtopic.gne?id={group-nsid}](http://www.flickr.com/groups_newtopic.gne?id={group-nsid})

[http://www.flickr.com/groups\\_newtopic.gne?id=34427469792@N01](http://www.flickr.com/groups_newtopic.gne?id=34427469792@N01)

<http://www.flickr.com/groups/{group-id}/discuss/{thread-id}/>

<http://www.flickr.com/groups/central/discuss/140537/>

<http://www.flickr.com/groups/{group-id}/discuss/{thread-id}/#comment{comment-id}>

<http://www.flickr.com/groups/central/discuss/140537/#comment1192964>

<http://www.flickr.com/groups/{group-id}/discuss/{thread-id}/{thread-action}>

<http://www.flickr.com/groups/{group-id}/discuss/{thread-id}/{comment-id}/{comment-action}/>

<http://www.flickr.com/groups/{group-id}/pool/>

<http://www.flickr.com/groups/central/pool/>

<http://www.flickr.com/groups/{group-id}/pool/map?mode=group>

<http://www.flickr.com/groups/{group-id}/pool/tags/>

<http://www.flickr.com/groups/{group-id}/pool/tags/{tag}/>

<http://www.flickr.com/groups/{group-id}/pool/{user-nsid}/>

**계정관리**



<http://www.flickr.com/photos/friends/>

<http://www.flickr.com/account>

<http://www.flickr.com/account?tab=e-mail>

<http://www.flickr.com/account/?tab=privacy>

<http://www.flickr.com/account/?tab=extend>

<http://www.flickr.com/account/order/history/>

<http://www.flickr.com/account/prefs/screenname/>

<http://www.flickr.com/account/prefs/layout/>

## 플리커 브라우징하기

<http://www.flickr.com/explore/>

<http://www.flickr.com/explore/interesting/>

“플리커에서 사진을 흥미롭게 만드는(혹은 그렇지 않게 만드는) 요소는 아주 많이 있다. 어느 사진에서 클릭이 많이 발생하는가, 누가 언제 사진에 코멘트를 작성하는가, 누가 좋아하는 사진으로 설정하였는가, 누가 태그를 설정하였는가 등등 많은 요소가 있고 이 요소들은 끊임없이 변한다. 흥미의 정도는 시간이 흐르면 플리커에 더 많은 재미있는 사진과 이야기가 추가됨에 따라 변경된다.”

<http://www.flickr.com/explore/interesting/7days/>

<http://www.flickr.com/explore/interesting/{year}/{month}/>

<http://www.flickr.com/explore/interesting/{year}/{month}/{day}>

<http://www.flickr.com/explore/interesting/{year}/{month}/{day}/show/>

<http://www.flickr.com/explore/interesting/2007/01/04/show/>

## 검색

### 기본 사진 검색

<http://www.flickr.com/search/?w={search-scope}&q={search-term}&m={search-mode}>

- 상세보기와는 대조적으로 &z=t을 사용하여 썸네일을 검색할 수 있다.
- 흥미의 정도나 최근 일자를 기준으로 정렬하려면 &s=int 또는 &s=rec 를 사용할 수 있다.
- 결과를 페이지단위로 보려면 &page={page-number} 를 사용할 수 있다.

### 고급수준의 사진 검색

<http://www.flickr.com/search/?q=flower+-rose+-tulip&m=tags&ct=0>

&ss={safe-search}

&ct={content-type}

- 사진의 경우는 0
- 스크린샷의 경우는 1
- 아트, 드로잉, CGI 그리고 기타에 대해서는 2
- 사진과 스크린샷의 경우는 3
- 스크린샷과 아트, 드로잉, CGI, 기타에 대해서는 4
- 사진과 아트, 드로잉, CGI, 기타에 대해서는 5
- 사진과 아트, 드로잉, CGI, 기타 그리고 스크린샷에 대해서는 6

&d={taken-or-posted}-{from-date}-{to-date}

&d=posted--20070702

&d=taken-20070613-20070702

&l={CC-license}

❖ **NOTE!** 그룹(<http://flickr.com/search/groups/>)과 사용자(<http://flickr.com/search/people/>)의 검색에 대한 URL 언어는 상세하게 분석하지 않는다

## 위치정보가 태깅된 플리커 사진

<http://www.flickr.com/map/>

[http://www.flickr.com/map/?&fLat={lat}&fLon={lon}&zl={zoom-level}&map\\_type={map-type}](http://www.flickr.com/map/?&fLat={lat}&fLon={lon}&zl={zoom-level}&map_type={map-type})

- 검색어를 사용하여 필터링하려면 다음을 덧붙인다.

&q={search-term}

- 그룹을 기준으로 필터링하려면 다음을 덧붙인다.

&group\_id={group-nsid}

- 사용자를 기준으로 필터링하려면 다음을 덧붙인다.

&user\_id={user-nsid}

- yyyy-mm-dd%20hh:mm:ss: 의 형태인 일자정보를 사용하여 사진이 촬영된 일자 (date-taken)를 경계로 필터링하려면 다음을 덧붙인다.

&min\_taken\_date={taken-date}

&max\_taken\_date={taken-date}

&min\_upload\_date={upload-date}

&max\_upload\_date={upload-date}

여기서 upload-date는 유닉스 시스템의 타임스탬프이다. (세계시간을 기준으로 하여 1970년 1월 1일을 기점으로 경과된 시간을 초단위로 나타낸 것).

- 페이지를 기준으로 필터링하려면 다음을 덧붙인다.

&page={page-number}

- 흥미의 정도를 기준으로 필터링하려면 다음을 덧붙인다.

&s=int

<http://www.flickr.com/map/?&q=flower&fLat=37.871268&fLon=-122.286414&zl=4>

<http://www.flickr.com/search/?&q=flower&m=text&s=rec&b=-122.346496,37.847598,-122.226333,37.894938&a=10&d=taken-19700101->

&b={lon0},{lat0},{lon1},{lat1}

&a={accuracy}

## 플리커의 사진, 비디오 관리

<http://www.flickr.com/photos/organize/>

[http://www.flickr.com/photos/organize/?start\\_batch=recent\\_uploads](http://www.flickr.com/photos/organize/?start_batch=recent_uploads)

[http://www.flickr.com/photos/organize/?start\\_tab=sets](http://www.flickr.com/photos/organize/?start_tab=sets)

[http://www.flickr.com/photos/organize/?start\\_batch=untagged&mode=together](http://www.flickr.com/photos/organize/?start_batch=untagged&mode=together)

## 최근 활동

[http://www.flickr.com/recent\\_activity.gne?days={time-period}](http://www.flickr.com/recent_activity.gne?days={time-period})

- 일자 수를 나타내는 자연수(어느 선까지 제한이 있겠지만 확인하려고 시도해본 적은 없다).
- 자연수와 뒤에 h를 붙여 시간을 나타낸다.
- 공백으로 두면 '마지막 로그인 이후'를 의미한다.

## 메일링 인터페이스

<http://www.flickr.com/messages.gne?ok=1>

[http://www.flickr.com/messages\\_sent.gne](http://www.flickr.com/messages_sent.gne)

[http://www.flickr.com/messages\\_read.gne?id={message-id}](http://www.flickr.com/messages_read.gne?id={message-id})

[http://www.flickr.com/messages\\_write.gne](http://www.flickr.com/messages_write.gne)

## 블로그와 인터페이스하기

<http://www.flickr.com/blogs.gne>

[http://www.flickr.com/blogs\\_edit.gne?id={blog-id}](http://www.flickr.com/blogs_edit.gne?id={blog-id})

[http://www.flickr.com/blogs\\_layout.gne?id={blog-id}&edit=1](http://www.flickr.com/blogs_layout.gne?id={blog-id}&edit=1)

## RSS와 Atom 피드의 배포

<http://www.flickr.com/services/feeds/>

## 무선을 통한 접근

<http://www.flickr.com/account/uploadbye-mail/>

<http://m.flickr.com>

## 서드파티 플리커 애플리케이션

<http://bighugelabs.com/flickr/>

## 크리에이티브 커먼즈 라이선스의 적용



<http://www.flickr.com/creativecommons/>

<http://www.flickr.com/creativecommons/{cc-license}/>

- by-2.0
- by-nd-2.0
- by-nc-nd-2.0
- by-nc-2.0
- by-nc-sa-2.0
- by-sa-2.0

<http://creativecommons.org/licenses/>

## 카메라

<http://www.flickr.com/cameras/>

<http://www.flickr.com/cameras/brands/>

`http://www.flickr.com/cameras/{camera-company}/{camera-model}/`

`http://www.flickr.com/cameras/canon/`

`http://www.flickr.com/cameras/canon/powershot_sd600/`

## URL 템플릿과 임베딩 패턴에 의한 매쉬업

`http://www.flickr.com/photos/{user-id}/archives/{date-taken-or-posted}/{year}/{month}/  
{archive-view}`

`http://www.flickr.com/photos/raymondye/archives/date-taken/2007/06/calendar/`

`http://blog.mashupguide.net/{year}/{month}`

`http://blog.mashupguide.net/2007/06/`



■ **\* NOTE!** <https://api.del.icio.us/v1/posts/dates>에서 일자별로 게시물의 목록을 얻어올 수 있으며 <https://api.del.icio.us/v1/posts/get?> 에서 특정 게시물을 가져올 수 있다. 또한 딜리셔스가 여러분이 매일 작성하는 게시물을 블로그로 전송하도록 설정할 수도 있다(<https://secure.del.icio.us/settings/user-id/blogging/posting>).

- ISBN
- Year, month, day
- 위도와 경도
- URL 자체. 예를 들어, <http://validator.w3.org?uri={uri-to-validate}>, 여기서 uri-to-validate 는 검증하려는 URL로 <http://validator.w3.org/check?uri=http%3A%2F%2Fvalidator.w3.org%2F>와 같은 값이다.

## 구글 맵

- 지도상에서 위치들을 검색할 수 있다.
- 지도상에서 회사들을 검색할 수 있다.
- 운전 시에 두 지점 간의 도로정보를 얻을 수 있다.
- My Maps 기능을 사용하여 여러분만의 지도를 생성할 수 있다.

## 구글 맵의 URL 언어

<http://maps.google.com/maps?f=q&hl=en&q=1600+Pennsylvania+Ave,+Washington,+DC&sl=36.60585,-121.858956&sspn=0.006313,0.01133&ie=UTF8&z=16&om=1&iwloc=addr>

---

8 <http://www.google.com/apis/maps/>

9 [http://news.yahoo.com/s/ap/20070405/ap\\_on\\_hi\\_te/google\\_maps](http://news.yahoo.com/s/ap/20070405/ap_on_hi_te/google_maps) - “검색코어 미디어 매트릭스에서 제공한 가장 최신 정보에 의하면 구글 맵은 이미 2월에만 미국 내 2천2백2십만의 사용자가 방문하는 최대의 서비스가 되었다. 이러한 결과로 구글 맵은 해당 분야에서 3위에 위치해 있으며 AOL의 맵퀘스트(4천5백10만 방문객)와 야후(2천9백10만 방문객)의 뒤를 추격하고 있다.

10 [http://mapki.com/index.php?title=Google\\_Map\\_Parameters&oldid=4145](http://mapki.com/index.php?title=Google_Map_Parameters&oldid=4145)의 내용이 [http://mapki.com/wiki/Google\\_Map\\_Parameters](http://mapki.com/wiki/Google_Map_Parameters)로 수정되었다



<http://maps.google.com/maps?q=1600+Pennsylvania+Ave,+Washington,+DC>

<http://maps.google.com/maps?q=1600+Pennsylvania+Ave,+Washington,+DC&z=0>

<http://maps.google.com/maps?q=1600+Pennsylvania+Ave,+Washington,+DC&z=17>

- mrad는 추가적인 목표지 주소를 명시할 수 있게 해준다.
- output=kml은 구글어스에 사용할 수 있는 KML파일을 생성해준다.
- layer=t는 교통상황을 나타내는 레이어를 추가한다.
- mrt=kmlkmz는 '사용자 생성 콘텐츠'를 보여준다. 예를 들어, 다음은 백악관 주위에 위치한 호텔에 대해서 사용자가 생성한 정보를 보여준다.

<http://maps.google.com/maps?f=q&hl=en&q=hotel&near=1600+Pennsylvania+Ave,+Washington,+DC&sl=36.60585,-121.858956&sspn=0.006313,0.01133&ie=UTF8&z=16&om=1&iwloc=addr&mrt=kmlkmz>

## 구글 맵의 KML 파일 보기

[http://services.google.com/earth/kmz/global\\_heritage\\_fund\\_n.kmz](http://services.google.com/earth/kmz/global_heritage_fund_n.kmz)

[http://maps.google.com/maps?q=http%2F%2Fservices.google.com%2Fearth%2Fkmz%2Fglobal\\_heritage\\_fund\\_n.kmz](http://maps.google.com/maps?q=http%2F%2Fservices.google.com%2Fearth%2Fkmz%2Fglobal_heritage_fund_n.kmz)

---

11 [http://mapki.com/wiki/Google\\_Map\\_Parameters](http://mapki.com/wiki/Google_Map_Parameters), accessed as <http://maps.google.com/maps?f=q&hl=en&q=1600+Pennsylvania+Ave,+Washington,+DC> on April 14, 2007

<http://maps.google.com/maps?q={URL-of-KML}>

## 야후!파이프와 구글 맵의 연결

[http://pipes.yahoo.com/pipes/pipe.info?location=94720&what=cafes&mindist=2&=Run+Pipe&\\_id=1mrlkB232xGjJDdwXqIxGw&\\_run=1](http://pipes.yahoo.com/pipes/pipe.info?location=94720&what=cafes&mindist=2&=Run+Pipe&_id=1mrlkB232xGjJDdwXqIxGw&_run=1)

[http://pipes.yahoo.com/pipes/pipe.run?\\_id=1mrlkB232xGjJDdwXqIxGw&\\_render=kml&\\_run=1&location=94720&mindist=2&what=cafes](http://pipes.yahoo.com/pipes/pipe.run?_id=1mrlkB232xGjJDdwXqIxGw&_render=kml&_run=1&location=94720&mindist=2&what=cafes)

[http://maps.google.com/maps?f=q&hl=en&geocode=&q=http%3A%2F%2Fpipes.yahoo.com%2Fpipes%2Fpipe.run%3F\\_id%3D1mrlkB232xGjJDdwXqIxGw%26\\_render%3Dkml%26\\_run%3D1%26location%3D94720%26mindist%3D2%26what%3Dcafes&ie=UTF8&ll=37.992916,-122.24556&spn=0.189398,0.362549&z=12&om=1](http://maps.google.com/maps?f=q&hl=en&geocode=&q=http%3A%2F%2Fpipes.yahoo.com%2Fpipes%2Fpipe.run%3F_id%3D1mrlkB232xGjJDdwXqIxGw%26_render%3Dkml%26_run%3D1%26location%3D94720%26mindist%3D2%26what%3Dcafes&ie=UTF8&ll=37.992916,-122.24556&spn=0.189398,0.362549&z=12&om=1)

## 구글 맵의 URL 언어를 이용한 기타 애플리케이션들



- `q={address}`를 사용하는 것만으로 해당 주소를 중심으로 한 지도를 생성할 수 있다는 것을 기억하라. 이보다 간단하게 특정 주소에 해당하는 지도를 생성하는 방법은 아마 생각해내기 어려울 것이다. 왜냐하면 주소를 지도상의 좌표로 바꾸어주는 과정이 필요 없기 때문이다.
- 주소가 정리된 스프레드시트에서 출발지 주소와 목표지 주소를 읽어 들여서 운전시에 참고할 수 있는 지도를 보여주는 URL을 생성할 수 있다. 예를 들면, Apress 출판사에서 컴퓨터 역사 박물관으로 가는 데 참고할 수 있는 지도는 아래 형식의 URL을 사용하면 된다.

`http://www.google.com/maps?saddr={source-address}&daddr={destination-address}`

`http://www.google.com/maps?saddr=2855+Telegraph+Ave,+Berkeley,+CA+94705&daddr=1401+N+Shoreline+Blvd,+Mountain+View,+CA+94043`

운전 시에 참고할 수 있는 지도를 생성하는 기능이 구글 맵 API에 최근에 추가되었지만<sup>12</sup>, `dirflg=h` 파라미터를 사용하여 고속도로 구간을 표시하지 않는 지도를 생성하는 것과 같은 기능은 API를 사용해서는 생성할 수 없다.<sup>13</sup> 그러므로, 여러분은 API를 사용하지 않고 아래 URL을 사용하여 Apress 출판사와 컴퓨터 역사 박물관 간에 고속도로를 경유하지 않는, 경치가 아름다운 노선을 다음과 같이 직접 생성할 수 있다.

`http://www.google.com/maps?saddr=2855+Telegraph+Ave,+Berkeley,+CA+94705&daddr=1401+N+Shoreline+Blvd,+Mountain+View,+CA+94043&dirflg=h`

- 프로그래머가 아니더라도 주소를 지도좌표로 변경하는 방법만으로 구글 맵을 사용할 수 있다. 계산하려는 지점의 위도와 경도를 나타내는 위치에 지도를 위치시키고 `ll` 파라미터 값을 추출하라. 만약 `ll` 파라미터 값이 존재하지 않는다면, 지도의 중심부분을 더블클릭하라. 그러면 해당 지점을 중심으로 지도가 다시 위치하게 될 것이다.

<sup>12</sup> [http://www.google.com/apis/maps/documentation/#Driving\\_Directions](http://www.google.com/apis/maps/documentation/#Driving_Directions)

<sup>13</sup> [http://groups.google.com/group/Google-Maps-API/browse\\_thread/thread/279ee413e4e0309/0dabfb71863af712?lnk=gst&q=avoid+highway&num=2#0dabfb71863af712](http://groups.google.com/group/Google-Maps-API/browse_thread/thread/279ee413e4e0309/0dabfb71863af712?lnk=gst&q=avoid+highway&num=2#0dabfb71863af712)

# 아마존

**\* NOTE!** 아마존이 책 이외의 다른 물품들도 판매하지만, 이 예제에서는 책을 사용하였다. 또한 미국 밖에 존재하는 고객을 위한 아마존 네트워크의 사이트 대신에 미국 내 고객을 대상으로 하는 사이트에 초점을 맞추었다.

## 아마존 아이템

<http://www.amazon.com/gp/product/{ASIN}>

<http://www.amazon.com/gp/product/0060514485>

<http://www.amazon.com/exec/obidos/ASIN/0060514485>

<http://amazon.com/o/ASIN/0060514485>

[http://amazon.com/s/ref=nb\\_ss\\_gw/102-1755462-2944952?url=search-alias%3Daps&field-keywords=flower&Go.x=0&Go.y=0](http://amazon.com/s/ref=nb_ss_gw/102-1755462-2944952?url=search-alias%3Daps&field-keywords=flower&Go.x=0&Go.y=0)

[http://amazon.com/s/ref=nb\\_ss\\_gw/102-8204915-1347316?url=search-alias%3Daps&field-keywords=flower&Go.x=0&Go.y=0&Go=Go](http://amazon.com/s/ref=nb_ss_gw/102-8204915-1347316?url=search-alias%3Daps&field-keywords=flower&Go.x=0&Go.y=0&Go=Go)

---

15 <http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=1>

16 <http://developer.amazonwebservices.com/connect/entry.jspa?externalID=703&categoryID=19>

17 <http://docs.amazonwebservices.com/AWSECommerceService/2007-04-04/DG/ItemIdentifiers.html>

<http://amazon.com/s/?url=search-alias%3Daps&field-keywords=flower&Go.x=0&Go.y=0&Go=Go>

<http://amazon.com/s/?url=search-alias%3Daps&field-keywords=flower>

[http://amazon.com/s/ref=nb\\_ss\\_gw/102-6984159-2338509?url=search-alias%3Dstripbooks&field-keywords=flower&Go.x=12&Go.y=6](http://amazon.com/s/ref=nb_ss_gw/102-6984159-2338509?url=search-alias%3Dstripbooks&field-keywords=flower&Go.x=12&Go.y=6)

<http://amazon.com/s/?url=search-alias%3Dstripbooks&field-keywords=flower>

<http://amazon.com/s/?url=search-alias%3Dgarden&field-keywords=flower>

<http://amazon.com/s/?url=search-alias%3D{amazon-dept}&field-keywords={keyword}>

## 리스트

<http://www.amazon.com/gp/registry/wishlist/>

[http://www.amazon.com/gp/registry/wishlist/1U5EXVPVS3WP5/ref=cm\\_wl\\_rlist\\_go/102-5889202-4328156](http://www.amazon.com/gp/registry/wishlist/1U5EXVPVS3WP5/ref=cm_wl_rlist_go/102-5889202-4328156)

<http://www.amazon.com/gp/registry/wishlist/1U5EXVPV3WP5>

<http://www.amazon.com/gp/registry/wishlist/{wishlist-id}>

## 태그

[http://www.amazon.com/tag/czeslaw%20milosz/ref=tag\\_dp\\_ct/102-8204915-1347316](http://www.amazon.com/tag/czeslaw%20milosz/ref=tag_dp_ct/102-8204915-1347316)

<http://www.amazon.com/tag/czeslaw%20milosz/>

<http://www.amazon.com/tag/{tag}/>

## 주제별 분류

[http://www.amazon.com/b/ref=amb\\_link\\_1760642\\_21/104-0367717-9318361?ie=UTF8&node=5&pf\\_rd\\_m=ATVPDKIKX0DER&pf\\_rd\\_s=center-3&pf\\_rd\\_r=0J0MADE0YSNIVRBA6XZS&pf\\_rd\\_t=101&pf\\_rd\\_p=233185601&pf\\_rd\\_i=1000](http://www.amazon.com/b/ref=amb_link_1760642_21/104-0367717-9318361?ie=UTF8&node=5&pf_rd_m=ATVPDKIKX0DER&pf_rd_s=center-3&pf_rd_r=0J0MADE0YSNIVRBA6XZS&pf_rd_t=101&pf_rd_p=233185601&pf_rd_i=1000)

<http://www.amazon.com/b/?ie=UTF8&node=5>

<http://www.amazon.com/b/?ie=UTF8&node={node-number}>

**▼ CAUTION!** 유일한 키 값이 아닌 목록 내에서의 알파벳 순서에 따라 부여되는 숫자값을 사용하여 노드 값이 명시된다는 사실 때문에 해당 링크가 오랜 시간이 지나도 유효할까라는 걱정을 하게 되었다. 5가 항상 컴퓨터 섹션을 가리킬까? 만약 알파벳 순서상 컴퓨터 섹션 앞에 다른 섹션이 추가되면 이 링크는 깨지게 될까?

■ 리스트매니아<sup>Listmania</sup> 목록

<http://www.amazon.com/lm/{list-mania-id}/>

<http://www.amazon.com/lm/1FH0E3G892IA/>

■ So You'd Like To 가이드

<http://www.amazon.com/gp/richpub/syltguides/fullview/{slltg-id}>

<http://www.amazon.com/gp/richpub/syltguides/fullview/3T3I3YDBG889B>

■ 사용자 프로필

<http://www.amazon.com/gp/pdp/profile/{user-id}/>

예를 들어

<http://www.amazon.com/gp/pdp/profile/A2D978B87TKMS2/>

■ 해당 책과 유사한 책들

[http://amazon.com/sim/{ISBN-10}/1/ref=pd\\_sexpl\\_esi/](http://amazon.com/sim/{ISBN-10}/1/ref=pd_sexpl_esi/)

예를 들어

[http://amazon.com/sim/0060514485/1/ref=pd\\_sexpl\\_esi/](http://amazon.com/sim/0060514485/1/ref=pd_sexpl_esi/)

# 딜리셔스

`http://del.icio.us/{user-id}`

`http://del.icio.us/rdhyee`

`http://del.icio.us/{user-id}/{tag}`

`http://del.icio.us/rdhyee/NYTimes`

`http://del.icio.us/tag/{tag}`

`http://del.icio.us/popular/{tag}`

`http://del.icio.us/popular/`

`http://del.icio.us/popular/?new`

<http://harpers.org/TheEcstasyOfInfluence.html>

<http://del.icio.us/url/53113b15b14c90292a02c24b55c316e5>

`md5.new("http://harpers.org/TheEcstasyOfInfluence.html").hexdigest()`

<http://del.icio.us/url?url=http://harpers.org/TheEcstasyOfInfluence.html>

<http://del.icio.us/url/53113b15b14c90292a02c24b55c316e5>

스크린-스크레이핑과 봇



- '채팅봇'은 인스턴트 메신저나 IRC를 통해 사용자에게 자동으로 응답한다.<sup>19</sup>
- 위키피디아 봇은 위키피디아의 모니터링, 유지보수, 그리고 수정작업을 자동화한다.<sup>20</sup>
- 티켓 구매 봇은 티켓판매상을 대신하여 티켓을 구매한다.
- 스팸을 생성하거나 분산된 서비스 거부 공격을 시작한다.

---

18 [http://en.wikipedia.org/w/index.php?title=Internet\\_bot&oldid=142845374](http://en.wikipedia.org/w/index.php?title=Internet_bot&oldid=142845374) 에서 [http://en.wikipedia.org/wiki/Internet\\_bot](http://en.wikipedia.org/wiki/Internet_bot) 으로 2007년 7월 11일에 수정됨

19 <http://en.wikipedia.org/wiki/Chatterbot>

20 <http://en.wikipedia.org/wiki/Wikipedia:Bots>

**! CAUTION!** 내가 법률가가 아니기 때문에, 다음의 논의를 포함하여 이 책의 어떠한 내용도  
법률적인 조언으로 여겨서는 안 된다.

[http://proquest.safaribooksonline.com/0596527063/I\\_0596527063\\_CHP\\_8\\_SECT\\_8](http://proquest.safaribooksonline.com/0596527063/I_0596527063_CHP_8_SECT_8)

- Webbots, Spiders, and Screen Scrapers, 마이클 슈렌크Michael Schrenk (No Starch Press, 2007)

---

21 폴 바우치(Paul Bausch), 타라 캘리스하인(Tara Calishain), 라엘 돈페스트(Rael Dornfest)가 집필한 Google Hacks 3판(오라일리, 2006)

- Spidering Hacks, 케빈 헤밍웨이|Kevin Hemenway, 타라 켈리스하인Tara Calishain (오라일리, 2003)

**\* NOTE!** 웹사이트 소유주들이 그들의 사이트를 확장가능하고 심지어는 해킹이 가능하도록 독려하는 최종 사용자 혁신에 관련한 몇몇 연구들이 최근에 진행되고 있다. 에릭 폰 히펠Eric Von Hippel의 책을 보도록 하라. 폰 히펠은 많은 제품과 혁신들이 최초에 제품의 제조사가 아닌 제품의 사용자에게 의해 이루어졌으며 제조사들은 이러한 혁신을 사실에 입각하여 가공하기만 하였다고 주장한다([http://en.wikipedia.org/wiki/Eric\\_Von\\_Hippel](http://en.wikipedia.org/wiki/Eric_Von_Hippel)).

## 요약



## **03**

---

### 태깅과 포크소너미에 대한 이해

---

■

■

■

■

■

태그에 의존하는 대신에 시각적인 검색을 할 수 있게 될까?

(<http://riya.com> 과 <http://www.riya.com/riyaAPI> 를 참고하라).

(<http://labs.systemone.net/retrievr/>)'

■

■

■

<http://www.flickr.com/photos/tags/>

<http://www.flickr.com/photos/tags/{tag}/>

<http://www.flickr.com/photos/tags/flower/>

<http://www.flickr.com/photos/tags/{tag}/interesting/>

<http://www.flickr.com/photos/tags/{tag}/clusters/>

<http://www.flickr.com/photos/tags/flower/clusters/>

<http://www.flickr.com/photos/{user-id}/tags/>

<http://www.flickr.com/photos/{user-id}/alltags/>

<http://www.flickr.com/photos/{user-id}/tags/{tag}/>



## 태그가 실제로 어떻게 사용되는가

- july4(예를 들어, <http://www.flickr.com/photos/tags/july4/>)
- fourthofjuly(예를 들어, <http://www.flickr.com/photos/tags/fourthofjuly/>)
- july4th(예를 들어, <http://www.flickr.com/photos/tags/july4th/>)
- july04(예를 들어, <http://www.flickr.com/photos/tags/july04/>)
- july4th2007(예를 들어, <http://www.flickr.com/photos/tags/july4th2007/>)

---

2 <http://www.flickr.com/groups/central/discuss/2026/> 과 <http://www.flickr.com/groups/central/discuss/2730/> 을 참고하라.

---

3 <http://www.flickr.com/services/api/misc.tags.html> 은 '클린' 버전의 태그와 '원본' 버전의 태그에 대한 차이점을 설명한다.

---

4    이전 버전의 <http://en.wikipedia.org/w/index.php?title=Folksonomy&oldid=145985651>    <sup>o</sup> <http://en.wikipedia.org/wiki/Folksonomy> 로 수정되었다.



---

5 <http://www.flickr.com/forums/bugs/31668/> 의 글은 타이틀과 설명에 대해서 어근분석을 사용한다는 플리커 운영자의 글을 포함하고 있다. <http://tech.groups.yahoo.com/group/yws-flickr/message/1913> 은 태그 문맥 내에서의 어근분석에 대해 언급하고 있다. <http://www5.flickr.mud.yahoo.com/help/forum/37259/#reply211324> 는 왜 이러한 일이 발생하는지 설명한다.

---

6 <http://base.google.com/support/bin/answer.py?answer=27882>

7 <http://docs.amazonwebservices.com/AmazonS3/2006-03-01/BasicObjects.html> 과 <http://docs.amazonwebservices.com/AmazonS3/2006-03-01/RESTObjectPUT.html> 을 참조하여 사용자 메타데이터(이름/값의 짝)을 추가할 수 있다.

## 플리커 태그를 사용하는 흥미로운 애플리케이션들

- 플리커 연관 태그 브라우저(Flickr Related Tag Browser, [http://www.airtightinteractive.com/projects/related\\_tag\\_browser/app/](http://www.airtightinteractive.com/projects/related_tag_browser/app/))는 관련이 있는 태그들 사이의 관계를 브라우징할 수 있게 해준다.
- findr(<http://www.forestandthetrees.com/findr/findr.html>)는 서로 관련된 태그들을 보여주며 관련 태그를 조합한 태그로 태깅된 사진을 보여준다.
- fastr(<http://randomchaos.com/games/fastr/>)는 여러분에게 제시된 사진을 보고 태그를 추측하는 게임이다.
- ZoneTag(<http://zonetag.research.yahoo.com/>)는 휴대폰 카메라로 촬영된 사진의 위치 데이터를 추가하기 위해 플리커 태그를 해킹한 예제이다.
- TagMaps(<http://tagmaps.research.yahoo.com/>)는 지역을 기준으로 위치정보가 태깅된 플리커 사진과 서로 연관된 인기 태그를 지도 위에 보여준다.

## 딜리셔스에서의 태깅

## 딜리셔스에서 태그를 추가하는 기법

1. <http://del.icio.us/post/>에 접속하여 URL(예를 들어, <http://www.rubyonrails.org/>)을 입력하고 Save 버튼을 클릭하도록 하자.

## 대소문자 구분과 여러 단어로 이루어진 문장의 처리

[http://del.icio.us/search/?fr=del\\_icio\\_us&p=San+Francisco&type=all](http://del.icio.us/search/?fr=del_icio_us&p=San+Francisco&type=all)

[http://del.icio.us/search/?fr=del\\_icio\\_us&p=tag%3Asanfrancisco&type=user](http://del.icio.us/search/?fr=del_icio_us&p=tag%3Asanfrancisco&type=user)

■

(<http://del.icio.us/tag/SanFrancisco>)

■

■





## 보다 많은 정보 얻기

### latin-8 과 유니코드 문자의 표시

먼저 태그가 latin-8 문자(예를 들어, 프랑스어의 français)에 대해서 어떻게 동작하는지 알아보고 그 후에 중국어에 대해서 알아볼 것이다.

#### 플리커에서

<http://flickr.com/photos/tags/fran%C3%A7ais/>에 접속해보도록 하자. 플리커는 français를 francais로 축약하지 않는다. <http://flickr.com/photos/raymondye/368644336/>의 사진을 보면 내가 français와 francais를 모두 태그로 사용하는 것을 볼 수 있다. 즉, API를 호출해보면\*1 확인할 수 있지만 두 개의 태그는 서로 유일한 태그이다.

<http://flickr.com/photos/raymondye/tags/%E4%BD%99%E4%BF%8A%E9%9B%84/>

```
import urllib
print urllib.unquote('%E4%BD%99%E4%BF%8A%E9%9B%84').decode('utf-8')u'\u4f59\u4fca\u96c4'
```

\*1 [http://api.flickr.com/services/rest/?method=flickr.tags.getListPhoto&api\\_key={api-key}&photo\\_id=368644336](http://api.flickr.com/services/rest/?method=flickr.tags.getListPhoto&api_key={api-key}&photo_id=368644336)

\*2 <http://flickr.com/photos/raymondye/79915850> 과 <http://flickr.com/photos/raymondye/tags/%E4%BD%99%E4%BF%8A%E9%9B%84/>

<http://flickr.com/photos/raymondyeetags/%E4%BD%99%E4%BF%8A%E9%9B%84/>

### 딜리셔스에서

프랑스-버클리(France-Berkeley) 프로그램에 대한 URL을 등록하였다.\*4

<http://del.icio.us/tag/fran%C3%A7ais>은 français를 풀텍스트 검색으로 검색한 결과뿐만 아니라 français를 태그로 등록된 많은 링크를 출력한다.\*5

```
>>> print chr(231)
ç
>>> print urllib.urlencode({'q':chr(231).decode('ISO-8859-1').encode('utf-8')})
q=%C3%A7
```

[http://www.youtube.com/results?search\\_query=%E4%BD%99%E4%BF%8A%E9%9B%84](http://www.youtube.com/results?search_query=%E4%BD%99%E4%BF%8A%E9%9B%84)

### rel-tag 에서

```
<a href="http://technorati.com/tag/Sant%C3%A9+et+bien-%C3%AAtre" rel="tag">
Santé et bien-être</a>
```

\*3 [http://www.cojak.org/index.php?function=code\\_lookup&term=4F59](http://www.cojak.org/index.php?function=code_lookup&term=4F59) 와 <http://www.unicode.org/cgi-bin/GetUniHanData.pl?codepoint=4F59>

\*4 <http://del.icio.us/url?url=http://ies.berkeley.edu/fbf/>

\*5 [http://del.icio.us/search/?fr=del\\_icio\\_us&p=fran%C3%A7ais&type=all](http://del.icio.us/search/?fr=del_icio_us&p=fran%C3%A7ais&type=all)

\*6 <http://del.icio.us/url?url=http://flickr.com/photos/raymondyeetags/79915850/>

\*7 <http://del.icio.us/tag/%E4%BD%99%E4%BF%8A%E9%9B%84>

\*8 [http://del.icio.us/search/?fr=del\\_icio\\_us&p=%E4%BD%99%E4%BF%8A%E9%9B%84&type=all](http://del.icio.us/search/?fr=del_icio_us&p=%E4%BD%99%E4%BF%8A%E9%9B%84&type=all)

\*9 [http://microformats.org/wiki/rel-tag#Encoding\\_issues](http://microformats.org/wiki/rel-tag#Encoding_issues) 로 <http://microformats.org/wiki?title=rel-tag&diff=0&oldid=18625> 이 수정되었음

```
import urllib
s = "Santé et bien-être"
u = s.decode('iso-8859-1')
print urllib.urlencode({'q':u.encode('utf8')})
```

위 코드는 아래를 문자열을 반환한다.

```
q= Sant%C3%A9+et+bien-%C3%AAtre
http://technorati.com/tag/Sant%C3%A9+et+bien-%C3%AAtre
```

이것은 Santé et bien-être 를 검색한 것이다.

## 테크노라티 태그를 통해 콘텐츠 수집하기

### 태그를 사용하여 테크노라티에서 검색하기

<http://technorati.com/tag/mashup>

<http://technorati.com/tag/{tag}>

- 블로그 포스트(<http://technorati.com/posts/tag/{tag}>)
- 비디오(<http://technorati.com/videos/tag/{tag}>)
- 사진(<http://technorati.com/photos/tag/{tag}>)
- 블로그(<http://technorati.com/blogs/tag/{tag}>)

<http://technorati.com/pop/>

## 테크노라티는 어떻게 웹에서 태그를 찾아낼까

■ [HTML과 XHTML 태그에 포함된 리퍼코더\(리노\)에 대한 보다 자세한 내용은 4장을 보라.](#)

- `<a href="http://technorati.com/tag/{tagname}" rel="tag">tagname</a>`와 같이 마이크로포맷인 rel-tag를 사용하여 링크에 포함된 태그들(완전한 설명은 18장을 참조하라).
- 
- 

## 테크노라티 태그에 있어 단어의 어형변화와 언어상의 제약

<http://technorati.com/tag/mouse>

<http://technorati.com/tag/mice>

<http://technorati.com/tag/san+francisco>

<http://technorati.com/tag/san-francisco>

태그를 사용하여 플리커와 딜리셔스를 매쉬업하기

<http://del.icio.us/rdhyee/set:Berkeley>

## 태깅을 사용하는 기타 시스템

- 
- 
- 
- 
- 

## 정규적인 분류 체계와 태그의 관계

---

9 <http://en.wikipedia.org/wiki/Tags>

10 <http://www.neato.co.nz/ultimate-tag-warrior/>

[http://catalog.loc.gov/cgi-bin/Pwebrecon.cgi?v3=1&DB=local&CMD=kisn+006019667X  
&CNT=10+records+per+page](http://catalog.loc.gov/cgi-bin/Pwebrecon.cgi?v3=1&DB=local&CMD=kisn+006019667X&CNT=10+records+per+page)

<http://catalog.loc.gov/cgi-bin/Pwebrecon.cgi?SC=Subject&SA=Mi%C5%82osz%C5%82aw%C5%82aw%20Translations%20into%20English>

<http://worldcatlibraries.org/wcpa/isbn/006019667X>

<http://worldcatlibraries.org/search?q=su%3AMi%C5%82osz%C5%82aw%C5%82aw>

<http://www.oclc.org/research/projects/fast/>

<http://www.worldcat.org/search?q=su%3AMi%C5%82osz%C5%82aw%C5%82aw>

<http://www.librarything.com/isbn/{isbn}>

<http://www.librarything.com/work/{librarything-work-id}>

<http://www.librarything.com/isbn/006019667X>

<http://www.librarything.com/work/3532375>

<http://www.librarything.com/work-info/{librarything-work-id}>

<http://www.librarything.com/work-info/161671>

<http://www.librarything.com/subject.php?subject=Milosz%2C+Czeslaw%09Translations+into+English>

**▼ CAUTION!** 라이브러리썬에는 Miłosz, Czesław가 잘못 표시되는 문자 인코딩 관련 오류가 있다.<sup>11</sup>

---

<sup>11</sup> <http://www.librarything.com/talktopic.php?topic=12559#138896>



## 요약



## 04

---

### 피드, RSS, 그리고 Atom으로 작업하기

---

- 
- 
- 
- 
-

■

■

- 어떻게 피드와 피드버너(<http://www.feedburner.com>), 야후!파이프를 리믹스하고 매쉬업하는가

**\* NOTE!**

<http://>

[www.w3schools.com/xml/](http://www.w3schools.com/xml/)

피드가 무엇이며 이것이 왜 중요한가?

■

■

■

## RSS2.0

<http://cyber.law.harvard.edu/rss/rss.html>

■

■

■

■

■

■

RSS1.0



<http://web.resource.org/rss/1.0/spec>

<http://web.resource.org/rss/1.0/schema.rdf>

---

2 <http://www.w3.org/2001/sw/>

3 [http://examples.mashupguide.net/ch04/RSS1.0\\_Apress.xml](http://examples.mashupguide.net/ch04/RSS1.0_Apress.xml)

■

- RSS1.0은 RDF와 관련된 XML 네임스페이스(<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)를 사용하며, RSS1.0에 관련된 디폴트 네임스페이스를 사용하여 <channel>, <item>, <title>과 같은 모든 엘리먼트들을 네임스페이스에 위치시킨다.

■

■

■



- <http://rss.slashdot.org/Slashdot/slashdot>
- [http://www.nature.com/nature/current\\_issue/rss/index.html](http://www.nature.com/nature/current_issue/rss/index.html)(<http://www.nature.com/webfeeds/index.html> 에서 추출된 목록)
- <http://www.w3.org/2000/08/w3c-synd/home.rss>
- <http://simile.mit.edu/blog/?feed=rdf>

**\* NOTE!** RSS를 업데이트하려는 노력들이 진행되고 있다. RSS2.0은 RSS 자문위원회(<http://www.rssboard.org/>)가 업데이트를 위해 설계를 진행해오고 있는 반면 RSS1.1(<http://inamidst.com/rss1.1/>)은 RSS1.0을 향상시키기 위한 소수의 개발자들에 의해 탄생되었다. RSS2.0과 RSS1.0은 두 가지 계열의 주요 명세들 중에서 가장 중요한 버전으로 남아있다.

## Atom1.0

<http://tools.ietf.org/html/rfc4287>



- Atom 관련 기본 XML 네임스페이스(<http://www.w3.org/2005/Atom>)를 사용한 것에 주목하라.
- RSS에서 사용하는 <description> 대신에 <subtitle> 엘리먼트를 사용하여 피드에 대한 내용을 설명한다.
- <feed>와 <entry> 엘리먼트는 모두 반드시 <updated> 엘리먼트를 포함해야 한다.
- <link rel="alternate" type="text/html" href="http://www.apress.com/" />은 이 문서가 웹 페이지 <http://www.apress.com/>을 다른 포맷으로 표현(즉, 피드를 말한다)한다는 것을 나타낸다.
- <link rel="self" href="http://examples.mashupguide.net/ch04/Atom1.0\_Apress.xml/" />은 이 피드 문서의 위치를 나타낸다.
- <summary> 엘리먼트 내의 type="html" 어트리뷰트는 엔티티가 인코딩된 HTML을 사용하였다는 것을 나타낸다.

[http://en.wikipedia.org/wiki/Atom\\_%28standard%29#Atom\\_Compared\\_to\\_RSS\\_2.0](http://en.wikipedia.org/wiki/Atom_%28standard%29#Atom_Compared_to_RSS_2.0)

<http://atompub.org/rfc4287.html#schema>

## RSS2.0과 Atom1.0에 대한 확장기능

**\* NOTE!** 여기서는 RSS1.0의 확장성에 대해서는 다루지 않으며 RSS1.0 모듈(<http://web.resource.org/rss/1.0/modules/>)과 더블린 코어(<http://web.resource.org/rss/1.0/modules/dc/>), 신디케이션(<http://web.resource.org/rss/1.0/modules/syndication/>), 그리고 콘텐츠(<http://web.resource.org/rss/1.0/modules/content/>)와 같은 표준 모듈들에 대해 연구하는 것은 독자들의 몫으로 남겨둔다.

---

5 외부 마크업은 특정한 문맥에 의해 명시적으로 금지되지 않는 한 허용된다. 좀 더 자세한 내용은 <http://tools.ietf.org/html/rfc4287#section-6>을 참고하라.

6 [http://examples.mashupguide.net/ch04/Atom1.0\\_Apress\\_ISBN.xml](http://examples.mashupguide.net/ch04/Atom1.0_Apress_ISBN.xml)

- 플리커에서 사용되는 Media RSS(<http://search.yahoo.com/mrss>)
- 포드캐스팅podcasting에 사용되는 iTunes(<http://www.apple.com/itunes/store/podcaststechspecs.html>)
- 오픈서치(<http://www.opensearch.org/Specifications/OpenSearch/1.1>)



## 플리커의 피드

<http://flickr.com/groups/central/>

[http://api.flickr.com/services/feeds/groups\\_discuss.gne?id=34427469792@N01&lang=en-us&format=rss\\_200](http://api.flickr.com/services/feeds/groups_discuss.gne?id=34427469792@N01&lang=en-us&format=rss_200)

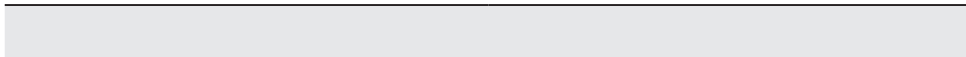
---

7 <http://www-128.ibm.com/developerworks/xml/library/x-atom10.html>, <http://www-128.ibm.com/developerworks/xml/library/x-extatom1/>, 그리고 <http://www-128.ibm.com/developerworks/xml/library/x-extatom2.html>



<http://www.flickr.com/services/feeds/>

## 플리커 피드의 파라미터





## 플리커 피드에 대한 검토

[http://api.flickr.com/services/feeds/photos\\_public.gne](http://api.flickr.com/services/feeds/photos_public.gne)

- id={사용자의 nsid}
- ids={콤마로 구분된 여러 개의 사용자 nsid}
- tags={콤마로 구분된 여러 개의 태그}
- tagmode={ all(기본값) 또는 any }

[http://api.flickr.com/services/feeds/photos\\_public.gne?tags=flower%2CBerkeley&format=rss2&lang=fr-fr](http://api.flickr.com/services/feeds/photos_public.gne?tags=flower%2CBerkeley&format=rss2&lang=fr-fr)

[http://api.flickr.com/services/feeds/photos\\_friends.gne?user\\_id={user-nsid}](http://api.flickr.com/services/feeds/photos_friends.gne?user_id={user-nsid})

- `display_all`은 기본값으로 설정되면 친구당 하나의 사진을 보여주지만 친구당 복수의 사진을 보여주도록 1로 설정될 수도 있다.
- `friends`는 요청한 사용자의 가족과 친구의 사진만을 보여주도록 1로 설정될 수 있다.

[http://api.flickr.com/services/feeds/photos\\_friends.gne?user\\_id=48600101146@N01&friends=0&display\\_all=0&lang=en-us&format=atom\\_1](http://api.flickr.com/services/feeds/photos_friends.gne?user_id=48600101146@N01&friends=0&display_all=0&lang=en-us&format=atom_1)

[http://api.flickr.com/services/feeds/groups\\_discuss.gne?id={group-nsid}](http://api.flickr.com/services/feeds/groups_discuss.gne?id={group-nsid})

[http://api.flickr.com/services/feeds/groups\\_pool.gne?{group-nsid}](http://api.flickr.com/services/feeds/groups_pool.gne?{group-nsid})

<http://api.flickr.com/services/feeds/forums.gne>

<http://api.flickr.com/services/feeds/activity.gne?user-id={user-nsid}>



[http://api.flickr.com/services/feeds/photos\\_comments.gne?user-id={user-nsid}](http://api.flickr.com/services/feeds/photos_comments.gne?user-id={user-nsid})

<http://api.flickr.com/services/feeds/news.gne>

■

■

- <http://purl.org/dc/elements/1.1/> 네임스페이스와 관련된 dc 접두어(예를 들어, <dc:date.Taken>)는 더블린 코어(DC; Dublin Core) 메타데이터 표준을 나타낸다. DC는 폭넓은 범위의 디지털 콘텐츠에 적용되도록 설계된 메타데이터의 핵심부분을 가리킨다.
- RSS2.0의 <enclosure> 엘리먼트를 보완해주는 Media RSS(<http://search.yahoo.com/mrss>)는 플리커에서 다음과 같은 사진관련 메타데이터를 저장하기 위해 사용된다.

.  
.   
.   
.   
.   
.   
.

[http://api.flickr.com/services/feeds/photos\\_public.gne?tags=tree&format=atom](http://api.flickr.com/services/feeds/photos_public.gne?tags=tree&format=atom)

## RSS와 Atom 외의 데이터 교환 포맷

[http://api.flickr.com/services/feeds/photos\\_public.gne?tags=tree](http://api.flickr.com/services/feeds/photos_public.gne?tags=tree)

## RSS0.91 과 RSS0.92

[http://api.flickr.com/services/feeds/photos\\_public.gne?tags=tree&format=rss\\_091](http://api.flickr.com/services/feeds/photos_public.gne?tags=tree&format=rss_091)

[http://api.flickr.com/services/feeds/photos\\_public.gne?tags=tree&format=rss\\_092](http://api.flickr.com/services/feeds/photos_public.gne?tags=tree&format=rss_092)

## JSON

[http://api.flickr.com/services/feeds/photos\\_public.gne?tags=tree&format=json](http://api.flickr.com/services/feeds/photos_public.gne?tags=tree&format=json)

## 다른 피드 포맷들

- 

- 

- 

- 

## 다른 웹사이트의 피드

## 피드를 찾아내고 자동발견하기

■

■

<http://www.syndic8.com/>

---

9 <http://code.google.com/apis/ajaxfeeds/>

10 <http://www.xml.com/pub/a/2004/02/11/googlexml.html>

- rel은 alternate 값으로 설정된다.
- href는 피드의 URL이다.
- type은 피드의 MIME 타입(application/rss+xml 또는 application/atom+xml)으로 설정된다.
- title은 선택항목이며 피드의 제목으로 설정된다.

<http://news.yahoo.com>

```
<link rel="alternate" type="application/rss+xml" title="Yahoo! News - Top Stories"
href="http://rss.news.yahoo.com/rss/topstories" />
```

#### 피드 자동발견 기능에 대한 공식적인 표준이 있는가?

비록 피드 자동발견이 폭 넓게 구현되어 있긴 하지만, 현재 합법적인 표준이 마련되어 있지는 않다. 피드 자동발견은 블로그([http://diveintomark.org/archives/2002/06/02/important\\_change\\_to\\_the\\_link\\_tag](http://diveintomark.org/archives/2002/06/02/important_change_to_the_link_tag))를 통해 수행되는 공동작업으로 시작되어 IETF의 초안(가장 마지막으로 소멸된 버전은 <http://www.ietf.org/internet-drafts/draft-snell-atompublish-autodiscovery-00.txt> 이었다.)으로 검토되었으며 현재 HTML5(<http://www.whatwg.org/specs/web-apps/current-work/#alternate>)의 일부로서 표준안으로 고려되고 있다.

한편에서는, 현재 사용되는 피드 자동발견에 대한 몇몇 내용이 <http://www.feedautodiscovery.org/doku.php>와 같은 위키에 문서화되고 있다.

## 블로그로부터의 피드

- Blogspot 블로그는 Atom 피드<sup>11</sup> (예를 들어, <http://googleblog.blogspot.com/atom.xml>과 <http://googleblog.blogspot.com/feeds/posts/default>)를 제공한다.
- WordPress 블로그<sup>12</sup> (예를 들면, <http://blog.mashupguide.net/feed/> 와 <http://blog.mashupguide.net/feed/atom/>)
- TypePad 블로그는 피드를 지원한다.<sup>13</sup>

## 위키피디아 피드

- <http://meta.wikimedia.org/wiki/RSS>
- <http://en.wikipedia.org/wiki/Wikipedia:Syndication>

<http://en.wikipedia.org/w/index.php?title={page-name}&action=history&feed={format}>

[http://en.wikipedia.org/w/index.php?title=Hurricane\\_Katrina&action=history&feed=atom](http://en.wikipedia.org/w/index.php?title=Hurricane_Katrina&action=history&feed=atom)

[http://en.wikipedia.org/w/index.php?title=Mashup\\_%28web\\_application\\_hybrid%29&action=history&feed=atom](http://en.wikipedia.org/w/index.php?title=Mashup_%28web_application_hybrid%29&action=history&feed=atom)

---

<sup>11</sup> 원문에는 <http://help.blogger.com/bin/topic.py?topic=8927>로 기재되어 있으나 실제로는 링크가 사라졌으며 <http://help.blogger.com/bin/topic.py?topic=12455> 이 유사할 것으로 보인다.

<sup>12</sup> [http://codex.wordpress.org/WordPress\\_Feeds](http://codex.wordpress.org/WordPress_Feeds)

<sup>13</sup> [http://support.typepad.com/cgi-bin/typepad.cfg/php/enduser/std\\_adp.php?p\\_faqid=86](http://support.typepad.com/cgi-bin/typepad.cfg/php/enduser/std_adp.php?p_faqid=86)



<http://en.wikipedia.org/wiki/Special:Recentchanges?feed={format}>

<http://en.wikipedia.org/wiki/?Special:Newpagesfeed={format}>

<http://feeds.feedburner.com/WikinewsLatestNews>

<http://en.wikipedia.org/w/api.php?action=feedwatchlist&feedformat={format}>

## 구글과 야후! 뉴스

[http://news.google.com/intl/en\\_us/new\\_feed\\_terms.html](http://news.google.com/intl/en_us/new_feed_terms.html)

<http://news.google.com/news?ned=us&topic={topic}&output={format}>



---

`http://news.google.com/news?ned=us&topic=h&output=rss`

`http://news.google.com/news?ned={region}&topic=n&output={format}`

---

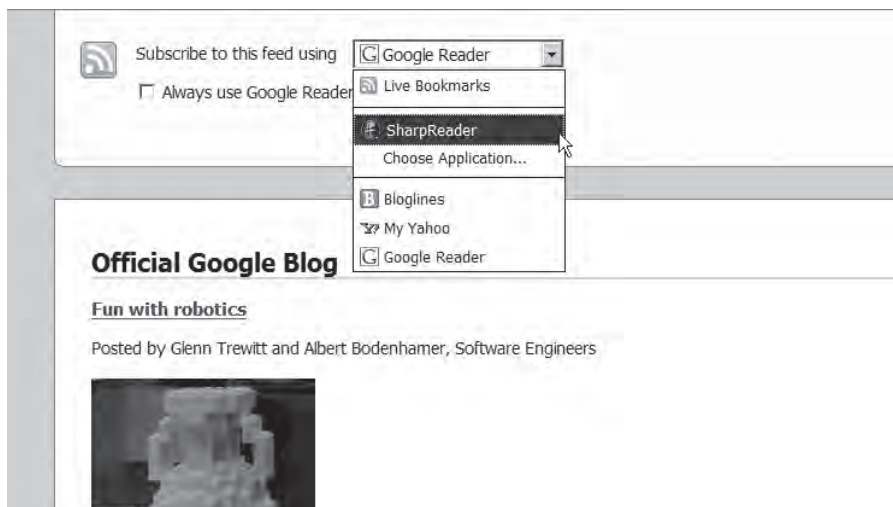
`http://news.google.com/news?q={query}&output={output}`

`http://news.google.com/news?q=mashup&output=rss`

<http://news.yahoo.com/rss>

<http://news.search.yahoo.com/news/rss?p=Hurricane+Katrina>

## 뉴스 수집기:플리커 피드를 다른 곳에서 보여주기



- 파이어폭스 라이브 북마크를 사용하면 파이어폭스의 북마크로 피드를 관리할 수 있다.<sup>14</sup> 이 기능은 사파리<sup>15</sup>와 인터넷 익스플로러<sup>16</sup>에 있는 기능과 유사해 보인다.
- 블로그라인(<http://www.bloglines.com/>)
- 윈도우용 데스크톱 기반 RSS 수집기/뉴스 리더기인 샤프리더(<http://www.sharpreader.net/>)
- 맥용 데스크톱 기반 뉴스 리더기인 넷뉴스와이어(<http://www.newsagtor.com/NGOLProduct.aspx?ProdID=NetNewsWire>)
- 구글리더(<http://www.google.com/reader/view/>)
- 마이 야후!에서는 <http://my.yahoo.com>에 RSS 피드를 추가할 수 있다. 예를 들면, RSS 피드를 <http://e.my.yahoo.com/config/cstore?.opt=rss&.page=p1> 이란 URL에 등록할 수 있다. 보다 자세한 정보는 다음을 참고하도록 하라.
  - <http://my.yahoo.com/s/faq/rss/>
  - <http://publisher.yahoo.com/rssguide>

## 피드의 검증

- <http://feedvalidator.org>는 온라인 서비스이며 또한 로컬에서도 실행할 수 있는 소프트웨어이다.<sup>17</sup>
- <http://validator.w3.org/feed/>는 <http://feedvalidator.org>에서 구할 수 있는 소프트웨어를 기반으로 개발된 W3C의 서비스이며 문법검사 기능은 SOAP 기반 웹서비스로도

<sup>14</sup> <http://www.mozilla.com/en-US/firefox/livebookmarks.html>

<sup>15</sup> <http://www.apple.com/macosx/features/safari/>

<sup>16</sup> <http://www.microsoft.com/windows/rss/default.msp>

<sup>17</sup> [http://feedvalidator.org/docs/howto/install\\_and\\_run.html](http://feedvalidator.org/docs/howto/install_and_run.html)

사용할 수 있다.<sup>18</sup>

- RSS1.0을 다룰 때는, W3C RDF 검증 서비스가 유용하다(<http://www.w3.org/RDF/Validator/>).

## GUI 툴을 사용하여 피드를 스크레이핑하기

<http://programmableweb.com/apis/directory>

<http://feedity.com/?http://programmableweb.com/apis/directory%40%40%40CAT%40%40%406>

---

<sup>18</sup> <http://validator.w3.org/feed/docs/soap>

<http://service.openkapow.com/rdhyee/programmablewebapis.rss>

## 피드를 피드버너와 리믹스하기

1. 회원가입을 하고 <http://www.feedburner.com/fb/a/myfeeds>에 접속하도록 하자. 여기서 피드의 URL에 블로그의 URL인 <http://blog.mashupguide.net> 을 대신 입력하였다.
2. 피드버너는 피드 자동발견 메커니즘을 통해 블로그에 관련된 다섯 개의 피드 중에서 하나를 선택하라는 입력창을 보여준다. 여기서는 Mashup Guide Atom Feed(<http://blog.mashupguide.net/feed/atom/>)을 선택하였다.
3. 기본설정으로 제공되는 제목(Mashup Guide)과 주소(<http://feeds.feedburner.com/MashupGuide>)는 그대로 사용하기로 하였다.

■

[http://www.w3.org/2005/08/online\\_xslt/xslt?xslfile=http%3A%2F%2Ffeeds.feedburner.com%2F%7Ed%2Fstyles%2Fatom10full.xsl&xmlfile=http%3A%2F%2Ffeeds.feedburner.com%2FMashupGuide+&content-type=&submit=transform](http://www.w3.org/2005/08/online_xslt/xslt?xslfile=http%3A%2F%2Ffeeds.feedburner.com%2F%7Ed%2Fstyles%2Fatom10full.xsl&xmlfile=http%3A%2F%2Ffeeds.feedburner.com%2FMashupGuide+&content-type=&submit=transform)

- 
- 
- 
- 
- 

야후!파이프와 피드를 리믹스하기



<http://pipes.yahoo.com/pipes/docs>

<http://pipes.yahoo.com/pipes/docs?doc=modules>

- 야후!뉴스
- 구글뉴스
- 위키뉴스
- 뉴욕 타임즈의 국내정보 섹션



## 야후!뉴스를 사용하는 첫 번째 파이프

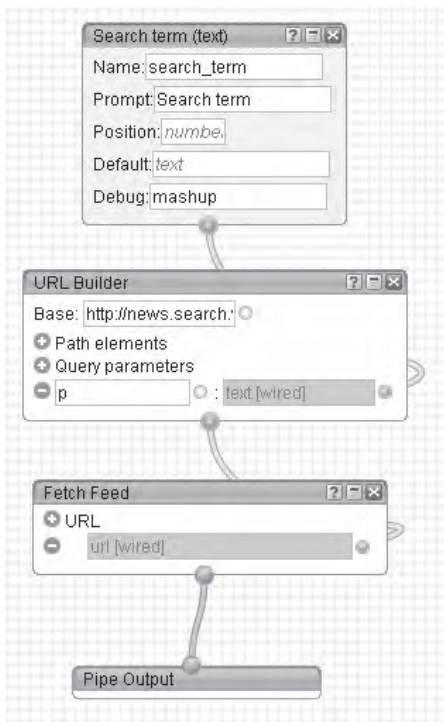
`http://news.search.yahoo.com/news/rss?p={search-term}`

`http://pipes.yahoo.com/pipes/pipe.info?_id=Rg_rh3NA3BGdECIel7okhQ`

`http://pipes.yahoo.com/pipes/pipe.run?_id={pipe-id}&_render=rss&search_term={search-term}`

`http://pipes.yahoo.com/pipes/pipe.run?_id=Rg_rh3NA3BGdECIel7okhQ&_render=rss&search_term=Hurricane+Katrina`

- 사용자로부터 검색어를 입력 받고 이 검색어를 다음에 설명할 URL Builder에 넘겨주기 위해 URL Builder와 연결되는 Text Input 모듈
- Base 파라미터로 `http://news.search.yahoo.com/news/rss`를 갖고 쿼리 파라미터로 `p`를 갖는 URL Builder 모듈
- URL Builder로부터 전달 받은 URL의 피드를 가져오는 Fetch Feed 모듈



## 구글 뉴스와 파이프를 리팩토링하기

<http://news.google.com/news?q={search-term}&output=atom>

`{base-URL}?{parameter_name}={parameter_value}{URL_suffix}`

[http://pipes.yahoo.com/pipes/pipe.info?\\_id=VoLceXZA3BGkqcJZJxOy0Q](http://pipes.yahoo.com/pipes/pipe.info?_id=VoLceXZA3BGkqcJZJxOy0Q)



[http://pipes.yahoo.com/pipes/pipe.info?\\_id=5NhmMndA3BGg5zQ5n0artA](http://pipes.yahoo.com/pipes/pipe.info?_id=5NhmMndA3BGg5zQ5n0artA)

[http://pipes.yahoo.com/pipes/pipe.info?\\_id=OKWv6nNA3BGkPtA8qWIyXQ](http://pipes.yahoo.com/pipes/pipe.info?_id=OKWv6nNA3BGkPtA8qWIyXQ)

**위키뉴스와 뉴욕 타임즈: 피드 필터링하기**

<http://feeds.feedburner.com/WikinewsLatestNews>

<http://www.nytimes.com/services/xml/rss/nyt/National.xml>

[http://pipes.yahoo.com/pipes/pipe.info?\\_id=KIYSv3pA3BGgloVbCB2yXQ](http://pipes.yahoo.com/pipes/pipe.info?_id=KIYSv3pA3BGgloVbCB2yXQ)

[http://pipes.yahoo.com/pipes/pipe.info?\\_id=PA7iqHpA3BGb0AiVX0sBXw](http://pipes.yahoo.com/pipes/pipe.info?_id=PA7iqHpA3BGb0AiVX0sBXw)

[http://pipes.yahoo.com/pipes/pipe.info?\\_id=yhBh7HxA3BGU\\_YRj1vC6Jw](http://pipes.yahoo.com/pipes/pipe.info?_id=yhBh7HxA3BGU_YRj1vC6Jw)

## 피드들을 함께 가져오기

- 검색어를 입력 받아 각각의 개별적인 뉴스 제공자의 파이프에 전달한다.
- Union 모듈을 사용하여 각각의 피드를 연결한다.
- Unique 모듈을 사용하여 동일한 링크를 가진 피드 아이템을 제거한다.

[http://pipes.yahoo.com/pipes/pipe.info?Search\\_term=Hurricane+Katrina&\\_cmd=Run+Pipe&\\_id=qlUkcn1A3BGeWlNQjknRlg&\\_run=1](http://pipes.yahoo.com/pipes/pipe.info?Search_term=Hurricane+Katrina&_cmd=Run+Pipe&_id=qlUkcn1A3BGeWlNQjknRlg&_run=1)

[http://pipes.yahoo.com/pipes/pipe.run?Search\\_term=Hurricane+Katrina&\\_id=qlUkc1A3BGeWlNQjknRlg&render=rss](http://pipes.yahoo.com/pipes/pipe.run?Search_term=Hurricane+Katrina&_id=qlUkc1A3BGeWlNQjknRlg&render=rss)

<http://pipes.yahoo.com/pipes/tag.info?namespace=user&tag=mashupguide>



# 05

---

## 블로그와 통합하기

---

■

■

■

- 

- 

## 블로그 통합 시나리오

- 

- 

-



### 블로깅을 시작할 때 사용할 수 있는 무료 블로그 호스팅 서비스

아직 블로그를 사용하지 않고 있다면, 실습용 계정을 설정하자. 블로깅 소프트웨어를 다운로드해서, 글을 작성하거나 또는 블로그 호스팅서비스에 돈을 지불할 수도 있지만 블로깅을 시작하기에 가장 쉬운 방법은 다음에 정리한 무료 블로깅 서비스를 이용하는 것이다.

- 워드프레스(<http://wordpress.com>)
- 블로거(<http://blogger.com>)
- 라이브저널(<http://livejournal.com>)

이 외에도 많은 서비스들(국내 서비스로는 [www.tistory.com](http://www.tistory.com), <http://www.egloos.com> 등이 있다)이 있지만 이 세 가지가 블로깅을 시작하기에 적합하다(<http://blogs.about.com/od/blogsoftwareandhosts/a/topfreeblogs.htm>)

## 플리커의 사진을 블로그로 전송하기

## 블로그와 통합되도록 플리커를 설정하기

1. <http://flickr.com/blogs.gne>에 접속하라. 먼저 플리커에 로그인하여야 한다.

•

•

•

•

•

•

•

- 
- 
- 

왜 제일 먼저 블로그 유형에 대한 목록을 제시하는가?

## 워드프레스

1. [http://flickr.com/blogs\\_add.gne](http://flickr.com/blogs_add.gne)에 접속하라. 이 예제를 테스트하려면 여러분 소유의 워드프레스 블로그를 가지고 있어야 한다. 워드프레스 블로그가 없다면 호스팅 서비스에 워드프레스를 설치하거나 무료 워드프레스 서비스를 사용하면 된다(‘블로깅을 시작할 때 사용할 수 있는 무료 블로그 호스팅 서비스’란을 참고하자).
2. “여러분은 어떤 종류의 블로그를 가지고 있습니까?” 라는 질문에 대해 워드프레스 블로그를 선택하라. 워드프레스 블로그를 선택하면 [http://flickr.com/blogs\\_add\\_](http://flickr.com/blogs_add_)

metaweblogapi.gne라는 URL(이것으로 미루어보아 워드프레스는 메타웹로그 API<sup>1</sup>를 통해 접근할 수 있음을 짐작할 수 있다)로 이동하게 된다.

3. 아래 파라미터들을 입력하라.

- API 엔드포인트(워드프레스 블로그에서 이 URL은 `http://{여러분의 블로그 URL}/xmlrpc.php`의 형태이며 예를 들어 `http://blog.mashupguide.net/xmlrpc.php`와 같이 사용된다.)
- 사용자 이름(Username)
- 비밀번호>Password)

6. 테스트 포스트를 등록하여 블로그 설정이 잘 되었는지 테스트할 수 있다. 테스트하려면, `http://flickr.com/blogs.gne`에 접속하여 해당하는 블로그란의 '테스트 포스트' 버튼을 클릭하면 된다. 모든 작업이 잘 수행되면, "[여러분의 블로그 이름]로의 테스트 포스트가 전송되었습니다. 완료된 후에는 삭제하셔도 됩니다." 라는 메시지를 볼 수 있으며 여러분의 블로그에 테스트용 포스트가 게시된 것을 확인할 수 있을 것이다.

## 블로거 블로그

1. `http://flickr.com/blogs_add.gne` 의 드롭-다운 메뉴에서 블로거 블로그를 선택한다. 블로거 블로그를 가지고 있어야 하므로 `http://www2.blogger.com/create-blog.g`에서 계정을 생성하도록 하라.

---

<sup>1</sup> <http://en.wikipedia.org/wiki/MetaWeblog>

## Your account / Blogs / Add a blog

### Authorize posting to your Blogger (beta) blog

If you're using the new [beta.blogger.com](http://beta.blogger.com), part of setting up your Blogger Beta blog to post from Flickr is you signing in at Google to establish permission.

So, here's the process:

1. Head to [google.com](http://google.com) and if you aren't already logged in you'll be asked to do so.
2. On the page you end up on, you need to "Grant Access" to flickr.com.
3. Return to Flickr, finish setting things up

[Head over to Google now](#)



플리커의 사진을 블로그하기



어떻게 플리커와 블로그의 통합이 이루어지는가?

■

•

•

•

•

■

■

■

■

■

■

## 데스크톱 기반 블로깅 툴

- w.bloggar(<http://www.wbloggar.com>)는 윈도우 기반 데스크톱 클라이언트이다.
- ecto(<http://ecto.kung-foo.tv/>)는 윈도우와 맥 OS X 기반으로 제공된다.
- MarsEdit(<http://www.red-sweater.com/marsedit/>)는 맥 OS X 기반으로 제공된다.
- Windows Live Writer(<http://windowslivewriter.spaces.live.com/>)는 윈도우 기반 데스크톱 클라이언트이다.
- BlogDesk(<http://www.blogdesk.org/en/index.htm>)는 윈도우 기반 데스크톱 클라이언트이다.
- ScribeFire(<http://www.scribefire.com/>)는 파이어폭스 내에서 실행되는 클라이언트이다.
- mo:Blog(<http://www.moblogworld.net/>)는 Palm OS 디바이스용 클라이언트이다.



[ 100% DE D1 ]

## Extracting text from a Word document

Although I'm writing [my mashup book](#) in Microsoft Word, I'd like to publish it in a variety of forms, including HTML, various varieties of XML, PDF, Wiki-markup. There are various ways to extract content out of my Word documents, including Word macros, external scripts using the COM interface, or saving the Word 2003 documents as Word XML. I'm partial to using Python to do some simple extraction of text as a first step:

```

import win32com.client
wd = win32com.client.Dispatch(
    doc = wd.Documents.Open(r'file:
    print doc.Content.Text

```

I've not been able to find complete references, probably, in [book documentation for Office](#).

[Edit this entry](#)

Posted 10:

---

**Post a Comment**

Logged in as [aaron@vab](#) [Log Off](#)

Message

**Post: 16**

**Extracting text from a Word document**

Although I'm writing [my mashup book](#) in Microsoft Word, I'd like to publish it in a variety of forms, including HTML, various varieties of XML, PDF, Wiki-markup. There are various ways to extract content out of my Word documents, including Word macros, external scripts using the COM interface, or saving the Word 2003 documents as Word XML. I'm partial to using Python to do some simple extraction of text as a first step:

**Recent Posts**

- ☐ 18 - Notebook for 2007.02.19
- ☒ 16 - Extracting text from a Word
- ☐ 14 - Douglas Comer's Internet Boc
- ☐ 13 - Notebook for 2007.01.15
- ☐ 12 - googleder.us sample query
- ☐ 11 - some academic research on n
- ☐ 10 - Flickr downloaders
- ☐ 9 - Let's see whether I get heard
- ☐ 8 - Wanted: a technical reviewe
- ☐ 7 - What you will find on mashup
- ☐ 6 - Hello world!
- ☐ 5 - other stuff to consider coveri
- ☐ 4 - The convergence of teachi

Extracting text from a Word document:

Although I'm writing [my mashup book](#) in Microsoft Word, I'd like to publish it in a variety of forms, including HTML, various varieties of XML, PDF, Wiki-markup. There are various ways to extract content out of my Word documents, including Word macros, external scripts using the COM interface, or saving the Word 2003 documents as Word XML. I'm partial to using Python to do some simple extraction of text as a first step:

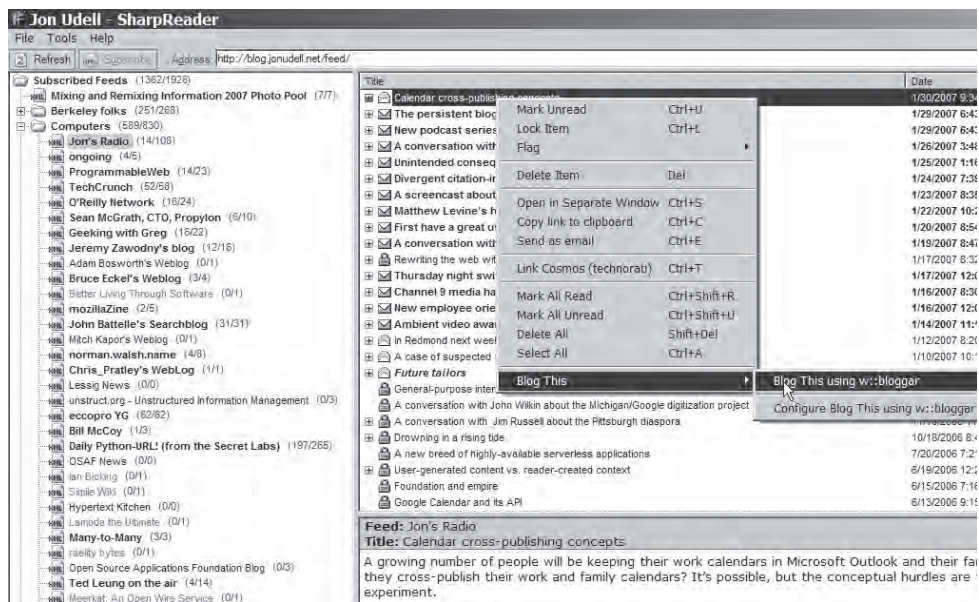
```

import win32com.client
wd = win32com.client.Dispatch(
    doc = wd.Documents.Open(r'file:
    print doc.Content.Text

```

Monday, February 04, 2007 08:32

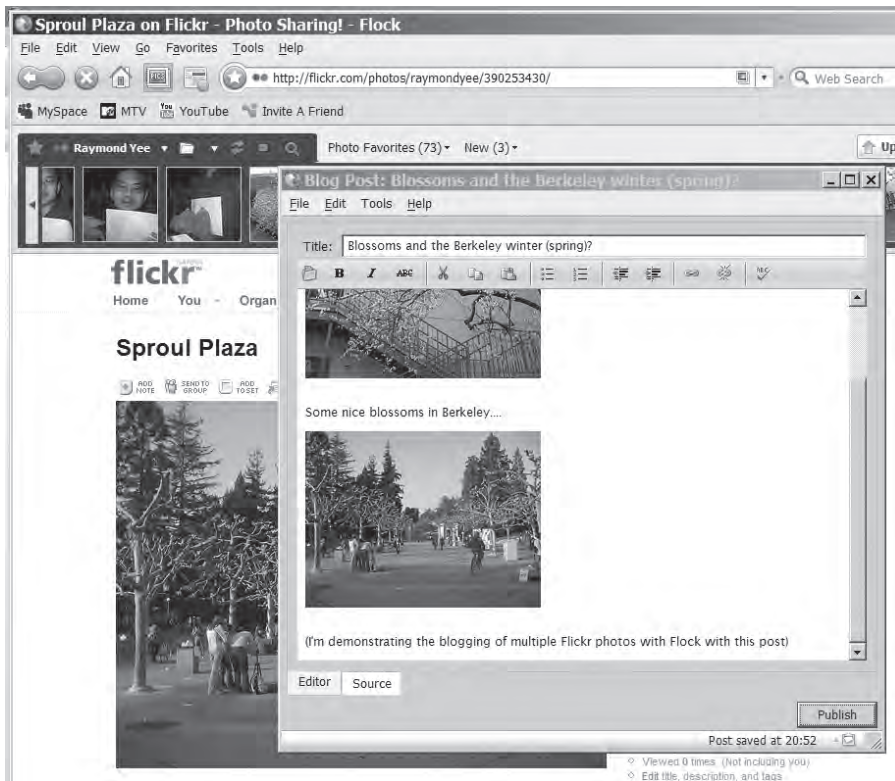
## 피드와 블로그의 조합을 통한 리블로깅



- 5 <http://www.sharpreader.net/>
- 6 <http://wbloggar.com/>
- 7 <http://www.sharpreader.net/plugins.html>

## 플락 : 블로그와 플리커를 함께 사용하기

- 플리커(Flickr), 포토버킷(Photobucket), 유튜브(YouTube)의 통합
- 
- 딜리셔스(del.icio.us), 매그놀리아(ma.gnolia.com)와 같은 소셜 북마크와의 통합
-



RSD: 블로그 API의 인지성

[http://en.wikipedia.org/wiki/Really\\_Simple\\_Discovery](http://en.wikipedia.org/wiki/Really_Simple_Discovery)

<http://cyber.law.harvard.edu/blogs/gems/tech/rsd.html>

<http://wordpress.com/blog/>

링크백

## 초기단계의 위키 통합

---

9 이전버전인 <http://en.wikipedia.org/w/index.php?title=Wiki&oldid=109882004>가 <http://en.wikipedia.org/wiki/Wiki> 로 수정되었다.

10 <http://www.docuverse.com/blog/donpark/2003/09/05/wiki-based-websites>

11 미디어위키는 <http://en.wikipedia.org/w/api.php>를, 그리고 인기 있는 무료 위키 호스팅 서비스 제공자인 PBWiki는 <http://api.pbwiki.com/>를 참고하라.

1. <http://futef.com/> 에 접속하여 Bach를 검색해보라.
2. 반환되는 검색결과에서 순서와 카테고리에 대해서 연구해보라.
3. FUTEF에서 본 결과와 위키피디아에서 동일한 검색을 수행한 결과를 비교해보라. 위키 피디아에서, 바흐Johann Sebastian Bach에 대한 페이지로 즉시 리다이렉션될 것이다. 바흐와 관련된 다른 검색어에 대해서는 바흐 관련 용어확인 페이지disambiguation page를 살펴 보도록 하라.<sup>12</sup>

[http://en.wikipedia.org/wiki/Wikipedia:Text\\_editor\\_support](http://en.wikipedia.org/wiki/Wikipedia:Text_editor_support)

<http://meta.wikimedia.org/wiki/Bot>

## 요약

■

■

■

---

<sup>12</sup> [http://en.wikipedia.org/wiki/Bach\\_%28disambiguation%29](http://en.wikipedia.org/wiki/Bach_%28disambiguation%29)

<sup>13</sup> <http://www.programmableweb.com/api/FUTEFWikipedia>



■

■



2부

---

# API를 사용하여 단일 웹 애플리케이션을 리믹스 하기

---



## 06

---

플리커를 통해  
웹서비스 API에 대해 배워보자

---

## 플리커 API 소개

### 플리커 API 키 발급받기

플리커 API를 사용하려면 키가 필요하다. 키 값은 API 요청을 보낼 때 여러분을 식별하기 위해 사용되는 것이며 숫자나 글자의 문자열로 이루어진다. 즉, API를 호출할 때 일반적으로 키를 전달해줘야 하며(또는 키로부터 생성된 다른 파라미터를 전달해야 한다) 이러한 키 값은 플리커에 애플리케이션을 등록하여 생성할 수 있다.

<http://www.flickr.com/services/api/keys/apply/>

사용할 키를 생성하여 이번 장과 다음 장에서 진행할 실습에 사용하도록 하자. 아래 URL에서 현재 소유하고 있는 키를 확인할 수 있다.

<http://www.flickr.com/services/api/keys/>

다음 장을 읽어보면, 키를 사용하는 것이 다른 애플리케이션 API에서도 사용되는 일반적인 방법이라는 것을 알게 될 것이다. 키를 통해서 API 제공자는 API 사용자를 식별할 수 있게 되며(일반적으로 다른 정보가 제공되지 않더라도 적어도 API키 소유자의 이메일 주소는 알 수 있다.) 사용자가 API에 접근하는 방법을 모니터링할 수 있다(API 호출 빈도 및 횟수, 요청 유형에 관해). 이러한 추적을 통해 API 제공자는 사용자에게 이메일을 보내 해당 키의 사용을 제한하는 것으로부터 과도한 사용자를 고소하는 것에 이르기까지 API 사용에 대한 사용 약관을 보강할 수도 있다.

[http://api.flickr.com/services/rest/?method=flickr.test.echo&api\\_key={api-key}](http://api.flickr.com/services/rest/?method=flickr.test.echo&api_key={api-key})

<http://www.flickr.com/photos/tags/{tag}/>

<http://www.flickr.com/photos/tags/puppy/>

[http://api.flickr.com/services/rest/?method=flickr.photos.search&api\\_key={api\\_key}&tags={tag}&per\\_page={per\\_page}](http://api.flickr.com/services/rest/?method=flickr.photos.search&api_key={api_key}&tags={tag}&per_page={per_page})

[http://api.flickr.com/services/rest/?method=flickr.photos.search&api\\_key=835b1db9b1b5af3bde216143b1a6dc0c&tags=puppy&per\\_page=3](http://api.flickr.com/services/rest/?method=flickr.photos.search&api_key=835b1db9b1b5af3bde216143b1a6dc0c&tags=puppy&per_page=3)



■

■

•

•

•

•

✱

■

■

■

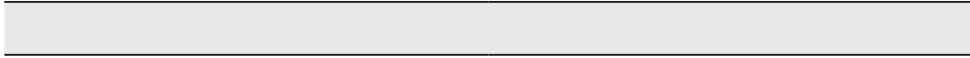
이 XML 응답은 무엇을 의미하는가?

<http://www.flickr.com/photos/tags/puppy/>

[http://api.flickr.com/services/rest/?method=flickr.photos.search&api\\_key={api\\_key}&tags=puppy&per\\_page=3](http://api.flickr.com/services/rest/?method=flickr.photos.search&api_key={api_key}&tags=puppy&per_page=3)

■

■



---

`http://www.flickr.com/photos/{user-id}/{photo-id}/`

`http://farm{farm-id}.static.flickr.com/{server-id}/{photo-id}_{photo-secret}.jpg`

`http://www.flickr.com/photos/7841384@N07/1153699093/`

`http://farm2.static.flickr.com/1023/1153699093_d1fba451c9.jpg`

**\* NOTE!** 원본 사진에 대한 URL을 어떻게 생성하는지 궁금해할 수도 있다. 원본 사진이 공개적으로 접근이 가능하다는 가정하에, `&extras=original_format`를 쿼리에 추가하면 `originalsecret`와 `originalformat` 어트리뷰트를 얻을 수 있게 된다.

왜 플리커의 UI상에서의 결과와 API상의 결과가 동일하지 않은가?

<http://flickr.com/account/prefs/optout/?from=privacy>

XML 응답을 가지고 무엇을 할 수 있는가?

<http://www.flickr.com/services/api/flickr.photos.search.html>



[http://api.flickr.com/services/rest/?method=flickr.photos.search&api\\_key={api\\_key}&tags=puppy&per\\_page=3](http://api.flickr.com/services/rest/?method=flickr.photos.search&api_key={api_key}&tags=puppy&per_page=3)

## API 문서, 커뮤니티, 그리고 정책

<http://www.flickr.com/services/api/>

- 플리커 API 메일링 리스트(<http://tech.groups.yahoo.com/group/yws-flickr/>)
- 플리커 내의 플리커 API 그룹(<http://www.flickr.com/groups/api/>)

## API의 이용약관

<http://www.flickr.com/services/api/tos/>

## 플리커 API 탐색기와 문서 사용하기

- Activity
- Auth
- Blogs
- Contacts
- Favorites
- Groups
- Groups.pools
- Interestingness
- People
- Photos
- Photos.comments
- Photos.geo
- Photos.licenses
- Photos.notes
- Photos.transform
- Photos.upload
- Photosets
- Photosets.comments
- Prefs
- Reflection
- Tags
- Test
- URLs



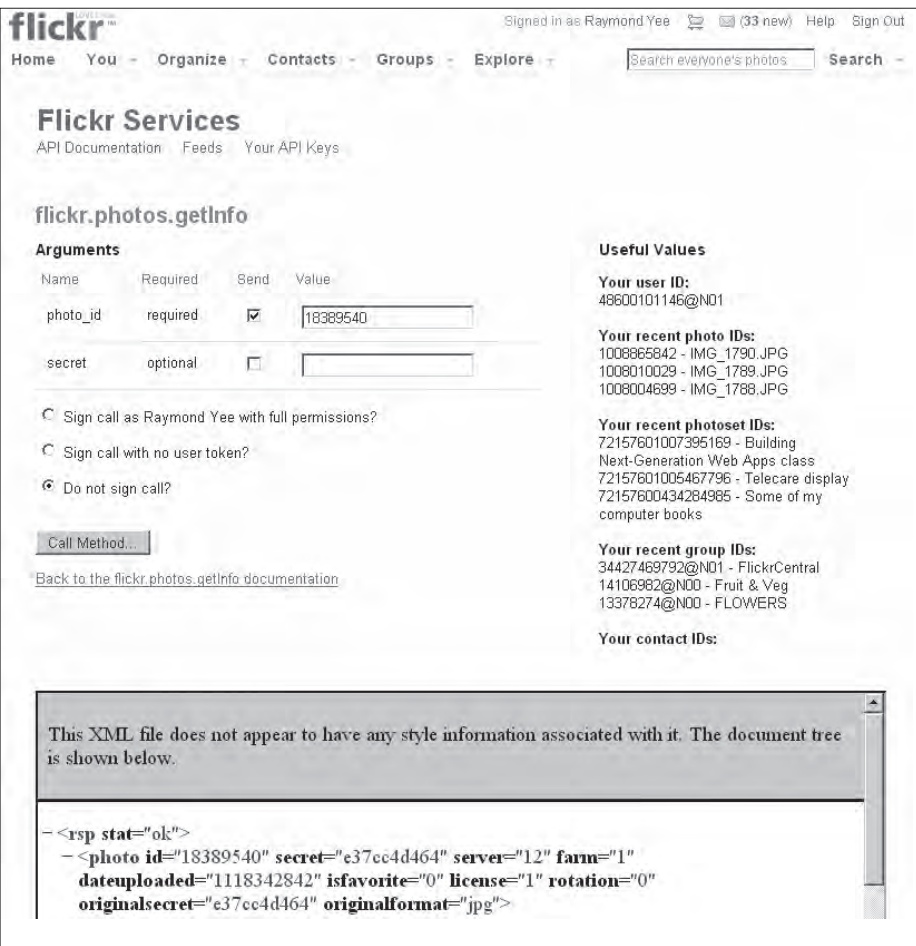
<http://www.flickr.com/services/api/{method-name}.html>

<http://www.flickr.com/services/api/flickr.photos.search.html>

- 메서드의 기능에 대한 설명
- 메서드가 사용자 인증을 필요로 하는지와, 그렇다면 요구되는 최소한의 권한 레벨은 read, write, 또는 delete 중 어떤 것인가. read는 개인적인 정보를 읽는 데 필요한 권한이며 write는 read 권한에 더하여 사진에 대한 메타데이터를 추가, 편집, 삭제할 수 있는 권한이며, delete는 write와 read 권한에 더하여 사진을 삭제할 수 있는 권한이다.
- 메서드가 서명<sup>sign</sup>되어야만 하는가. 사용자 인증이 필요한 모든 메서드는 서명되어야 한다. Auth 그룹에 속하는 모든 메서드들(예들 들어, flickr.auth.getToken)과 같은 메서드들은 사용자 인증은 필요치 않지만 서명되어야 한다. 이번 장의 후반부에 서명에 관련된 기술에 대해서 설명할 것이다.
- 파라미터의 목록, 각 파라미터의 이름, 파라미터들의 필수여부와 각 파라미터에 대한 간략한 설명
- 응답의 예제
- 에러코드

<http://www.flickr.com/services/api/explore/?method={method-name}>

<http://www.flickr.com/services/api/explore/?method=flickr.photos.search>



[http://api.flickr.com/services/rest/?method=flickr.photos.search&api\\_key={api\\_key}&tags=puppy](http://api.flickr.com/services/rest/?method=flickr.photos.search&api_key={api_key}&tags=puppy)



[http://api.flickr.com/services/rest/?method=flickr.photos.search&api\\_key={api\\_key}&tags=puppy&auth\\_token=72157601583650732-e30f91f3313b3d14&api\\_sig=3d7a2d1975e9699246a299d2deaf5b70](http://api.flickr.com/services/rest/?method=flickr.photos.search&api_key={api_key}&tags=puppy&auth_token=72157601583650732-e30f91f3313b3d14&api_sig=3d7a2d1975e9699246a299d2deaf5b70)

## PHP에서 기본적인 플리커 API 메서드 호출하기

`http://api.flickr.com/services/rest/?method=flickr.photos.search&api_key={api_key}&tags={tag}&per_page={per_page}`

`http://api.flickr.com/services/rest/?method=flickr.photos.search&api_key={api_key}&tags=puppy&per_page=3`

`http://www.flickr.com/photos/7841384@N07/1153699093/`

`http://farm2.static.flickr.com/1023/1153699093_d1fba451c9.jpg`

▶ **TIP!** 웹서비스를 디버깅할 때, 와이어샤크(<http://en.wikipedia.org/wiki/Wireshark>)와 같은 네트워크 프로토콜 분석기를 사용하면 많은 도움이 된다. 적절하게 구성된 웹 서비스 호출도 종종 시행착오를 동반한다. 와이어샤크가 지원하는 HTTP 관련 기능을 통해, HTTP 헤더, 응답코드, 본문의 내용을 포함하여 무엇을 전송하였고 수신했는지 정확하게 볼 수 있다.

## HTTP 클라이언트

`http://us3.php.net/curl`

<http://us3.php.net/manual/en/function.curl-setopt.php>

[http://code.blitzaffe.com/pages/phpclasses/files/libcurl\\_emulator\\_52-7](http://code.blitzaffe.com/pages/phpclasses/files/libcurl_emulator_52-7)

**\* NOTE!** 이 책에서는 HTTP 요청을 설명하기 위해 자주 `curl`을 사용한다. 보다 자세한 정보가 필요하다면 <http://curl.haxx.se/> 를 참고하자.

## HTTP에 대해 좀 더 자세히 알아보자

<http://tools.ietf.org/html/rfc1945>

<http://tools.ietf.org/html/rfc2616>

<http://www.w3c.org/Protocols/>

---

1 <http://www.w3.org/TR/html401/interact/forms.html#h-17.13.1>  
2 <http://www.w3.org/TR/html401/interact/forms.html#h-17.13.4>

### 안전한(safe) 메서드와 멍등(idempotent) 메서드

HTTP 명세에는 안전한 메서드와 멍등 메서드에 대해 아래 URL에서 정의하고 있다.

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>

## HTTP의 공식적인 구조

- 메서드method. verb나 action이라고도 알려져 있으며 GET과 POST에 더하여, HTTP 명세는 OPTIONS, HEAD, PUT, DELETE, TRACE, CONNECT와 같은 6가지의 메서드를 정의하고 있다. GET과 POST는 웹 브라우저와 프로그래밍 라이브러리에서 폭넓게 지원하고 있다.
- 경로path. URL 중에서 호스트명의 우측에 기술되는 부분을 말한다.



- 하나 이상의 요청 헤더. <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html> 을 보라
- 요청의 본문body. 비어있을 수도 있다.
- 응답코드. <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html> 에서 긴 에러 코드 목록을 볼 수 있다. 응답코드의 예로는 200 OK, 400 Bad Request, 500 Internal Server Error 가 있다.
- 응답 헤더. <http://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html> 을 보라.
- 응답의 본문

`http://api.flickr.com/services/rest/?method=flickr.photos.search&api_key={api-key}&tags=puppy&per_page=3`

`curl -verbose "http://api.flickr.com/services/rest/?method=flickr.photos.search&api_key={api-key}&tags=puppy&per_page=3"`



- 
- 
- 
- 
- 
- 

- 요청과 응답 시에 데이터를 압축할 수 있도록 해주는가. 이것과 관련된 HTTP 요청과 응답 헤더들은 Accept-Encoding과 Encoding이다.
- 캐싱을 사용할 수 있도록 해주는가. 이것과 관련된 HTTP 요청과 응답 헤더들은 ETag와 If-Modified-Since, 그리고 ETag와 Last-Modified 이다.
- 가장 일반적인 HTTP 사용자 인증 형태인 Basic, Digest, WSSE를 다룰 수 있도록 지원 하는가.
- HTTP의 리다이렉트를 다룰 수 있도록 해주는가
- HTTP 쿠키를 다룰 수 있도록 도와주는가
- 파이썬의 경우 http-lib2 라이브러리(<http://code.google.com/p/http-lib2/>)

- 아파치 자카르타 프로젝트의 HttpClient(<http://jakarta.apache.org/commons/httpclient/>)
- GET 메서드 외의 메서드를 추가적으로 지원하기 위해 루비의 open-uri를 수정한 rest-open-uri(<http://rubyforge.org/projects/rest-open=uri/>)

## XML 처리

<http://us3.php.net/simplexml>

<http://devzone.zend.com/node/view/id/688>







**\* NOTE!** PHP 객체의 구조를 알고 싶을 때는 `print_r`, `var_dump`, 또는 `var_export`와 같은 함수들의 사용을 고려해보라.



## 사진을 표시하는 간단한 HTML을 생성하기

- 
- 

1. 전체 사진의 숫자를 표시한다.(\$xml->photos['total'])
2. 다음의 루프연산을 통해 사진의 배열에 루프연산을 수행한다.
3. 다음 라인에 포함된 로직을 통해 썸네일의 URL과 사진 페이지의 URL을 생성한다.



이 코드가 의미하는 것은 무엇인가?

플리커 API에 대해 전반적으로 알아보자

<http://www.flickr.com/services/api/>

flickr.reflection 메서드 사용하기

- 
- 
- 
- 
- 
-

`http://api.flickr.com/services/rest/?method=flickr.reflection.getMethodInfo&api_key={api-key}&method_name={method-name}`

`http://api.flickr.com/services/rest/?method=flickr.reflection.getMethodInfo&api_key={api-key}&method_name=flickr.photos.geo.setLocation`



- 사용 시에 권한을 요구하지 않는 메서드들은 어떤 것이 있는가?
- 어떤 메서드들이 서명되어야 하는가?
- 플리커 API에서 사용되는 파라미터들로는 어떤 것이 있는가? 어떤 메서드가 어떤 파라미터를 사용하는가? 어떤 메서드들이 동일한 파라미터를 공통으로 사용하는가?

**▼ CAUTION!** 플리커 API에 있어서, 이러한 리플렉션 메서드들은 최신 상태로 유지되고 정확한 정보를 전달할 때에만 유용하다. 내 경우는 리플렉션 API를 사용할 때, 리플렉션 메서드가 API에서 가장 중요한 요소 중 하나로 고려되고 있는가를 의심하게 된 몇 가지 문제(예를 들어, <http://tech.groups.yahoo.com/group/yws-flickr/message/3263>)를 발견하게 되었다.

## PHP에서 플리커의 리플렉션 메서드 사용하기













■

■

■

**\* NOTE!** 이 책에서 제시한 플리커 API의 예제는 모두 `write`나 `delete` 권한을 필요로 하지 않았기 때문에 HTTP GET을 사용해 왔다. 만약 호출이 `write`나 `delete` 권한을 필요로 한다면, 반드시 HTTP POST를 사용하여 플리커 메시지를 호출하여야 한다

## 요청과 응답의 포맷

■

- 기본적으로는 XML이지만 자바스크립트(JSON)나 PHP(PHP의 `unserialize` 메서드에 입력값으로 사용될 수 있는)도 포함할 수 있는 반환값을 가진 HTTP 응답을 처리한다.

**\* NOTE!** 비록 웹서비스가 반드시 HTTP를 사용할 필요는 없지만(예를 들어, SOAP은 SMTP에 바인딩될 수도 있다), HTTP는 플리커 API에서 유일하게 지원되는 전송 프로토콜이다. 그리고 대다수의 웹서비스들, 특히 매쉬업에 사용되는 웹서비스들이 HTTP를 기반으로 개발되었으므로 이 책에서는 HTTP 이외의 전송 프로토콜의 사용에 대해서는 다루지 않을 것이다.

■

**\* NOTE!** 7장에서 REST라는 용어에 대해 다시 상세하게 다룰 것이다. 플리커가 말하는 REST 방식은 웹서비스의 구성에 자주 사용되는 패턴이지만 보다 정확하게 말하자면 REST와 RPC의 하이브리드 형태에 가깝다.

- SOAP은 봉투와 같은 개념의 래퍼로 요청을 감싸서 고수준의 추상화가 가능하도록 해주지만 REST보다 복잡하며 일반적으로 좀 더 전문화된 라이브러리와 툴을 필요로 한

다. 우리는 이 주제에 대한 내용을 다음 장에서 다룰 것이며 플리커의 SOAP 요청 포맷과 기타 API의 SOAP 인터페이스에 대해서 알아볼 것이다. SOAP은 중요한 웹서비스 관련 기술이며 특히 웹서비스를 사용하여 엔터프라이즈 환경을 다루는 사람들에게겐 더욱 그러하다.<sup>4</sup>

**\* NOTE!** SOAP의 버전 1.1에서는, SOAP이 Simple Object Access Protocol의 두문자어(앞글자들에서 따온 단어)였다. 하지만 버전 1.2의 SOAP 명세는 더 이상 SOAP이 Simple Object Access Protocol의 두문자어가 아니라고 밝히고 있다.

■

`http://api.flickr.com/services/rest/?method=flickr.test.echo&api_key={api-key}&format={format}`

---

<sup>4</sup> <http://en.wikipedia.org/wiki/SOAP>

# 플리커의 권한부여

<http://www.flickr.com/services/api/auth.spec.html>

- 웹 애플리케이션
- 데스크톱 애플리케이션
- 모바일 애플리케이션

- 
- 
-



- 

- 

- 

- 

왜 비밀번호를 전송하는 방식이 효과적이지 않은가

웹 애플리케이션에서의 권한부여

Lois와 ContactLister의 예제를 설정하기

플리커가 권한부여 과정을 중재해야 한다.

단계 1 : ContactLister는 플리커를 통해 Lois에게 권한을 요청한다.

[http://flickr.com/services/auth?api\\_key={api\\_key}&perms={perms}&api\\_sig={api\\_sig}](http://flickr.com/services/auth?api_key={api_key}&perms={perms}&api_sig={api_sig})

호출에 서명하기: 어떻게 ContactLister가 호출을 생성하고 전송하는가?

- 

- api\_sig는 사용될 문자열의 md5 해시 값을 16진수 다이제스트 형태로 나타내면 된다.

- Key: 020338ddabd2f41ae7ce9413a8d51429
- Shared secret: f0fc085289c7677a

{secret}api\_key{api\_key}perms{perms}

f0fc085289c7677aapi\_key{api\_key}permsread

f9258a76e4ad3cb5fa40bd8b0098d119

http://flickr.com/services/auth?api\_key={api\_key}&perms=read&api\_sig= f9258a76e4ad3  
cb5fa40bd8b0098d119

**플리커는 서명된 호출을 어떻게 사용하는가?**

**\* NOTE!** md5는 암호화된 해시 값을 만들어내는 데 한계가 있음이 밝혀졌다. 연구원들은 하나의 md5 해시 값과 동일한 해시 값을 생성하는 다른 문자열을 어떻게 만들어 내는지에 대해 설명해냈다. 이러한 취약성이 가짜 플리커 호출을 생성하는 용도로 사용될 수 있을까? 잘 모르겠다. 자세한 정보는 <http://en.wikipedia.org/wiki/MD5> 를 참조하도록 하라.

**단계 2: 플리커가 ContactLister를 대신해 Lois에게 권한부여를 요청한다.**

[http://flickr.com/services/auth?api\\_key={api\\_key}&perms=read&api\\_sig=f9258a76e4ad3cb5fa40bd8b0098d119](http://flickr.com/services/auth?api_key={api_key}&perms=read&api_sig=f9258a76e4ad3cb5fa40bd8b0098d119)

단계 3: 플리커는 ContactLister가 토큰을 가져갈 수 있도록 알려준다.

`http://www.flickr.com/services/api/keys/{api-key}/`


`callback-URL?frob={frob}`

단계 4: ContactLister는 이제 권한을 부여받아 서명된 호출을 생성할 수 있다.

auth-token={token}


<http://flickr.com/services/auth/list.gne>

PHP를 사용하여 권한부여 과정 구현하기



Signed in as Raymond Yee
📧 (23 new)
Help
Sign Out

[Home](#)
[You](#)
[Organize](#)
[Contacts](#)
[Groups](#)
[Explore](#)

[Search](#)



## Hi Raymond Yee



**Authorization examples for MashupGuide Chapter 6 wants to link to your Flickr account.**

You should not authorize [Authorization examples for MashupGuide Chapter 6](#) unless you trust them with access to your account.

By authorizing this link, you'll allow the [Authorization examples for MashupGuide Chapter 6](#) service to provide:

- ♦ **Access** to your photos (including private photos)

OK, I'LL ALLOW IT

Or, [cancel this and go back to Authorization examples for MashupGuide Chapter 6](#).

**What's going on here?**  
Flickr encourages other developers to build cool tools for you to play with, but you must authorize these third parties to access your account.

**Want to know more?**  
A wealth of information lies within the [Flickr Services](#) page.

Your password will always remain private, but [Authorization examples for MashupGuide Chapter 6](#) will have the permissions listed above until you revoke this link to your account. You can revoke account links at any time in [your account](#).

**Authorization examples for MashupGuide Chapter 6 provides the following description:**

This key and shared secret is for use of Mashup Guide







**\* NOTE!** 사진 업로드는 플리커 API에서 주요한 부분이지만 이 책에서 다루지는 않다. 업로드에 관해서는 문서(<http://www.flickr.com/services/api/upload.api.html>)를 참조하고 API 키트를 사용하면 편리하게 구현할 수 있을 것이다.

## 플리커 API 키트 사용하기

<http://www.flickr.com/services/api/>

### INCLUDE\_PATH와 플리커의 API 키 값 설정하기

서드파티 라이브러리를 사용할 때면 항상 PHP 패스(include\_path 변수)를 올바르게 설정하여 PHP 코드에서 라이브러리를 찾을 수 있도록 해주어야 한다. php.ini 파일에 접근할 수 있다면 이 파일을 사용하라. 또한 코드 내에서 include\_path 변수를 설정하는 함수인 ini\_set() 함수를 사용할 수도 있다. 아래 코드에서는 include\_path 변수가 적절하게 설정되어 있다고 가정한다.

또한, 여러분의 플리커 API 키와 secret 값을 외부 파일에 저장해두고 인클루드하는 것도 편리한 방법이다. 이어지는 예제에서는 flickr\_key.php 라는 파일에 다음의 내용을 저장해 놓았다.

## PEAR::Flickr\_API

Pear install -of [http://code.iamcal.com/php/flickr/Flickr\\_API-Latest.tgz](http://code.iamcal.com/php/flickr/Flickr_API-Latest.tgz)

■

■

■

## phpFlickr

---

8 [http://pear.php.net/package/XML\\_Tree](http://pear.php.net/package/XML_Tree) -- 이 패키지는 [XML\\_Serializer](http://pear.php.net/package/XML_Serializer)([http://pear.php.net/package/XML\\_Serializer](http://pear.php.net/package/XML_Serializer))에 의해 대체되었다.

1. <http://phpflickr.com/docs/?page=install> 에 자세히 정리된 지침을 따르도록 하라.  
<http://sourceforge.net/projects/phpflickr> 에서 최신 ZIP 파일을 다운로드 받도록 하자. 이 책이 집필되는 시점에서 최신버전은 다음과 같다.<sup>9</sup>

<http://downloads.sourceforge.net/phpflickr/phpFlickr-2.1.0.tar.gz>

<http://downloads.sourceforge.net/phpflickr/phpFlickr-2.1.0.zip>

3. 코드가 동작하는지 확인해보기 위해 다음의 코드를 복사하여 붙여 넣도록 하자.

- 생성자는 필수 파라미터인 API 키와 두 개의 선택 파라미터인 `secret`, 그리고 `die_on_error`(에러 상황에서 실행을 멈출지에 대한 Boolean 값) 등을 받도록 되어 있다. `$api`에 대해서 `getErrorCode()`와 `getErrorMsg()` 함수들을 사용할 수 있다는 사실을 기억하

---

<sup>9</sup> [http://sourceforge.net/project/showfiles.php?group\\_id=139987&package\\_id=153541&release\\_id=488387](http://sourceforge.net/project/showfiles.php?group_id=139987&package_id=153541&release_id=488387)

<sup>10</sup> `pear install -o http://downloads.sourceforge.net/phpflickr/phpFlickr-2.1.0.tar.gz` 를 실행하면 "Could not extract the package.xml file from /home/rdhyee/pear/temp/download/phpFlickr-2.1.0.tar.gz" 라는 메시지를 출력한다.

도록 하라.<sup>11</sup>

- 아래 내용은 공식문서로부터 발췌한 내용이다.
  - 플리커 API상의 모든 메서드가 `phpFlickr` 클래스 내에 구현되었다.
  - 메서드를 호출하려면, 플리커 API 이름에서 `flickr.` 를 제거하고 마침표를 밑줄(\_)로 교체하라. 파라미터를 연관배열로 넘겨주는 `flickr.photos.search` 를 제외하고는 플리커 문서에 정리된 파라미터의 순서와 동일하게 함수에 파라미터를 넘겨주어 호출할 수 있다.
  - 캐싱 기능을 활성화하려면 `phpFlickr::enableCache()` 함수를 사용하라.

## Phlickr

<http://drewish.com/projects/phlickr/>

[http://sourceforge.net/project/showfiles.php?group\\_id=129880](http://sourceforge.net/project/showfiles.php?group_id=129880)

---

<sup>11</sup> <http://phpflickr.com/docs/?page=install>

\*

플리커 API의 한계

■



- 

- 

- 

- 

- API는 현재 태그 클러스터를 지원하지 않는다(<http://tech.groups.yahoo.com/group/yws-flickr/message/1596>).

<http://www.flickr.com/photos/{user-id}/archives/>

요약

# 07

---

## 다른 웹 API에 대해 알아보자

---

- 

- 

-



## XML-RPC

▶ **TIP!** 이번 절을 읽기 전에, 6장의 'HTTP에 대해 다시 알아보기' 절을 읽어보고 HTTP 요청과 응답의 구조, 그리고 다양한 HTTP 요청 메서드에 대해 숙지하는 것이 도움이 될 것이다.

`http://api.flickr.com/services/rest/?method=flickr.test.echo&api_key={api-key}`

<http://docs.python.org/lib/module-xmlrpclib.html>

<http://api.flickr.com/services/xmlrpc/>

- <http://phpxmlrpc.sourceforge.net/>
- [http://pear.php.net/package/XML\\_RPC/](http://pear.php.net/package/XML_RPC/)

`pear install XML_RPC`

- <http://time.xmlrpc.com/>에 파라미터가 필요하지 않은 함수를 호출하여 현재 시간을 가져올 수 있다.
- `search_example()`에서, `flickr.photos.search` 함수를 호출할 수 있다.

- flickr\_client 클래스는 search\_example()이 보다 많은 플리커 메서드를 다룰 수 있도록 어떻게 일반화할 수 있는지 보여준다.





네트워크상에서는 어떤 일이 벌어지는가?

HTTP 메시지의 분석과 구성에 와이어샹크와 킬 사용하기

▶ **TIP!** 와이어샹크를 사용할 때, Promiscuous Mode 옵션에 있는 캡처 패킷 옵션을 끄고 사용하는 것이 도움이 될 것이다. 또한 웹서비스의 트래픽을 연구하려면, HTTP 트래픽만을 필터링하도록 하라. 그렇지 않으면, 너무 많은 데이터가 표시될 것이다.

curl은 URL 문법을 사용한 파일전송, FTP, FTPS, HTTP, HTTPS, SCP, SFTP, TFTP, TELNET, DICT, FILE 그리고 LDAP를 지원하는 커멘드라인 툴이다. curl은 SSL 인증서, HTTP POST, HTTP PUT, FTP 업로딩, HTTP 폼 기반 업로드, 프록시, 쿠키, 사용자 패스워드 인증(Basic, Digest, NTLM, Negotiate, Kerberos…), 이어서 파일전송하기, 프록시 터널링, 그밖의 많은 유용한 기능을 제공한다.

- <http://curl.haxx.se/docs/manpage.html>은 curl의 메뉴얼 페이지이다.
- <http://curl.haxx.se/docs/https scripting.html>은 구체적인 예제를 제공하기 때문에 여러 면에서 가장 도움이 되는 페이지이다.

```
curl http://www.yahoo.com
```

```
curl -v -X GET http://www.yahoo.com
```

## XML-RPC 트래픽을 파싱하기

POST /services/xmlrpc/ HTTP/1.0





**\* NOTE!** 윈도우, 맥 OS X, 그리고 리눅스 시스템의 커멘드라인에서 `curl`을 이용한 호출이 동작하도록, XML 전체를 감싸는 용도로 큰따옴표를 사용하고 XML 내부에서 작은 따옴표를 사용하도록 재작성하였다.

SOAP

**WSDL과 SOAP을 사용하면 웹서비스 호출이 쉬워진다는 것은 희망사항이다**

1. 사용하고자 하는 적절한 플리커 메서드를 찾아낸다
2. 전달할 파라미터를 찾아내고 어떻게 값들을 패키징할지 알아낸다
3. XML로 반환되는 결과 값을 파싱한다.



geocoder.us

`http://geocoder.us/service/rest/geocode?address={address}`

`http://geocoder.us/service/rest/geocode?address=2855+Telegraph+Ave%2C+Berkeley%2C+CA`

`http://geocoder.us/dist/eg/clients/GeoCoderPHP.wsdl`

**\* NOTE!** PHP5를 사용하면 geocoder.us가 참조하는 첫 번째 WSDL 문서(`http://geocoder.us/dist/eg/clients/GeoCoder.wsdl`)를 다루기가 어렵기 때문에, 이번 장에서는 대신 두 번째 WSDL 문서를 사용한다.

■

■

■

## oXygen XML 에디터 사용하기

**\* NOTE!** oXygen은 상용 제품으로 30일 동안 무료로 사용해볼 수 있다. 또 다른 상업제품인 XML Spy(<http://www.altova.com/>) 역시 유사한 WSDL 툴을 제공한다. 비주얼한 환경에서 WSDL 문서를 탐색하고 오퍼레이션을 호출할 수 있는 오픈소스 프로젝트로 웹 툴 프로젝트(<http://www.eclipse.org/webtools/>)의 일부인 이클립스용 웹 서비스 탐색기가 있다

- 서비스(Services)
- 포트(Ports)
- 오퍼레이션(Operations)

WSDL SOAP Analyser

WSDL

Services: GeoCode\_Service

Ports: GeoCode\_Port

Operations: geocode

Actions: geocode

URL: geocode\_address

SOAP Action: http://rpc.geocoder.us/Geo/Coder/US#geocode

Request Attachments

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <oxy:geocode xmlns:oxy="http://rpc.geocoder.us/Geo/Coder/US/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <location>STRING</location>
    </oxy:geocode>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Send Open Save Regenerate

Response

```
<!-- Auto-generated server sample response -->
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <oxy:geocode xmlns:oxy="http://rpc.geocoder.us/Geo/Coder/US/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <results/>
    </oxy:geocode>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



- 

- 

- 

- <http://rpc.geocoder.us/Geo/Coder/US#geocode> 의 SOAP 액션. geocoder.us에서 사용되는 SOAP의 버전인 SOAP1.1에서, SOAP 액션은 SOAPAction HTTP 요청 헤더로 서 전송된다.
- SOAP 호출이 요청되는 타겟 URL(또는 위치)은 <http://rpc.geocoder.us/service/soap/> 이다.

## SOAP1.1 과 SOAP1.2

이상적인 경우라면, 사용자는 SOAP 프로토콜에 대해 자세하게 알아야 할 필요가 없다. 결국, SOAP의 요점은 웹서비스에 대한 접근을 여러분의 데스크톱이나 서버에 존재하는 오브젝트를 프로그래밍하는 것처럼 보이게 하는 것이다. 그러나 라이브러리와 서비스들은 실제 사용되는 SOAP의 버전에 결정적인 의존성을 가지고 있다. 예를 들어보자.

SOAP은 W3C 권고안이 되었으며 최신 버전은 1.2이다.

<http://www.w3c.org/TR/soap12-part1/>

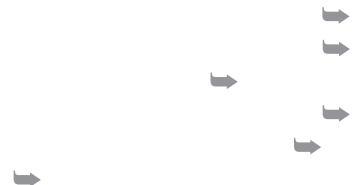
SOAP의 이전 버전들은 여전히 많이 사용되고 있으며 1.2 버전보다 많이 사용되고 있을 것이다. 1.1 버전에 대한 명세는 아래에 정리되어 있다.

<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/?>>

아래에 이 두 가지 명세들 간의 중대한 차이점들에 대해 정리하였다(이들 간의 차이점은 <http://www.w3c.org/TR/2007/REC-soap12-part0-20070427/#L4697>에 자세히 설명되어 있다).

- SOAP 메시지를 감싸는 봉투(envelope)에 다른 네임스페이스를 사용한다(버전 1.2는 <http://www.w3.org/2003/05/soap-envelope>에서 버전 1.1은 <http://schemas.xmlsoap.org/soap/envelope/>에서 확인하도록 하자). 이 부분을 보면 어떤 버전의 SOAP을 다루고 있는지 확인할 수 있다.
- SOAP HTTP 바인딩에 필요한 SOAPAction 파라미터의 사용방법이 다르다. SOAP1.2에서는 SOAPAction HTTP 요청 헤더가 더 이상 사용되지 않는다.
- SOAP1.2는 식별을 위해 HTTP 응답 헤더의 Content-Type에 "application/soap+xml"을 사용한다.

라이브러리와 툴들이 각각 다른 버전의 SOAP를 지원하기 때문에 이러한 차이점들에 대해 언급하였다.



**\* NOTE!** 구체적인 예제로 플리커를 사용하지 않는 이유는 플리커가 비록 SOAP 인터페이스를 제공하는데는 하지만 WSDL 문서를 제공하지 않기 때문이다. 플리커에 대해서는 이 장의 후반부인 'SOAP을 통한 플리커 API'에서 다루도록 할 것이다.

<http://www.w3.org/2000/06/webdata/xslt?xslfile=http://tomi.vanek.sk/xml/wSDL-viewer.xsl&xmlfile=http://geocoder.us/dist/eg/clients/GeoCoderPHP.wsdl&transform=Submit>

## 파이썬의 SOAPpy 사용하기

**\* NOTE!** SOAPpy는 <http://pywebsvcs.sourceforge.net/>에서 다운로드할 수 있으며 마크 필그림 Mark Pilgrim이 운영하는 사이트인 Dive Into Python([http://www.diveintopython.org/soap\\_web\\_services/index.html](http://www.diveintopython.org/soap_web_services/index.html))은 SOAPpy에 대한 튜토리얼을 제공한다.





## PHP에서 SOAP 사용하기

PHP에서 SOAP을 호출하는데 사용할 수 있는 몇 가지 라이브러리를 아래에 정리하였다.

- NuSOAP(<http://sourceforge.net/projects/nusoap/>)
- PEAR::SOAP 패키지(<http://pear.php.net/package/SOAP>)
- PHP 설치 시에 enable-soap 플래그를 주면 PHP5에 내장된 SOAP 라이브러리(<http://us2.php.net/soap>)를 사용할 수 있다.

이 책에서는 PEAR::SOAP 라이브러리를 사용한다.

## PHP의 PEAR::SOAP 사용하기

**\* NOTE!** PEAR::SOAP을 사용하여 주어진 WSDL 파일에서 이용 가능한 오퍼레이션을 가져오는 방법은 알아내지 못했다

## 아마존 ECS

<http://www.amazon.com/E-Commerce-Service-AWS-home-page/b?ie=UTF8&node=12738641>

### 아마존 ECS 계정 설정하기

서비스를 이용하려면 플리커에서와 같이 아래 URL에서 계정을 등록하여 키를 발급받아야 한다

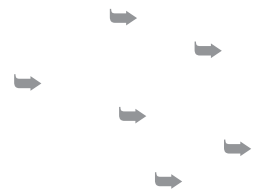
<http://www.amazon.com/gp/aws/registration/registration-form.html>

만약 이미 계정을 가지고 있다면, 아래 URL에서 키 값을 찾을 수 있다.

<http://aws-portal.amazon.com/gp/aws/developer/account/index.html?ie=UTF8&action=access-key>

<http://webservices.amazon.com/AWSECommerceService/AWSECommerceService.wsdl?>





■

■

■





**\* NOTE!** SOAP과 WSDL을 강력하게 지원하는 자바나 .NET 프로그래밍 환경을 사용한다면 이 기술들을 사용하기가 훨씬 쉬울 수도 있다. 그렇다고 SOAP과 WSDL을 사용하지 말라는 것은 아니다. 아파치 프로젝트의 Axis(<http://ws.apache.org/axis/java/index.html>)와 같은 라이브러리나 .NET에 포함된 WSDL 관련 기능들이 SOAP 개발의 삶을 보다 편하게 해줄 수 있는지 알아낼 수 있기를 바란다.

## SOAP을 통해서 플리커 API를 사용하기

<http://www.flickr.com/services/api/request.soap.html>

<http://www.flickr.com/services/api/response.soap.html>

## 특정 웹 API에 대해서 알아보자



**\* NOTE!** 웹서비스에 대한 디렉터리 서비스를 제공하는 참고할 만한 사이트로는 공개적으로 사용할 수 있는 SOAP 서비스를 정리해 놓은 <http://www.xmethods.net/>이 있으며 무료로 시험해볼 수 있는 상용 웹서비스를 제공하는 <http://strikeiron.com/>도 있다.

## Programmableweb.com

- <http://www.programmableweb.com/apis>는 등록된 최신 API들과 매쉬업에서 가장 많이 사용되는 API 들을 목록으로 보여주는 'API 대쉬보드' 이다.
- <http://www.programmableweb.com/apilist/bycat>는 카테고리별로 API를 보여준다. 여기에 정리된 다양한 카테고리를 이해하면 어떤 분야의 사람들이 의욕적으로 API를 개발하고 있는지 아는 데 도움이 될 것이다.
- <http://www.programmableweb.com/apilist/bymashups> 는 Programmableweb.com 에 등록된 매쉬업에 해당 API가 몇 번이나 사용되었는가를 기준으로 API를 보여준다.

- 
- 93개의 API가 연관된 WSDL 파일을 제공하고 있다.
  - SOAP을 지원하는 131개의 API중 42개가 REST를 지원한다. 나머지 89개는 SOAP은 지원하지만 REST는 지원하지 않는다. 88개의 API는 SOAP만을 지원한다.
  - 9개의 API는 단지 XML-RPC만을 지원한다.
  - 25개의 API는 자바스크립트만을 지원한다.
- 
- REST가 웹 API를 제공하는 가장 주요한 형태이지만 SOAP만을 지원하는 API도 상당한 숫자에 이른다.
  - 상대적으로 적은 숫자의 API가 XML-RPC만을 유일한 옵션으로 지원하고 있다.

**\* NOTE!** 많은 수의 API들이 자바스크립트를 프로토콜로 지원하고 있다. 다음 장에서 이러한 API들에 대해서 다룰 것이다.



유튜브

[http://www.youtube.com/api2\\_rest?method=youtube.users.get\\_profile&dev\\_id={youtube-key}&user=rdhyee](http://www.youtube.com/api2_rest?method=youtube.users.get_profile&dev_id={youtube-key}&user=rdhyee)

[http://www.youtube.com/api2\\_rest?method=youtube.users.list\\_favorite\\_videos&dev\\_id={youtube-key}&user=rdhyee](http://www.youtube.com/api2_rest?method=youtube.users.list_favorite_videos&dev_id={youtube-key}&user=rdhyee)

[http://www.youtube.com/api2\\_rest?method=youtube.videos.get\\_details&dev\\_id={youtube-key}&video\\_id=XHnE4umovw4](http://www.youtube.com/api2_rest?method=youtube.videos.get_details&dev_id={youtube-key}&video_id=XHnE4umovw4)

[http://www.youtube.com/api2\\_rest?method=youtube.videos.list\\_by\\_tag&dev\\_id={youtube-key}&tag=HolidayWeekend&page=1&per\\_page=100](http://www.youtube.com/api2_rest?method=youtube.videos.list_by_tag&dev_id={youtube-key}&tag=HolidayWeekend&page=1&per_page=100)

**▼ CAUTION!** 유튜브 API는 다른 구글의 API(<http://code.google.com/apis/youtube/overview.html>)들과 비슷하게 발전해 갈 것으로 예상된다.

## GData와 블로거 API

<http://tools.ietf.org/wg/atopub/draft-ietf-atopub-protocol/>

<http://tools.ietf.org/wg/atopub/>

<http://code.google.com/apis/gdata/index.html>

[http://code.google.com/apis/blogger/developers\\_guide\\_protocol.html](http://code.google.com/apis/blogger/developers_guide_protocol.html)

## 권한인증 토큰 획득하기

<https://www.google.com/accounts/NewAccount>

[http://code.google.com/apis/blogger/developers\\_guide\\_protocol.html#client\\_login](http://code.google.com/apis/blogger/developers_guide_protocol.html#client_login)

<https://www.google.com/accounts/ClientLogin>

- Email: 지메일 주소 (예를 들어, raymond.yee@gmail.com)
- Password: 구글 비밀번호
- source: 프로그램을 식별하기 위해 회사명-애플리케이션명-버전ID 형태로 구성되는 문자열(예를 들면, mashupguide.net-Chap7-v1)
- service: 구글의 서비스명, 이 경우에는 blogger



## 블로거 사용자 ID를 알아내기

<http://www.blogger.com/create-blog.g>

<http://www.blogger.com/home>

<http://www.blogger.com/profile/{userID}>

<http://www.blogger.com/profile/13847941708302188690>

## 사용자의 블로그 목록과 특정 블로그의 포스팅 목록 가져오기

<http://www.blogger.com/profile/{userID}>

<http://www.blogger.com/feeds/{userID}/blogs>

`http://www.blogger.com/feeds/{blogID}/posts/default`

`http://www.blogger.com/feeds/5586336/posts/default`

## 블로그에 새로운 글 작성하기

1. 다음과 같이 <entry> Atom 엘리먼트로 포맷팅된 블로그 포스트를 생성하자.



2. 위 내용을 파일에 작성하고 `blogger.message.1.xml` 이란 이름으로 저장하자.
3. 블로그에 새로운 포스트(위의 정의에서 말한 '요청 URI에 의해 식별되는 새로운 엔티티')를 생성하기 위해서 블로그의 모든 포스트를 피드로 제공하는 `http://www.blogger.com/feeds/{blogID}/posts/default` 에 이 파일내용을 POST로 전송하는 curl 호출을 생성하자.



5. 이 예제에서, POST 요청은 포스트 아이디 4092273492173517704를 가진 새로운 블로그 포스트를 생성하였다.

## 블로그 포스트를 수정하기



블로그의 포스트를 삭제하기



블로거 API를 HTTP 메서드에 기반한 표준 인터페이스로 사용하기

요약



## 08

---

Ajax/자바스크립트 위젯과  
위젯 API에 대해 배워보자

---

■

■

■

■

■

■

■

■

---

1 브루스 페리(Ajax Hacks, 2006년 오라일리 출판)

어떤 것을 알아야 하는가

■

■

■

■

■

■

■

■

■

Ajax는 어떤 차이점을 만들어 내는가?

- 파이어폭스에서 도구 ▸ 설정 ▸ 내용 ▸ 자바스크립트 사용 옵션을 체크하지 않는다.
- 인터넷 익스플로러에서 도구 ▸ 인터넷 옵션 ▸ 보안 ▸ 사용자지정 수준 ▸ 스크립팅 ▸ Active 스크립팅 ▸ 사용 안함 옵션을 체크한다.
- 오페라에서 도구 ▸ 빠른 설정 ▸ 자바스크립트 사용 옵션을 체크하지 않는다.
- 사파리에서 편집 ▸ 기본설정 ▸ 보안 ▸ 자바스크립트 활성화 옵션을 체크하지 않는다.

- 특정 사진을 보여주는 플리커 페이지(다른 말로 하면, <http://flickr.com/photos/{user-id}/{photo-id}/>)에서, “플리커의 전체 기능을 사용하려면, 자바스크립트가 사용 가능한 브라우저를 사용하고 매크로미디어 플래시의 최신버전을 설치해야 합니다”라는 메시지를 볼 수 있을 것이다. 사진 상단의 모든 버튼들이 더 이상 작동하지 않으므로 사진을 편집하기 위해서는 제목, 설명, 그리고 태그를 클릭하는 대신에 편집이라는 링크를 클릭해야 한다.

■



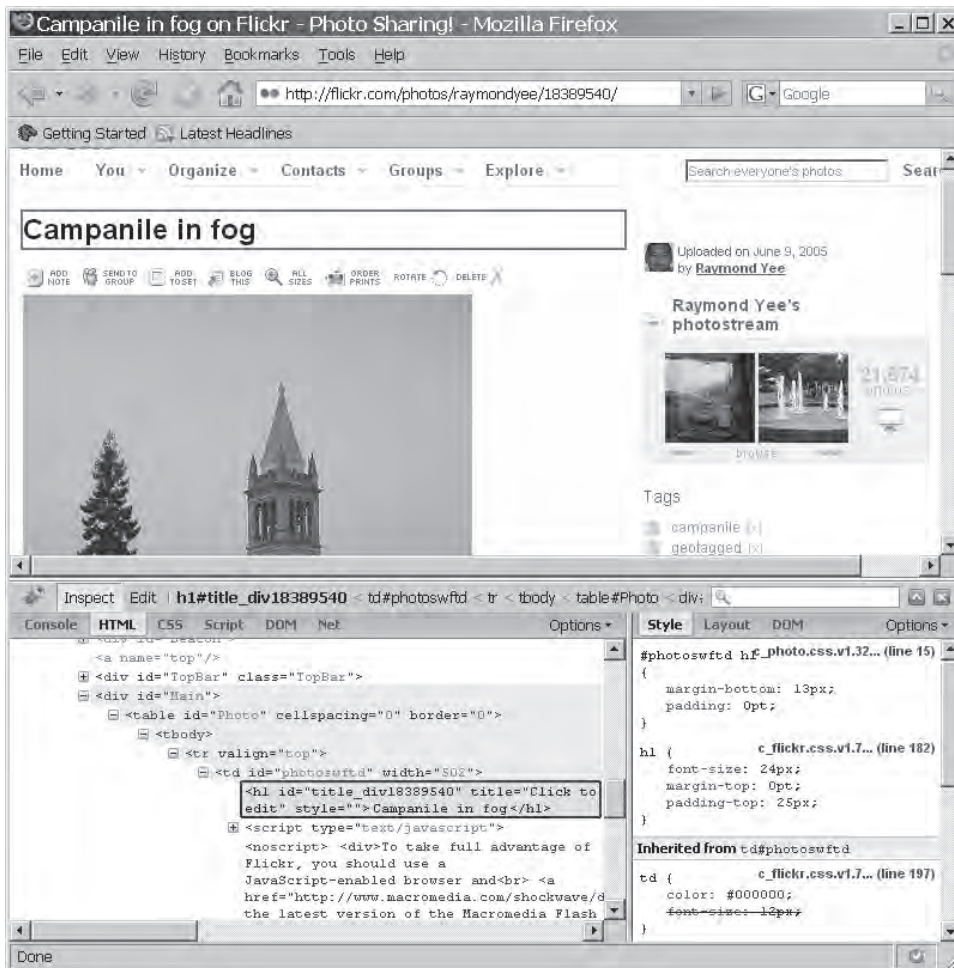
# 파이어버그, DOM 인스펙터, 그리고 자바스크립트 셀 에 대해서 알아보자

✱

## DOM 인스펙터 사용하기

[http://kb.mozillazine.org/DOM\\_Inspector](http://kb.mozillazine.org/DOM_Inspector)

## 파이어폭스에서 파이어버그 확장모듈 사용하기



- 실행 중인 소스를 볼 수 있다(즉, 원래 소스가 무엇인지가 아니라 해당 시점의 DOM을 구성하는 HTML을 볼 수 있다).
- HTML을 수정하여 바로 반영할 수 있다(파이어버그에서 HTML을 수정할 수 있으며 페이지에 적용된 변경내용을 확인할 수 있다).
- 요청 헤더와 응답 헤더를 볼 수 있으며, 이러한 헤더 정보는 디버깅 시에 매우 귀중하게 활용될 수 있다.

■

■

■

## 자바스크립트 웹 사용하기

**\* NOTE!** 이미 언급한 툴들 이외에 인터넷 익스플로러, 오페라, 그리고 사파리와 함께 사용할 수 있는 파이어버그 라이트가<sup>2</sup> 있으며 벤크만 자바스크립트 디버거Venkman JavaScript Debugger<sup>3</sup> 를 사용할 수도 있다

## 자바스크립트 라이브러리 사용하기

- 도조(Dojo)
- 모키킷(Mochikit)
- 프로토타입/스크립태쿨러스(Prototype/script.aculo.us)
- 야후! UI 라이브러리(YUI)

---

2 <http://www.getfirebug.com/lite.html>

3 <http://www.mozilla.org/projects/venkman/>

4 <http://simonwillison.net/2006/Jun/26/libraries/>

5 <http://www.openlaszlo.org/>

# YUI 위젯

## YUI 캘린더 사용하기

- API 관련 문서는 <http://developer.yahoo.com/yui/examples/calendar/index.html>를 참고하라.
- 위젯의 메서드 목록을 보려면 <http://developer.yahoo.com/yui/docs/YAHOO.widget.Calendar.html> 을 참고하라.

1. <http://developer.yahoo.com/yui/examples/calendar/quickstart.html> 에 접속하라.
2. 날짜를 클릭하라.
3. 선택한 날짜를 가져오기 위해 파이어버그나 자바스크립트 셸의 콘솔에 다음의 내용을 입력하도록 하자(그림 8-2를 보라).

```
YAHOO.example.calendar.cal1.getSelectedDates()
```

4. 캘린더가 어떻게 동작하는지 보기 위해 몇 가지 다른 메서드를 입력해보자.

Calendar Control: Quickstart Tutorial

Calendar Control: Quickstart Tutorial

This basic example walks you through the need to be included as well as the basic m

< August 2007 >

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

http://developer.yahoo.com - JavaScript

**JavaScript Shell 1.4**  
Features: autocompletion of property names with Tab, mu  
Values and functions: ans, print(string), props(object), blin  
Using bookmarklet version of shell: commands will run in c  
YAHOO.example.calendar.call.getSelectedDates()  
Fri Aug 31 2007 00:00:00 GMT-0700 (Pacific Daylight T  
YAHOO.example.calendar.call.hide() // Hides the co  
YAHOO.example.calendar.call.show() // Shows the c

<http://developer.yahoo.com/yui/treeview/>

1. [http://developer.yahoo.com/yui/examples/treeview/default\\_tree.html](http://developer.yahoo.com/yui/examples/treeview/default_tree.html) 에 접속하자.
2. 파이어버그나 자바스크립트 셸의 콘솔에서, 각각 트리를 펼치거나 접기위해 다음을 입력하도록 하라.

## YUI를 여러분의 서버호스트에 설치하기

1. <http://developer.yahoo.com/yui/download/> 에 접속하여 여러분이 호스팅을 받는 환경이나 여러분의 장비로 라이브러리를 다운로드하라.
2. 다운로드한 파일의 압축을 풀고 적절한 위치로 복사하도록 하라. 내 경우에는, 압축을 푼 뒤에 yui라는 이름의 디렉터리를 `/home/rdhyee/examples.mashupguide.net/lib/yui` 로 복사하였으며 이 디렉터리는 <http://examples.mashupguide.net/lib/yui/>의 yui 디렉터리에 매핑된다. 실행 시에 라이브러리에서 가장 중요한 부분은 `yui/build` 디렉터리이다.

<http://examples.mashupguide.net/lib/yui/examples/calendar/quickstart.html>

## 구글 맵에 대해 알아보자

3. 서비스 약관을 읽어보고<sup>10</sup> 약관에 동의한다면 테스트 파일을 위치시킬 디렉터리의 URL을 입력하도록 하라. 예를 들어, 내 경우에는 `http://examples.mashupguide.net/ch08` 이 URL이다. 신청결과로 발급받은 키를 적어두도록 하라.
4. 호스팅 디렉터리에 페이지를 만들고 HTML 코드를 붙여넣기 하도록 하라. 아래와 유사한 HTML 코드가 만들어질 것이다. 여러분의 API 키 값을 사용하도록 수정하자.<sup>11</sup>

---

8 <http://www.google.com/apis/maps/documentation/#Introduction>

9 <http://www.google.com/apis/maps/signup.html>

10 <http://www.google.com/apis/maps/terms.html>

11 <http://maps.google.com/maps/api/signup?url=http%3A%2F%2Fexamples.mashupguide.net%2Fch08>



5. 이제 `map` 변수 앞에 위치한 `var` 키워드를 제거하여 전역변수로 만들어 자바스크립트 셸에서 접근이 가능하도록 예제를 수정하자. 즉, 다음의 표현식

```
var map = new GMap2(document.getElementById("map"));
```

```
map = new GMap2(document.getElementById("map"));
```

---

12 <http://examples.mashupguide.net/ch08/google.map.htm>

13 <http://www.google.com/apis/maps/documentation/reference.html#GMap2>

```
map.getZoom()
```

---

---

```
map.getCenter()
```

---

---

```
map.setCenter(new GLatLng(37.872035, -122.257844), 13);
```

```
map.panTo(new GLatLng(37.872035, -122.257844));
```

```
map.addControl(new GSmallMapControl())
```

```
map.addControl(new GMapTypeControl())
```

```
window.kh = new GKeyboardHandler(map)
```

---

---

```
map.setZoom(0)
```

```
map.setZoom(17)
```

```
maptypes = map.getMapTypes()
```

---

---

```
map.getMapTypes()[0].getName()
```

---

---

```
map.getCurrentMapType()
```

---

---

```
map.getCurrentMapType().getName()
```

---

---

```
map.setMapType(maptypes[1]);
```

```
map.zoomIn()
```

```
map.zoomOut()
```

```
point = new GLatLng(37.87309185260284, -122.25508689880371)
```

---

---

```
marker = new GMarker(point)
```

---

---

```
map.addOverlay(marker);
```

```
GEvent.addListener(marker, 'click', function() {  
marker.openInfoWindowHtml('hello'); })
```

---

---

## 자바스크립트에서 플리커에 접근하기

1. 플리커 익스플로러에서 flickr.photos.search 페이지를 열도록 하고(<http://www.flickr.com/services/api/explore/?method=flickr.photos.search>), tag 파라미터를 flower, 그

---

14 <http://www.google.com/apis/maps/documentation/#Introduction>

15 <http://www.flickr.com/services/api/response.json.html>

`http://api.flickr.com/services/rest/?method=flickr.photos.search&api_key=<API_key>&tags=flower&per_page=3`

위 URL을 사용하면 이전 장들에서 자주 보았던 플리커 XML 결과 값을 얻을 수 있을 것이다.

2. 이제 URL의 끝에 `format=json` 이라는 파라미터를 추가하여 JSON 포맷으로 결과 값을 받도록 하자.<sup>16</sup> 다음과 같은 결과 값을 수신하게 될 것이며 읽기에 편하도록 정리하였다.

---

<sup>16</sup> `http://api.flickr.com/services/rest/?method=flickr.photos.search&api_key={api-key}&tags=flower&per_page=3&format=json`

3. 이제 플리커 응답의 다양한 부분에 접근하는 자바스크립트를 어떻게 작성하는지 테스트할 수 있도록 약간의 HTML과 자바스크립트를 작성해보자.



브라우저에서 이 페이지를 로딩하고 자바스크립트 셀을 호출하면 응답 데이터에 접근하는 몇몇 핵심 코드를 실행해볼 수 있다.

4. 이제 `jsonFlickrApi()`가 사진들을 화면에 표시하도록 해보자.

5. 이전 예제는 간단하지만 JSON 데이터를 로딩하면 즉시 `jsonFlickrApi()`를 호출한다는 면에서 제한이 있다. `jsoncallback` 파라미터를 사용하여 콜백함수의 이름을 사용자 정의할 수 있으며(예를 들어, `jsoncallback=MyHandler`와 같은 형태로) 가공되지 않은 JSON 데이터를 리턴받기 위해 `nojsoncallback=1` 을 파라미터로 사용할 수도 있다.<sup>17</sup>

---

<sup>17</sup> [http://api.flickr.com/services/rest/?method=flickr.photos.search&api\\_key={api-key}&tags=flower&per\\_page=3&format=json&nojsoncallback=1](http://api.flickr.com/services/rest/?method=flickr.photos.search&api_key={api-key}&tags=flower&per_page=3&format=json&nojsoncallback=1)

1. 아래 URL에 HTTP GET 요청을 생성한다.

`http://api.flickr.com/services/rest/?method=flickr.photos.search&api_key={api-key}&tags=flower&per_page=3&format=json&nojsoncallback=1`

2. 응답을 자바스크립트 오브젝트로 생성하여 원할 때 파싱하도록 하라.

## 그리즈명키를 사용하여 뉴욕 타임즈가 제공하는 퍼머링크에 접근하기

- <http://diveintogreasemonkey.org/>
- 마크 필그림Mark Pilgrim이 집필한 그리즈명키 분석(Greasemonkey Hacks, 오라일리 2005)
- <http://en.wikipedia.org/wiki/Greasemonkey>
- <http://www.greasespot.net/> (그리즈명키 프로젝트의 공식 블로그)



<http://www.nytimes.com/2007/04/04/education/04colleges.html>

<http://www.nytimes.com/2007/04/04/education/04colleges.html?ex=1333339200&en=3b7aac16a1ce4512&ei=5090&partner=rssuserland&emc=rss>

<http://nytimes.blogspot.com/genlink?q=http://www.nytimes.com/2007/04/04/education/04colleges.html>

<http://www.nytimes.com/aponline/us/AP-Imus-Protests.html>

<http://nytimes.blogspot.com/genlink?q=http://www.nytimes.com/aponline/us/AP-Imus-Protests.html>

1. 여러분이 뉴욕 타임즈 기사를 보여주는 페이지에 위치하고 있다면, 해당 링크를 뉴욕 타임즈 링크 생성기로 전송한다.
2. 만약 퍼머링크(첫 번째 <a> 태그의 href 어트리뷰트가 genlink가 아니라 http:로 시작한다면 퍼머링크가 존재함을 알 수 있다)가 존재하면, <ul id="toolsList">의 제일 끝에 새로운 <li> 엘리먼트를 추가한다.

1. 아직 설치하지 않았다면 그리즈명키 확장모듈을 설치하라.<sup>20</sup>
2. 다음의 두 가지 방법들 중에 하나를 사용하여 새로운 그리즈명키 스크립트를 생성하라.
  - <http://examples.mashupguide.net/ch08/newyorktimespermalinker.user.js>에 접속하여 설치할 수 있다.
  - 도구(Tools)->그리즈명키->새로운 유저스크립트(New User Script) 를 선택하고 이름/네임스페이스/설명/동작할 페이지에 값을 채워넣고 다음의 코드를 작성하도록 하자(영구적인 링크를 찾아내기 위해 GM\_xmlhttpRequest를 사용하는 것에 주의 하자. 10장에서 xmlhttpRequest에 숨겨진 로직에 대해서 보게 될 것이다).<sup>21</sup>

---

20 <https://addons.mozilla.org/en-US/firefox/addon/748>

21 [http://wiki.greasespot.net/GM\\_xmlhttpRequest](http://wiki.greasespot.net/GM_xmlhttpRequest) 와 [http://diveintogreasemonkey.org/api/gm\\_xmlhttprequest.html](http://diveintogreasemonkey.org/api/gm_xmlhttprequest.html)

3. 이제 <http://www.nytimes.com/2007/04/04/education/04colleges.html>에 접속하면 이 그리즈명키 스크립트가 동작하여 Share 버튼 아래에 퍼머링크가 새롭게 추가됨을 볼 수 있을 것이다. 퍼머링크를 찾아내야 하기 때문에 이 링크가 화면에 표시되기까지는 몇 초 정도 걸릴 것이다.

## 자바스크립트와 Ajax에 대해 좀 더 알아보기

- 데이비드 플래너건David Flanagan이 집필한 JavaScript, the Definitive Guide, Fifth Edition (오라일리, 2006 - '자바스크립트 완벽가이드'란 제목으로 인사이트에서 번역서를 출간했다)
  - 존 레식John Resig이 집필한 Pro JavaScript Techniques (Apress, 2006 - '프로 자바스크립트 테크닉'이란 제목으로 인사이트에서 번역서를 출간했다)
  - 셸리 파워즈Shelley Powers가 집필한 Learning JavaScript (오라일리, 2007 - '자바스크립트 for 웹 2.0'이란 제목으로 한빛미디어에서 번역서를 출간했다)
- 
- 사이트에서 제공하는 리소스에 대한 설명 페이지<sup>22</sup>와 그가 자바스크립트에 대해 소개한 내용을 보도록 하라.<sup>23</sup>
  - 피터-폴-코흐는 자신의 책(<http://www.quirksmode.org/book/>)에서 발췌한 예제 스크립트를 제공하고 있으며 <http://www.quirksmode.org/book/examplescripts.html> 에서 확인할 수 있다.

## 요약

---

<sup>22</sup> <http://www.quirksmode.org/resources.html>

<sup>23</sup> <http://www.quirksmode.org/js/intro.html>





3부

## 매쉬업 작성하기





## 09

---

API와 리믹스에 사용되는 요소에 대한  
내용에서 매쉬업 관련 내용까지  
알아보기

---

## ProgrammableWeb에 대한 소개

여러분은 이 사이트에서 많은 매쉬업의 목록을 볼 수 있다. 이 목록이 존재하는 모든 매쉬업을 담을 수 있을까?

아니다. 이 목록은 모든 매쉬업의 일부만을 정리한 것이다. 웹 매쉬업의 세계는 너무 방대하고 변화무쌍해서 한 장소에 카탈로그로 만들어 놓을 수가 없다. 그리고

ProgrammableWeb은 매쉬업이 무엇인가에 대해 대부분의 사람들이 동의하는 정의가 있다고 가정하고 이러한 정의를 기준으로 매쉬업의 카탈로그를 생성하였지만 실제로는 아직 모든 사람이 매쉬업이 무엇인가라는 질문에 대해 공통적인 기준을 가지고 있지는 않다. 이런 점을 감안하면, 현재 ProgrammableWeb은 가장 다양한 매쉬업 관련 데이터를 구조화 시켜놓은 사이트이다.

## ProgrammableWeb에서 사용자가 생성한 데이터

<http://www.programmableweb.com/register>

<http://www.programmableweb.com/add>

**매쉬업 관련 디렉터리 서비스가 유지될 수 있을까?**

## 전반적인 매쉬업 현황에 대해 알아보기

<http://www.programmableweb.com/tour>

<http://www.programmableweb.com/mashups>

The screenshot shows the ProgrammableWeb.com website interface. The top navigation bar includes links for Sign up/Sign in, Add a Link, Subscribe, Tag Cloud, Google Custom Search, Advertise on PW, Contact Us, Press, About, and FAQ. The main header features the ProgrammableWeb logo, an Adobe AIR banner, and a navigation menu with Home, News, Mashups, APIs, Verticals, How-To, Contests, and Members. Below the navigation menu is a secondary menu with Dashboard, Directory, Newest, Most Popular, Mashup Matrix, Tag Cloud, and Random.

The main content area is divided into several sections:

- Subscribe:** Includes a link to "All New Mashups" and statistics for the Mashup Directory (Total Mashups Listed: 2664, Past 7 Days: 19, Past 30 Days: 78).
- Mashup Dashboard:** Features a "Mashup of the Day, January 14, 2008" section highlighting "AOLMail for Facebook" with a description and tags (email, social).
- Top Mashup Tags:** A pie chart showing the distribution of tags for the last 14 days. The tags and their percentages are: mapping (27%), shopping (11%), video (11%), social (10%), photo (8%), rss (6%), realstate (6%), music (6%), search (5%), and enterprise (5%).
- Popular Directory Searches:** A list of popular search terms including Celebrity Mashups, Video Mashups, Popular New Mashups, All Popular Mashups, Maps Mashups, Photo Mashups, Shopping Mashups, Sports Mashups, Government Mashups, and Mashups.
- Newest Mashups:** A list of the latest mashups, including AOLMail for Facebook, Home Valuations, Goboz, SmartRatings, NetVibes Slideshow, and iMustix.
- Popular New Mashups:** A list of popular new mashups, including The Orion Atlas, Custom iGoogle Skins v2, Federal Contracts by State, Trip Organizer, Access Denied Map, and How Far Does Santa Have To Travel.
- Sponsors:** A section featuring various sponsors and their services, including mash-up & develop, VOICE MASHUPS, Got maps? Get cash., Widget Overlay, openkapow, Connect. Collect. Mashup. Everything!, and Be a rocket scientist.

- 
- 

## 매쉬업 디렉터리

<http://www.programmableweb.com/mashups/directory>

<http://www.programmableweb.com/mashups/directory/{page}?sort={sort}&view={view}>

<http://www.programmableweb.com/mashups/directory/5?sort=popular&view=desc>

<http://www.programmableweb.com/popular>

## 매쉬업을 추적하기 위해 피드 사용하기

<http://feeds.feedburner.com/programmableweb/mashup>

<http://feeds.feedburner.com/ProgrammableWeb>

<http://feeds.feedburner.com/programmableweb/apis>

## 매쉬업을 설명하기 위한 태그의 사용

<http://www.programmableweb.com/add>

<http://www.programmableweb.com/mashups>



<http://www.programmableweb.com/search>

<http://www.programmableweb.com/mashups/directory>

- 지도mapping
- 사진photo
- 쇼핑shopping
- 검색search
- 여행travel
- 비디오video
- 뉴스news
- 스포츠sports
- 부동산realestate
- 메시징messaging

<http://www.programmableweb.com/tag/{tag}>

<http://www.programmableweb.com/tag/mapping>

<http://www.programmableweb.com/tag/{tag}/{page}?sort={sort}&view={view}>

<http://www.programmableweb.com/tag/mapping/2?sort=date&view=desc>

<http://www.programmableweb.com/apitag/mapping>



<http://www.programmableweb.com/tag/mapping>

<http://www.programmableweb.com/apis/directory/1?sort=category>

## API 와 매쉬업 버티컬

<http://www.programmableweb.com/markets>

- <http://www.programmableweb.com/shopping>
- <http://www.programmableweb.com/government>
- <http://www.programmableweb.com/mapping>
- <http://www.programmableweb.com/telephony>
- <http://www.programmableweb.com/social>
- <http://www.programmableweb.com/video>

<http://www.programmableweb.com/featured/telephony-mobile-apis-and-mashups>

## 특정 매쉬업 프로필에 대해 알아보기

<http://www.programmableweb.com/mashup/{mashup-handle}>

<http://www.programmableweb.com/mashup/flash-earth>

■

■

■

■

■

■

■

■

<http://www.flashearth.com/>

## 특정 API를 사용하는 매쉬업 찾아내기

<http://www.programmableweb.com/apis/directory/1?sort=mashups>

<http://www.programmableweb.com/api/google-maps>

<http://www.programmableweb.com/api/google-maps/mashups>

## 매쉬업을 사용하여 해결할 수 있는 문제

## 관심있는 책 추적하기

- 
- 
- 
- 
- 
- 
- 
- 
- 

- 
- 

- 지역 도서관에 있는 책을 찾기 위해 사용하는 <http://worldcatlibraries.org/>
- 지역 도서관(버클리 시립 도서관과 UC 버클리의 도서관)의 온라인 도서 카탈로그
- 읽고 있는 책에 대한 내용을 입력하고 다른 사람들이 무엇을 읽는지를 알아보는 데 사용하는 라이브러리씽LibraryThing
- 참고서적을 찾기 위해 사용하는 파이어폭스 확장기능인 조테로(<http://zotero.org>)

- 반즈&노블(bn.com)과 기타 온라인 서점
- 중고책을 구매할 때 사용하는 <http://www.half.ebay.com/>

- 

- 

- 

- 

<http://www.programmableweb.com/mashups/directory/1?q=books&sort=popular>

<http://www.programmableweb.com/search>

<http://www.programmableweb.com/tag/books/1?sort=popular>

<http://www.programmableweb.com/tag/books/1?sort=date>

<http://www.programmableweb.com/tag/books/1?sort=popular>

<http://www.programmableweb.com/api/amazon-ecommerce>

<http://www.programmableweb.com/tag/books/1?apis=Amazon+eCommerce&sort=popular>

## 고급검색을 사용하여 매쉬업과 API 검색하기

고급검색을 사용하면 매쉬업과 API 프로필을 검색할 수 있다. 매쉬업을 검색하려면, 아래에 접속하자.

<http://www.programmableweb.com/mashups/directory>

Advanced Search 링크를 클릭하면 고급검색 옵션이 화면에 나타날 것이다. 고급검색 옵션을 사용하면, 다음과 같은 조건을 명시할 수 있다.

- 매쉬업에 사용되는 API를 세 개까지 명시할 수 있다(API의 이름을 입력하면 자동완성 기능이 이름을 채워준다.).
- 매쉬업과 관련이 있는 태그를 세 개까지 명시할 수 있다.
- 몇 가지 옵션 중에 하나를 선택하여 매쉬업 프로필이 언제 생성되었는지를 명시할 수 있다.

API를 검색하려면, 먼저 아래에 접속하라.

<http://www.programmableweb.com/apis/directory>

Advanced Search 링크를 클릭하면 고급검색 옵션이 화면에 나타날 것이다. 고급검색 옵션을 사용하면, 다음과 같은 조건을 명시할 수 있다.

- API와 연관된 태그를 세 가지까지 명시할 수 있다
- 카테고리
- 회사
- 프로토콜(예를 들어, REST 또는 자바스크립트)
- 몇 가지 옵션 중에 하나를 선택하여 API 프로필이 언제 생성되었는지를 명시할 수 있다.

<http://www.programmableweb.com/mashup/blueorganizer>

<http://www.adaptiveblue.com/>

[http://www.adaptiveblue.com/smartlinks\\_books.html](http://www.adaptiveblue.com/smartlinks_books.html)

■

■

## 구루립, 북범프, 그리고 라이브러리씽과 유사한 서비스들

<http://www.nytimes.com/2007/03/04/business/yourmoney/04novel.html>

■ <http://www.programmableweb.com/mashup/gurulib>

■ <http://www.programmableweb.com/mashup/bookbump>



## 도서 매쉬업에 대한 몇 가지 결론

- 
- 
- 
- ProgrammableWeb을 사용하는 것만으로는 다루지 못하는 사각지역이 존재한다. 예를 들어, 내 도서관의 최신 책들을 찾아보려면 <http://www.berkely-public.org/ftlist>에서 최근 입고된 도서를 찾아보아야 하지만 API를 제공하지 않는다.

## 언제 항공기 티켓을 구매해야 하는지 알아보기

<http://www.programmableweb.com/tag/travel>

<http://blog.programmableweb.com/2007/10/29/5-travel-apis-from-comparison-to-booking/>

<http://www.programmableweb.com/apis/directory/1?q=travel>

<http://www.programmableweb.com/apis/directory/1?q=travel&sort=mashups>

- <http://www.programmableweb.com/api/yahoo-travel>
- <http://www.programmableweb.com/api/kayak>

<http://www.kayak.com/labs/api/search/>

<http://www.programmableweb.com/mashup/kayak.com-deals-from-cell-phone>

## 꿈에 그리던 집 찾기

<http://www.programmableweb.com/tag/realestate/1?view=desc>

<http://www.programmableweb.com/apitag/realestate/1?sort=mashups>

- 질로우(Zillow, <http://www.programmableweb.com/api/zillow>)
- 트룰리아(Trulia, <http://www.programmableweb.com/api/trulia>)

<http://www.trulia.com/rss2/CA/Berkeley/>

<http://www.programmableweb.com/mashups/directory/1?apis=trulia%2Czillow>

<http://www.programmableweb.com/tag/realestate?apis=Google+Base>

## 최신뉴스 맵핑하기

<http://www.programmableweb.com/tag/news/1?view=desc>

- <http://www.programmableweb.com/mashup/bbc-news-map>은 이제 존재하지 않는 매쉬업으로 영국의 BBC 뉴스를 지도와 매쉬업한 것이다.
- <http://www.programmableweb.com/mashup/ap-national-news-google-maps>는 <http://www.81nassau.com/apnews/>를 가리키며 AP통신사의 피드(국내뉴스, 스포츠, 그리고 비즈니스 분야를 포함하여)로부터 선택된 기사들을 구글 맵과 매쉬업하여 보여 준다.
- <http://www.programmableweb.com/mashup/mapified-rss> 는 <http://o.gosselin.free.fr/Projects/MapifiedRss/>를 가리키며 미리 설정된 RSS피드(예를 들어, 로이터, AP 통신의 헤드라인이나 구글 뉴스) 중의 항목 또는 사용자가 입력한 피드 URL의 항목을 구글 맵과 맵핑한다.

<http://pipes.yahoo.com/pipes/docs?doc=operators#LocationExtractor>

<http://pipes.yahoo.com/raymondye/locationextractor>

<http://www.nytimes.com/services/xml/rss/nyt/International.xml>

[http://pipes.yahoo.com/pipes/pipe.run?InputURL=http%3A%2F%2Fwww.nytimes.com%2Fservices%2Fxml%2Frss%2Fnyt%2FInternational.xml&\\_id=cInT4D7B3BGmoxPNiXrL0A&\\_render=kml](http://pipes.yahoo.com/pipes/pipe.run?InputURL=http%3A%2F%2Fwww.nytimes.com%2Fservices%2Fxml%2Frss%2Fnyt%2FInternational.xml&_id=cInT4D7B3BGmoxPNiXrL0A&_render=kml) ➡

<http://tinyurl.com/yvx8qy>

[http://maps.google.com/maps?f=q&hl=en&geocode=&time=&date=&ttype=&q=http%2F%2Fpipesyahoo.com%2Fpipes%2Fpipe.run%3FInputURL%3Dhttp%253A%252F%252Fwww.nytimes.com%252Fservices%252Fxml%252Frss%252Fnyt%252FInternational.xml%26\\_id%3DcInT4D7B3BGmoxPNiXrL0A%26\\_render%3Dkml&ie=UTF8&dl=28.921631,53.4375&spn=150.976999,360&z=2&om=1](http://maps.google.com/maps?f=q&hl=en&geocode=&time=&date=&ttype=&q=http%2F%2Fpipesyahoo.com%2Fpipes%2Fpipe.run%3FInputURL%3Dhttp%253A%252F%252Fwww.nytimes.com%252Fservices%252Fxml%252Frss%252Fnyt%252FInternational.xml%26_id%3DcInT4D7B3BGmoxPNiXrL0A%26_render%3Dkml&ie=UTF8&dl=28.921631,53.4375&spn=150.976999,360&z=2&om=1)

<http://tinyurl.com/yp8k2b>

<http://www.programmableweb.com/mapping>

요약

# 10

---

다양한 서비스로부터  
매쉬업 생성하기

---

■

■

설계



■

■

■

■

■

■

■

■

■

✱

## 배경지식: 플리커에서의 위치정보 태깅

namespace:predicate=value

<http://flickr.com/photos/raymondye/18389540/>

geo:lon=-122.257704

geo:lat=37.8721

`http://api.flickr.com/services/rest/?method=flickr.tags.getListPhoto&api_key={api_key}&photo_id=18389540`

**\* NOTE!** geo:lat와 geo:lon 태그가 새로운 태그로 대체되었는데 왜 위도와 경도에 대한 시스템 태그를 가져오는지 궁금할 수도 있다. 이러한 기법을 보여주는 것은 역사적인 흥미로움 때문이기도 하고 오래된 소프트웨어들이 이것을 사용하기 때문이다(GMiF 그리즈 명키 스크립트와 같은 소프트웨어는 예전 스타일의 위치정보 태깅을 사용한다).

- 주어진 지도영역 내에서 사진을 검색할 수 있다. 위치정보가 태깅된 사진전체를 크롤링하여 사진과 위치에 대한 데이터베이스를 구성하고 이러한 데이터베이스를 사용하여 위치정보를 검색하는 시스템을 구성하지 않는 한 정기적인 태그를 사용하여 이러한 일을 수행할 방법은 없다. 플리커가 이러한 일을 여러분 대신에 처리해준다.
- 위치정보를 노출시킬 것인지 아닌지를 사진과는 무관하게 제어할 수 있다(즉, 사진은 화면에 노출되도록 하고 위치정보는 노출시키지 않을 수 있다). 예전 스타일의 위치정보 태깅에서는 만약 사진이 노출된다면 해당 사진의 태그 또한 노출되었기 때문에 geo:lat/geo:lon 태그가 화면에 표시되었었다.
- 새로운 스타일의 위치정보 태깅 방법은 위치정보를 태깅하는 공식적인 방법이다. 반면에 예전 스타일의 태깅 방법은 플리커의 공식적인 지원을 받은 적이 없다. 공식적인 방법이 지원되면서 위치정보가 포함된 사진을 지도에 링크해주는 플리커 맵 등의 많은 기능들이 제공되었다.

<http://flickr.com/photos/raymondye/18389540/>

[http://api.flickr.com/services/rest/?method=flickr.photos.geo.getLocation&api\\_key={api\\_key}&photo\\_id=18389540](http://api.flickr.com/services/rest/?method=flickr.photos.geo.getLocation&api_key={api_key}&photo_id=18389540)

```
>>> rsp = flickr.client.photos_geo_getLocation(photo_id=18389540)
```

```
>>> rsp = flickr.client.photos_geo_removeLocation(photo_id=18389540)
```

```
>>> rsp = flickr.client.photos_geo_setLocation(photo_id=18389540,lat=37.8721,lon=-  
122.257704,accuracy=16)
```

```
http://www.flickr.com/services/api/flickr.photos.search.html
```

```
bbox=lon0,lat0,lon1,lat1
```

SW: 37.81778516606761, -122.34374999999999

NE: 37.92619056937629, -122.17208862304686

`http://api.flickr.com/services/rest/?api_key={api_key}&method=flickr.photos.search&bbox=-180%2C-90%2C180%2C90&min_upload_date=820483200&accuracy=1`

`http://api.flickr.com/services/rest/?api_key={api_key}&method=flickr.photos.search&bbox=-122.34374999999999%2C+37.81778516606761%2C+-122.17208862304686%2C+37.92619056937629&min_upload_date=820483200&accuracy=1`

배경지식: XMLHttpRequest와 라이브러리 포함하기

## XMLHttpRequest를 직접 사용하기

<http://examples.mashupguide.net/ch10/weather.php?p=94720>

<http://examples.mashupguide.net/ch10/xhr.html>





- 이 코드는 XHR의 인스턴스를 생성할 때까지 여러 가지 방법을 사용하여 XHR의 초기화를 시도하며 이러한 모든 방법이 제대로 동작하지 않는다면 실패하게 된다.

- 아래와 같은 코드를 통해

```
xmlhttp.onreadystatechange = writeResults;
```

- xmlhttp.responseXML은 XML DOM의 형태로 HTTP 응답의 바디를 반환한다.

## 야후 커넥션 매니저 사용하기

<http://developer.yahoo.com/yui/connection/>

<http://developer.yahoo.com/yui/examples/connection/index.html>

<http://developer.yahoo.com/yui/examples/connection/weather.html>

<http://examples.mashupguide.net/lib/yui/examples/connection/weather.html>

<http://examples.mashupguide.net/lib/yui/examples/connection/assets/weather.php>

<http://examples.mashupguide.net/lib/yui/examples/connection/assets/weather.php?p=94720>

<http://xml.weather.yahoo.com/forecastrss?p=94720>





## 서버 측 프록시 작성하기

XHR을 사용하여 직접 API를 호출하면 어떤 일이 발생할까?

---

1 <http://examples.mashupguide.net/ch10/direct.connect.html>



Error: uncaught exception: Permission denied to call method XMLHttpRequest.open

<http://developer.yahoo.com/javascript/howto-proxy.html>

```
var sUrl = "http://xml.weather.yahooapis.com/forecastrss?p=94720";
```

```
var sUrl = "./weather.php?p=94720";
```

**위치정보가 포함된 사진을 가져오는 서버 측 스크립트 작성하기**

<http://www.flickr.com/services/api/flickr.photos.search.html>

<http://examples.mashupguide.net/ch10/flickrgeo.php>

<http://examples.mashupguide.net/ch10/flickrgeo.php.txt>

- user\_id
- tags
- tag\_mode
- text
- min\_upload\_date
- max\_upload\_date
- min\_taken\_date
- max\_taken\_date
- license
- sort
- privacy\_filter
- accuracy

- safe\_search
- content\_type
- machine\_tags
- machine\_tag\_mode
- group\_id
- place\_id
- extras
- per\_page
- page

- rest는 플리커 API로부터 기본형태의 결과 값을 리턴한다.
- json은 플리커 API로부터 JSON 형태의 결과 값을 리턴한다.
- html은 HTML 폼과 사진 목록을 리턴한다.
- kml은 검색결과를 KML로 리턴한다(보다 자세한 내용은 13장을 보라).
- nl은 결과 값을 KML 네트워크 링크로 리턴한다(보다 자세한 내용은 13장을 보라).



`http://examples.mashupguide.net/ch10/flickrgeo.php?tags=cat&lat0=-90&lon0=-180&lat1=90&lon1=180&page=1&per_page=10&o_format=html`

## 간단한 클라이언트 측 프레임워크 생성하기

- 

- 

엘리먼트를 읽고 작성하기

---

---

---

---

---

---

---

3 <http://examples.mashupguide.net/ch10/dom.html>  
4 <http://examples.mashupguide.net/ch10/square1.html>

document

---

---

document.form[0].innerHTML

---

---

document.form[0].elements[0].value

---

---

`document.form[0].num.value`

---

---

`span.document.getElementById('answer')`

---

---

`Span.document.getElementById('answer').innerHTML = 16`

---

---

**간단한 이벤트를 입력 폼과 연결하고 폼 전송의 결과를 화면에 표시하기**

## 클라이언트 측 프레임워크를 플리커에 연결하기

- 폼 파라미터들을 flickrgeo.php에 대한 쿼리로 변환한다.
- flickrgeo.php에 요청을 수행하기 위해 XHR을 사용하고 결과로 수신한 JSON 응답 결과를 화면에 표시한다.
- 이렇게 수신한 JSON을 HTML로 변환하여 화면에 표시한다.



flickrgeo.php에 쿼리하는 URL 작성하기

JSON 형태의 데이터를 읽기 위해 YUI 커넥션 매니저를 통해 XHR 사용하기





- `<script>`엘리먼트는 커백션 매니저를 사용하기 위해 야후! UI 라이브러를 포함시킨다
- `callback` 오브젝트에 의해 참조되며 `flickrgeo.php`에 대한 성공적인 호출과 실패한 호출을 각각 처리하는 콜백 함수(`handleSuccess`와 `handleFailure`)에 대한 정의. 만약 호출이 성공한다면, `flickrgeo.php`로부터 JSON 결과 값이 `<div id="pics"></div>`에 추가된다.
- `var request = YAHOO.util.Connect.asyncRequest('GET', url, callback);` 라인의 야후! 커백션 매니저에 대한 호출. HTTP GET 요청이 `url`에 수행되며 HTTP 응답이 `callback` 오브젝트에 포함된 함수에 리턴된다.

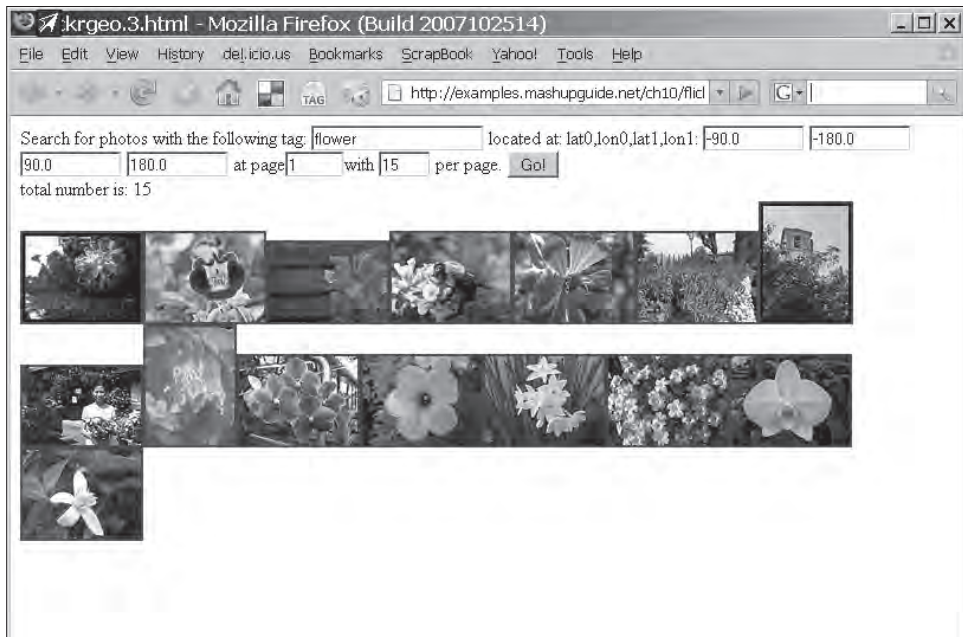
## JSON을 HTML로 변환하기

---

8 <http://www.json.org/js.html>

9 <http://examples.mashupguide.net/ch10/flickrgeo.3.html>





## 플리커와 구글 맵 API를 매쉬업하기

1. 기본 구글 맵을 설정한다.
2. 지도가 표시되는 영역이 변경될 때 지도가 반응하도록 설정한다.
3. 코드를 하나의 파일로 합쳐서 플리커와 구글 맵이 동일한 HTML에 표시되도록 한다.  
아직은 각각의 코드가 함께 존재만 할 뿐 연동되지는 않는다.
4. 화면에 표시되는 구글 맵의 영역을 나타내는 사각형에서 위도와 경도의 좌표를 얻어서 처리하도록 연결한다.
5. 플리커 사진 검색에 사용되는 입력 항목인 lat0/lon0와 lat1/lon1에 좌표를 표시한다.
6. 사진들이 지도상에 표시되도록 처리한다.

## 기본 구글 맵 설정하기

1. 먼저 도메인에 필요한 구글 맵의 키를 생성해야 한다. 내가 가지고 있는 도메인은 `http://examples.mashupguide.net/ch10`이다. 아래 URL에서 이에 해당하는 API의 키 값을 계산할 수 있다.

`http://www.google.com/maps/api/signup?url=http%3A%2F%2Fexamples.mashupguide.net%2Fch10`

2. 크기를 조절할 수 있는 컨트롤, 맵의 유형을 선택할 수 있는 컨트롤, 그리고 화살표 키들을 사용하여 맵을 제어할 수 있는 키보드 핸들러가 포함된 지도를 UC 버클리를 중심으로 위치시키기 위해 아래 코드를 복사하고 여러분의 키 값으로 대체하라.<sup>10</sup>



---

<sup>10</sup> `http://examples.mashupguide.net/ch10/gmap.1.html`

## 표시되는 영역이 변경될 때 지도가 반응하도록 설정하기

[http://www.google.com/apis/maps/documentation/#Events\\_overview](http://www.google.com/apis/maps/documentation/#Events_overview)

<http://www.google.com/apis/maps/documentation/reference.html#GMap2>

```
onMapMoveEnd = function () {alert("You moved or zoomed the map");}
```

---

---

```
GEvent.addListener(map,'moveend', onMapMoveEnd);
```

---

---



## 플리커와 구글 맵의 코드 합치기

<http://examples.mashupguide.net/ch10/gmapflickr1.html>

## 구글 맵의 지도 표시 영역을 연결하기







## 사진들을 지도상에 표시하기







- 현재 지도의 경계선을 표시하는 <div id="mapinfo">을 제거하는 등의 작업을 통해

매쉬업의 룩앤필look and feel을 세련되게 다듬는다(지도의 경계선을 표시하는 <div id="mapinfo">의 제거를 포함하여).

- 마커를 클릭할 때 표시되는 윈도우가 지도를 이동시키지 않도록 처리한다.
- 플리커 맵 인터페이스에서 처리되는 것과 같이 동일한 위치정보를 가진 사진을 그룹으로 묶는다.

## 플리커 사진을 표시해주는 구글 맵플릿

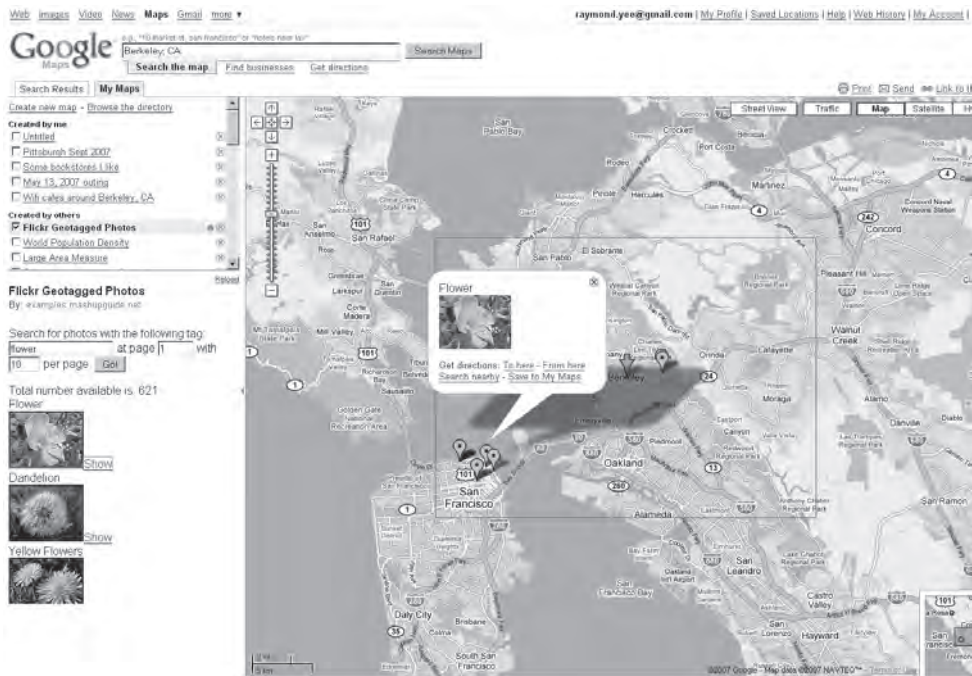
<http://www.google.com/apis/maps/documentation/mapplets/>

<http://examples.mashupguide.net/ch10/flickr.mapplet.xml>

## 구글 My Maps에 맵플릿 추가하기

<http://maps.google.com/> 에 접속하도록 하고 만약 로그인되어 있지 않다면 구글 맵에 로그인하도록 하자.

1. My Maps 탭을 클릭하라.
2. Browse the Directory 버튼이나 링크를 클릭하도록 하라.
3. '구글 지도 콘텐츠 검색' 버튼의 우측에 있는 'URL로 추가' 링크를 클릭하도록 하라.
4. 맵플릿 소스의 URL(예를 들어, <http://examples.mashupguide.net/ch10/flickr.maplet.xml>)을 입력하고 '추가' 버튼을 클릭하자.
5. 'Google Maps로 돌아가기'를 클릭하라.
6. 이제 My Maps 탭 아래의 Created by Others 란에서 "Flickr Geotagged Photos" 항목을 볼 수 있을 것이다. 체크박스를 사용하면 맵플릿을 사용하거나 중지시킬 수 있다.









- flickrgeo.php와 같은 서버 측 프록시를 생성하지 않고 코드상에서 플리커 API에 대한 URL에 직접 접근하기 때문에 \_IG\_FetchContent() 메서드가 플리커 API에 접근하는 부분이 상당히 간결하다.

- 맵플릿이 사이드바에 콘텐츠를 표시할 영역을 충분히 제공하고 있지 않으므로 작은 공간을 활용할 수 있도록 맵플릿을 좀 더 최적화해야 한다.

## 요약



# 11

---

## 툴을 사용하여 매쉬업 생성하기

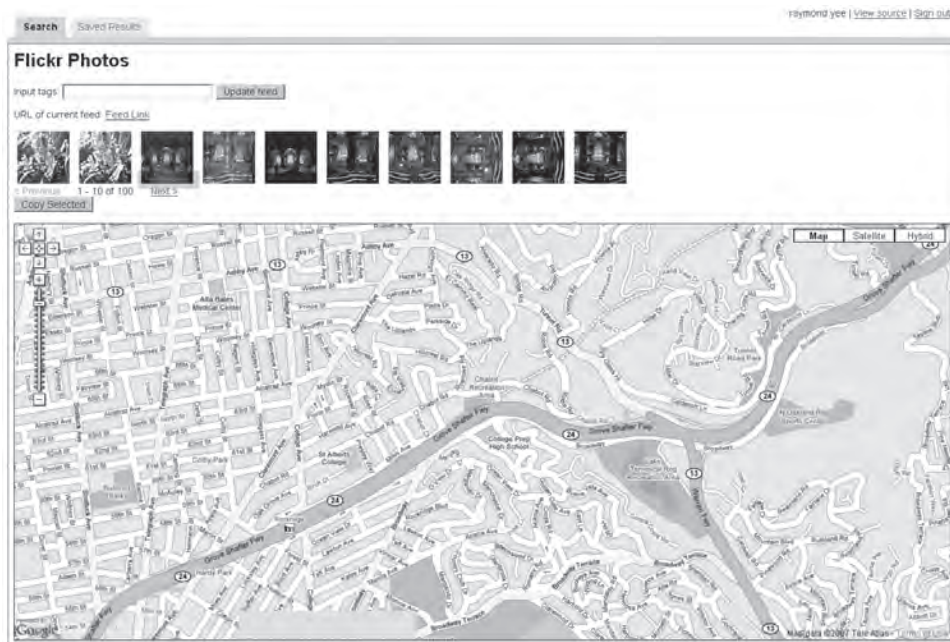
---

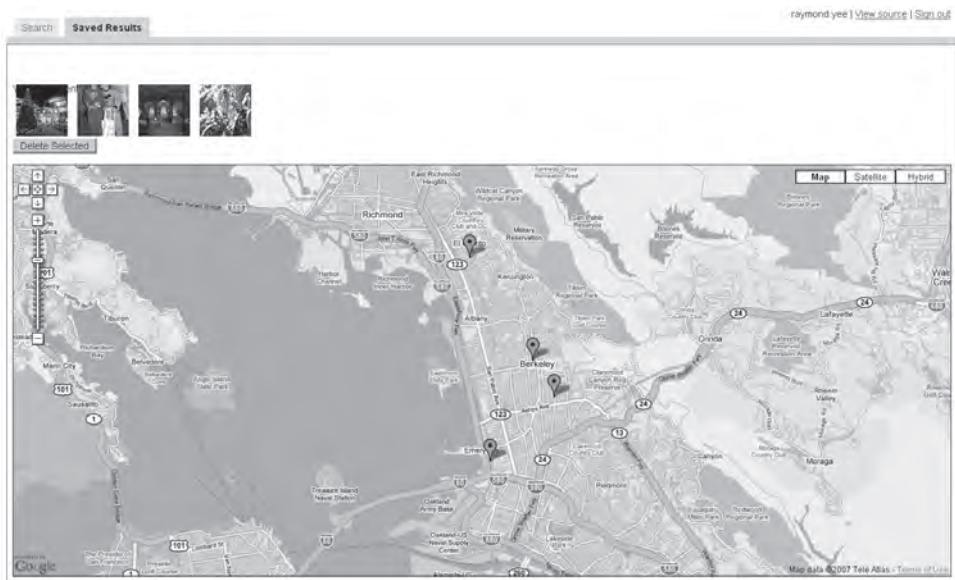
- 
-

매쉬업 툴이 해결해주는 문제

## 이번 장에서 개발할 것은 무엇인가

<http://mashup-raymond-yee-flickrfeed4.googlemashups.com/>





매쉬업의 생성 : 단계별로 예제를 개발해보자



- 
- 
- 
- 
- 
- 
- 
- 

## 구글 매쉬업 에디터에 익숙해지기

<http://code.google.com/gme/>

**\* NOTE!** 이 책이 출판될 때까지, 구글 매쉬업 에디터는 여전히 클로즈 베타 상태이다.

<http://editor.googlemashups.com/docs/reference.html>

피드를 읽고 화면에 표시하기(심플 템플릿)

**\* NOTE!** 현재, 구글 매쉬업 에디터는 RSS2.0과 Atom 피드 이외의 XML 데이터는 읽지 못한다. 여러분은 야후! 파이프를 사용하여 XML을 RSS 2.0으로 변환한 후에 구글 매쉬업 에디터에서 사용할 수 있다.

[http://api.flickr.com/services/feeds/geo/?g=34427469792@N01 &lang=en-us&format=rss\\_200](http://api.flickr.com/services/feeds/geo/?g=34427469792@N01&lang=en-us&format=rss_200)



사용자 정의 템플릿 사용하기



<http://mashup-raymond-yee-flickrfeed1.googlecode.com/svn/trunk/index.gml>

<http://mashup-raymond-yee-flickrfeed1.googlemashups.com/>

- 
- 
- 
- 

#### 서브버전(SVN)을 사용하여 프로젝트에 접근하기

브라우저 기반의 에디터를 사용하여 코드를 편집하는 대신에, 서브버전([http://en.wikipedia.org/wiki/Subversion\\_\(software\)](http://en.wikipedia.org/wiki/Subversion_(software)))을 사용하면 코드를 가져와 수정을 한 뒤에 수정된 내용을 체크인하는 형태로 작업할 수 있다. 구글 매쉬업 에디터에서 어떻게 SVN을 사용하는지 설명한 문서는 아래 URL에서 찾을 수 있다.

<http://code.google.com/support/bin/answer.py?answer=76145&topic=11689>

구글 매쉬업 에디터를 사용하여 생성한 매쉬업 프로젝트들은 구글 코드에 의해 호스팅되는 개별적인 프로젝트로 생성된다. 구글 코드 내에서 이러한 프로젝트들의 목록을 찾아보려면, 다음 URL을 방문하도록 하라.

<http://code.google.com/hosting/>

서브버전을 사용할 때 필요한 구글 코드의 패스워드를 확인하려면 Settings를 클릭하도록 하라. 여러분이 생성한 프로젝트의 목록을 보려면 My Profile 탭을 클릭하라. 내 경우에는, My Profile 탭을 클릭하면 다음 URL로 연결된다.

<http://code.google.com/u/raymond.yee/>

특정 프로젝트에 대해 알아보도록 하자. 내가 구글 매쉬업 에디터를 통해 생성한 프로젝트 중의 하나는 다음 URL에서 찾아볼 수 있다.

<http://code.google.com/p/mashup-raymond-yee-flickrfeed2/>

위 프로젝트의 코드는 다음 URL에서 찾아볼 수 있다.

<http://mashup-raymond-yee-flickrfeed2.googlecode.com/svn/>

이 프로젝트의 index.gml 코드는 아래에서 확인할 수 있다.

<http://mashup-raymond-yee-flickrfeed2.googlecode.com/svn/trunk/index.gml>

기본 설정에 의해 여러분이 구글 매쉬업 에디터를 통해 생성한 코드는 아파치 2.0 라이선스를 따르도록 설정된다는 것에 주목하라. 이 설정은 Administrator 탭에서 변경할 수 있다.

<http://code.google.com/p/mashup-raymond-yee-flickrfeed2/admin>

SVN을 사용하여 코드를 체크아웃하는 방법은 아래 URL에서 볼 수 있다.

<http://code.google.com/p/mashup-raymond-yee-flickrfeed2/source>

일단 코드를 체크아웃 하였으면, 에디팅에 웹 브라우저만을 사용할 수 있는 것이 아니라 여러분이 즐겨 사용하는 데스크톱용 에디터를 사용할 수 있다. 게다가, 데스크톱용 에디터를 사용하면 브라우저 기반 에디터를 사용할 때 한 번에 하나의 파일만 수정할 수 있는 것과는 달리 동시에 여러 파일을 보다 쉽게 수정할 수 있다.

## 플리커 접근 시 야후! 파이프 사용하기

[http://pipes.yahoo.com/raymondye/flickr\\_photos\\_search](http://pipes.yahoo.com/raymondye/flickr_photos_search)

[http://pipes.yahoo.com/pipes/pipe.info?\\_id=YG9eZGWO3BGukZGJTqoASA](http://pipes.yahoo.com/pipes/pipe.info?_id=YG9eZGWO3BGukZGJTqoASA)

- 
- 
- 
- 

- 각 항목에 대해 `item.image_prefix`를 생성하는 연산. `item.image_prefix`는 다양한 크기의 플리커 이미지를 가리키는 URL의 앞부분으로 사용된다.
- `item.image_prefix`와 `_s`를 연결하여 `item.image_small_URL`을 생성하는 두 번째 연산
- 사진을 표시하는 플리커 페이지에 대한 링크를 나타내는 `item.link`를 계산하고 할당하는 세 번째 연산
- 플리커 사진의 작은 정사각형 버전을 표시하는 HTML을 포함한 `item.description`을 계산하고 할당하는 네 번째 연산

[http://pipes.yahoo.com/pipes/pipe.run?\\_id=YG9eZGWO3BGukZGJTqoASA&\\_render=rss](http://pipes.yahoo.com/pipes/pipe.run?_id=YG9eZGWO3BGukZGJTqoASA&_render=rss) ➡  
&api\_key={api\_key}&extras=geo&lat0=37.817785166068&lat1=37.926190569376&lon0=-  
122.34375&lon1=-122.17208862305&min\_upload\_date=820483200&per\_page=10

[http://pipes.yahoo.com/pipes/pipe.run?\\_id=YG9eZGWO3BGukZGJTqoASA&\\_render=kml](http://pipes.yahoo.com/pipes/pipe.run?_id=YG9eZGWO3BGukZGJTqoASA&_render=kml)

#### 야후! 파이프에서 비표준 RSS 2.0 엘리먼트 사용하기

다양한 크기의 이미지를 가리키는 URL의 연산에 보다 직관적으로 사용할 수 있는 비표준 엘리먼트(예를 들어, <image\_prefix>)가 야후! 파이프가 생성하는 RSS 2.0 피드에 포함되었으면 하고 바랬었다. 하지만 아래에서 볼 수 있듯이 JSON 피드는 이러한 비표준 파라미터들을 포함하지만 RSS 2.0 피드에 이러한 정보가 포함되지 않는 것으로 밝혀졌다.

[http://pipes.yahoo.com/pipes/pipe.run?\\_id=YG9eZGWO3BGukZGJTqoASA&\\_render=json&api\\_key={api\\_key}&extras=geo&lat0=37.817785166068&lat1=37.926190569376&lon0=-122.34375&lon1=-122.17208862305&min\\_upload\\_date=820483200&per\\_page=10](http://pipes.yahoo.com/pipes/pipe.run?_id=YG9eZGWO3BGukZGJTqoASA&_render=json&api_key={api_key}&extras=geo&lat0=37.817785166068&lat1=37.926190569376&lon0=-122.34375&lon1=-122.17208862305&min_upload_date=820483200&per_page=10) ➡

이 문제를 해결하는 한 가지 방법은 PHP를 사용하여 비표준 엘리먼트들을 포함한 JSON을 RSS 2.0으로 변환하는 웹서비스를 작성하는 것이다. 하지만 이것은 작업을 좀 더 쉽게 만들려고 하는 원래의 목적을 해치게 된다.\*1

\*1 [http://discuss.pipes.yahoo.com/Message\\_Boards\\_for\\_Pipes/threadview?m=tm&bn=pip-DeveloperHelp&tid=2513&mid=2517&tofr=-1&rt=2&frt=2&off=1](http://discuss.pipes.yahoo.com/Message_Boards_for_Pipes/threadview?m=tm&bn=pip-DeveloperHelp&tid=2513&mid=2517&tofr=-1&rt=2&frt=2&off=1)

## <gm:map>를 사용하여 플리커 사진을 화면에 표시하기



[http://pipes.yahoo.com/pipes/pipe.run?\\_id=YG9eZGWO3BGukZGJTqoASA&\\_render=rss](http://pipes.yahoo.com/pipes/pipe.run?_id=YG9eZGWO3BGukZGJTqoASA&_render=rss) ➡  
&api\_key={api\_key}&extras=geo&lat0=37.817785166068&lat1=37.926190569376&lon0=  
-122.34375&lon1=-122.17208862305&min\_upload\_date=820483200&per\_page=100 ➡



<http://mashup-raymond-yee-flickrfeed2.googlemashups.com/>

■

■

■

- 

## 매쉬업에 자바스크립트 추가하기

- 

- 

### 사용자 정의 자바스크립트를 구글 매쉬업 에디터에서 사용하기

현재 작업 중인 메인 코드에 새로운 요소(이 경우에는 자바스크립트 이벤트 처리)를 추가하기 전에, 종종 이러한 아이디어를 시험하기 위해 추가적으로 작은 프로그램을 작성하곤 한다. 다음은 입력폼과 전송 이벤트를 포함하는 간단한 프로그램으로 10장에서 다룬 예제(<http://examples.mashupguide.net/ch10/square2.html>)와 유사하다.



<http://mashup-raymond-yee-flickrfeed3.googlecode.com/svn/trunk/index.gml>

<http://mashup-raymond-yee-flickrfeed3.googlemashups.com/>





■

■

어떻게 피드를 영속화하고 탭을 사용할 것인가

<http://mashup-raymond-yee-tabs0.googlecode.com/svn/trunk/index.gml>







<http://mashup-raymond-yee-tabs0a.google mashups.com/>

- 세 개의 태그가 두 개의 탭을 생성하기 위해 사용되었다. 탭을 초기화하기 위해 `<gm:tabs>` 태그가 사용되었으며 이 태그는 `<gm:container>`를 가리키는 `target` 어트리뷰트를 가지고 있다. 그리고 `<gm:container>` 엘리먼트는 각 탭을 나타내는 `<gm:section>`을 차례로 포함한다.<sup>3</sup>
- 썸네일의 Copy 버튼을 클릭하면 `copy_this()` 함수가 호출된다. 이 함수는 선택된 DOM 엘리먼트에 매칭되는 피드 항목을 식별하며 해당 엘리먼트를 ID가 `savedEntries`인 피드로 복사한다.
- `<gm:page>`의 `authenticate` 어트리뷰트가 `true`로 설정되어 있으므로 사용자가 이 애플리케이션을 사용하려면 구글 계정으로 로그인해야만 한다.

<http://code.google.com/support/bin/answer.py?answer=76140&topic=11689>



<http://mashup-raymond-yee-tabs0a.google mashups.com/feeds/public/user/raymond.yee%40gmail.com/crud> ➡

<http://mashup-raymond-yee-tabs0a.google mashups.com/feeds/user/crud>

---

<sup>3</sup> <http://code.google.com/gme/docs/samples.html#tabs>

**\* NOTE!** 구글 매쉬업 에디터 피드를 작성할 때, 피드에 포함되는 최대 항목의 개수가 1000개라는 것을 명심하자.

## 사진에 대한 선택과 삭제 프로세스의 변경

<http://mashup-raymond-yee-tabs1.googlecode.com/svn/trunk/index.gml>





<http://mashup-raymond-yee-tabs1.googlemashups.com/>



**저장된 항목을 지도상에 표시하는 최종 애플리케이션**

<http://mashup-raymond-yee-tabs1.googlemashups.com/>

<http://mashup-raymond-yee-flickrfeed3.googlemashups.com/>











- 구글 매쉬업 에디터가 제공하는 코멘트 추가 기능을 구현하여 사용자들이 하여금 피드 항목에 대해 태그를 추가하고 평점을 부여하도록 할 수 있다(<http://code.google.com/gme/docs/data.html#annotations>).

- 

- 

구글 매쉬업 에디터와 야후! 파이프의 사용에 따르는  
트레이드-오프 분석

- 

- 

- 

- 

-

■

■

■

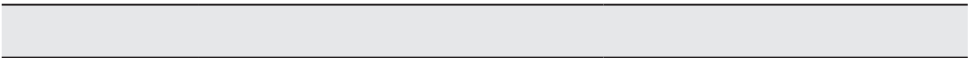
■

■

■

■

기타 매쉬업 툴





## 요약

# 12

---

사이트를 매쉬업 가능하도록  
만들기

---

- 공식적인 API를 생성하지 않고도 웹사이트와 콘텐츠가 매쉬업 가능하도록 만들 수 있는 방법
- 매쉬업이 용이하도록 해주는 기능들을 제공하여 API를 구성하는 방법

▶ **TIP!** 숙련된 API 제작자의 관점에서 API를 어떻게 생성, 운영, 그리고 유지/보수하는가에 대한 상세한 내용은 플리커의 칼 헨더슨Cal Henderson이 집필한 **Building Scalable Web Sites**(오라일리, 2006)의 11장을 참고하도록 하라.

## 왜 웹사이트를 매쉬업 가능하도록 만들어야 하는가



- 

- 

- 

- 

- 

- 

- 

API에 의존하지 않는 기술 사용하기

일관성 있고 풍부한 URL 언어를 사용하라

웹사이트 개발에 W3C 표준을 사용하라

웹 접근성에 주의를 기울여라

사용자들이 콘텐츠에 태그를 추가할 수 있도록 고려하라

피드를 제공하라.

여러분의 콘텐츠를 블로그와 다른 웹사이트에 쉽게 게시할 수 있도록 하라

**명확한 라이선스를 표시하여 콘텐츠의 공유를 장려하라**

<http://en.wikipedia.org/wiki/Wikipedia:Copyrights>

<http://www.freebase.com/signin/licensing>

사용자 콘텐츠에 대해 다양한 불러오기, 내보내기 옵션을 개발하라

사용자들이 여러분의 콘텐츠를 어떻게 리믹스하고 리믹스하기 쉽게 만드는지 연구하라

매쉬업이 용이한 API 만들기

다른 API로부터 배우고 모방하라.

<http://www.23hq.com/doc/api/>

<http://phpflickr.com/phpFlickr/README.txt>

**API의 사용자를 잊지 말라**

**배우기 쉬운 API를 개발하라**

- 플리커는 잘 문서화되어 있으며 <http://www.flickr.com/services/api/explore/?method=flickr.photos>에서 볼 수 있는 플리커 API 탐색기와 같이 학습하기에 편리한 구조를 채택하고 있다(플리커의 API 문서는 상당히 명쾌하게 정리되어 있으며 여러분은 쿼리를 실행하고 즉시 결과를 확인해 볼 수 있다).
- 
- 

## API의 사용성을 테스트하라



느슨하게 결합된 아키텍처를 구성하여 API의 생성이 다른 사용자에게 도움이 되는 만큼 여러분에게도 도움이 되도록 하라

REST 방식을 채택하고 가능하다면 SOAP와 XML-RPC도 지원하라

<http://blog.whatfettle.com/2007/01/11/good-web-apis-are-just-web-sites/>

REST의 특정 예로서 Atom 출판 프로토콜의 사용을 고려해보라

서드파티 개발자나 내부 개발자의 API 키트 개발을 장려하라

## API에서 광범위한 에러 리포팅을 지원하라

**\* NOTE!** SOAP의 가장 최근 버전(1.2) 명세서에는 어떻게 다양한 2XX, 3XX, 4XX HTTP 상태코드를 사용할 것인지에 대한 지침을 제공한다.<sup>1</sup>

---

<sup>1</sup> <http://www.w3.org/TR/2007/REC-soap12-part2-20070427/#http-reqbindwaitstate>

출력과 입력을 위해 여러 가지 입출력 형식을 지원하라

[http://blog.wired.com/monkeybites/2007/06/upload\\_to\\_flick.html](http://blog.wired.com/monkeybites/2007/06/upload_to_flick.html)

UI에서 제공되는 기능을 API에서도 지원하라

여러분의 사이트를 위한 검색기능을 포함시켜라

API의 버전을 관리하라

<http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=118>

**개발자 커뮤니티를 육성하라**

**API를 너무 통제하려 들지 마라**

**서비스 레벨 보증서의 생성을 고려하라**

<http://www.amazon.com/b?ie=UTF8&node=379654011>

**API 사용자가 여러분의 리소스를 현명하게 사용하도록 도와라**

**여러분의 애플리케이션을 오픈소스화하는 것에 대해 고민하라**

## 쉽게 이해할 수 있는 데이터 표준

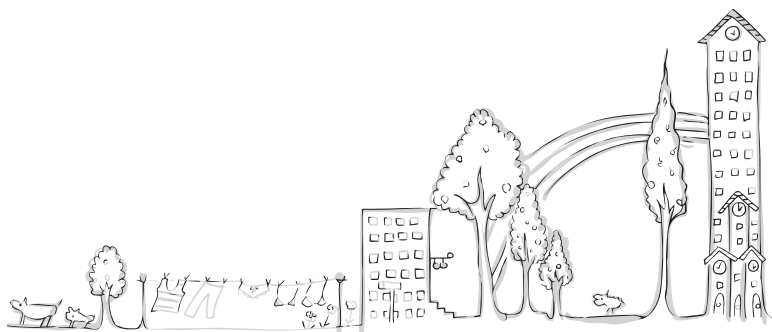
OASIS는 명료함, 품질 그리고 디지털 정보의 수명을 향상시키는 공개 표준의 사용을 문서화하고 장려하기 위해 커버페이지를 공개 리소스로 제공한다.

- RSS와 Atom 피드뿐만 아니라 KML을 다룰 때도 도움이 되는 <http://feedvalidator.org/> (4장을 참고하라)
- (X)HTML과 CSS의 유효성을 각각 체크할 수 있는 W3C 검증기(<http://validator.w3.org/> and <http://jigsaw.w3.org/css-validator/>)

## 요약







4부

---

# 다른 매쉬업 관련 주제에 대해 알아보자

---



# 13

---

온라인 지도와 3D 디지털 세계를  
리믹스하기

---

**\* NOTE!** 우리는 지도를 읽는 데 능숙하며 지도상의 사물들이 어떻게 연관되어 있는지 이해할 수 있다. 또, 지도상에 점을 추가하고 선을 그리는 데 익숙하므로 지도상에 다른 사물, 즉 점, 선, 그림 그리고 심지어 보다 추상적인 데이터를 추가하는 것이 그리 어려운 일은 아니다. 공간에 위치한 사물들은 자연히 지도상에 위치를 갖게 되며 이러한 점 때문에 지도는 강력한 상징성을 갖게 된다

- 구글 맵, 야후! 맵, 그리고 마이크로소프트 라이브 서치 맵과 같은 주요 지도 서비스 제공자의 API를 어떻게 사용하는지 설명할 것이다.
- 어떻게 프로그래밍을 하지 않고 웹 기반 지도를 생성하는지 설명할 것이다.
- KML과 GeoRSS, 그리고 CSV 등을 생성하여 어떻게 선언적인 방법으로 지도를 다루는지 설명할 것이다.
- KML에 대한 기초적인 내용과 구글 어스를 사용하여 기본적인 프로그래밍을 어떻게 하는지 알아볼 것이다.
- KML을 사용하여 플리커, 구글 어스, 그리고 구글 맵의 매쉬업을 어떻게 생성하는지 보여줄 것이다.

- 마이클 퍼비스Michael Purvis, 제프리 샘벨즈Jeffrey Sambells, 그리고 카메론 터너Cameron

Turner가 집필한 *Beginning Google Maps Applications with PHP and Ajax: From Novice to Professional*(Apress 출판사, 2006)

- 안드레 루이스Andre Lewis, 마이클 퍼비스Michael Purvis, 제프리 샘벨즈Jeffrey Sambells, 그리고 카메론 터너Cameron Turner가 집필한 *Beginning Google Maps Applications with Rails and Ajax: From Novice to Professional* (Apress 출판사, 2007)

## 온라인 지도 서비스의 범람

---

1 [http://en.wikipedia.org/wiki/Google\\_maps](http://en.wikipedia.org/wiki/Google_maps) and <http://www.adaptivepath.com/publications/essays/archives/000385.php>

2 [http://www.oreillynet.com/etel/blog/2005/05/hackers\\_tap\\_into\\_the\\_functiona.html](http://www.oreillynet.com/etel/blog/2005/05/hackers_tap_into_the_functiona.html)

## 지도 기반 매쉬업 예제들

- <http://tutorlinker.com/>은 지도형태의 인터페이스를 통해 강사와 학생을 연결해준다.
- <http://flashearth.com>은 플래시 인터페이스를 통해 다양한 온라인 지도를 매쉬업한다.

## 프로그래밍 없이 지도를 생성하기



## Mapbuilder.net

## Mapbuilder.net을 어떻게 사용하는가

- 
- 3 <http://www.mapbuilder.net/About.php>
  - 4 <http://www.imkblue.pwp.blueyonder.co.uk/index.html>
  - 5 <http://www.mapbuilder.net/Popular.php?OP=ALL>
  - 6 <http://www.pacwireless.com/distributor-locations.shtml>

1. 계정을 등록하라.<sup>7</sup>
2. New Map 링크를 클릭하라.<sup>8</sup> 지도의 이름은 문자, 숫자, 밑줄(\_), 그리고 대시(-)만을 포함할 수 있다는 점에 주의하라.
3. 다음 두 가지 방법 중의 하나로 점을 표시할 수 있다.
  - ‘Location Search & Quick Navigation’ 하단에 추가할 주소를 입력하라
  - 지점을 명시하기 위해 지도를 클릭하라.

---

7 <http://www.mapbuilder.net/SignUp.php>

8 <http://www.mapbuilder.net/Map.Add.php>

9 <http://www.mapbuilder.net/users/rdhyee/9329>

- “Some bookstores I like” (구글 맵 버전)<sup>10</sup>는 자바스크립트 인젝션<sup>11</sup>을 통해 최신상태로 유지된다.
- “Some bookstores I like” (야후 맵 버전)<sup>12</sup>

## 구글 My Maps

<http://local.google.com/support/bin/answer.py?hl=en&answer=68480>

<http://maps.google.com/maps/ms?f=q&hl=en&geocode=&ie=UTF8&msa=0 &msid=116029721704976049577.0000011345e68993fc0e7&z=14&om=1>

<http://examples.mashupguide.net/ch13/embedded.GMap.html>

---

<sup>10</sup> <http://examples.mashupguide.net/ch13/SomeBookstoresGMap.html>

<sup>11</sup> 자바스크립트 인젝션을 어떻게 사용하는지에 대한 설명은 <http://www.mapbuilder.net/Map.Implementation.php>를 참고하라

<sup>12</sup> <http://examples.mashupguide.net/ch13/SomeBookstoresYMap.html>

<sup>13</sup> 한 가지 문제를 발견하였는데 윈도우 XP의 파이어폭스 1.5.0.7을 사용해서는 어떻게 해도 특정한 지점을 삭제할 수 없었다. 결국 오페라를 사용하여 Mapbuilder.net에 로그인한후에야 해당 지점을 삭제할 수 있었다.

<sup>14</sup> <http://local.google.com/support/bin/answer.py?answer=72644>

<http://maps.google.com/maps/ms?f=q&msa=0&output=kml&msid={my-map-id}>

<http://maps.google.com/maps/ms?f=q&msa=0&output=kml&msid=116029721704976049577.000011345e68993fc0e7>

## 매쉬업 기회 : 야후! 로컬 컬렉션을 지도 서비스와 매쉬업하기

1. 야후! 로컬 API를 통해 정보를 가져온다.
2. Mapbuilder.net에 정보를 넘겨주기 위해 컬렉션 정보를 적절하게(XML에서 CSV로) 변환한다.

**getCollection을 사용하여 야후! 로컬에서 XML 데이터를 가져오기**

<http://collections.local.yahooapis.com/LocalSearchService/V1/getCollection>



---

[http://collections.local.yahooapis.com/LocalSearchService/V1/getCollection?appid={appid}&collection\\_id=1000014156](http://collections.local.yahooapis.com/LocalSearchService/V1/getCollection?appid={appid}&collection_id=1000014156)

▶ **TIP!** 야후! Appid는 <https://developer.yahoo.com/wsregapp/index.php>에서 발급받을 수 있다.



야후! 로컬의 XML 파일을 Mapbuilder.net에서 사용 가능한 CSV 포맷으로 변환하기



<http://examples.mashupguide.net/ch13/yahooCollectionToCSV.php?cid=1000014156>

<http://local.yahoo.com/userreviews?target=p0TJ1rUjf64lpQPwpZGZmXVTOyaM&rvwtype=COLLECTION>



**\* NOTE!** 이렇게 스크립트를 작성하는 것이 CSV만을 생성할 수 있게 해줄 뿐 여전히 Mapbuilder.net에서 수동으로 지도를 생성해야 한다고 생각할 수도 있다. 우리는 뒷부분에서 지도 생성 시에 사용하기 좀 더 사용하기 쉬운 포맷으로 이 컬렉션을 변환할 것이다.

## 마이크로소프트 라이브 서치 맵에서의 컬렉션 생성

`http://maps.live.com/?v=2&cid={collection-id}&encType=1`

`http://maps.live.com/?v=2&cid=74B8FFD299EDD840!106&encType=1`

`http://maps.live.com/GeoCommunity.aspx?action=retrieverss&mkt=en-us &cid={collection-id}`

`http://maps.live.com/GeoCommunity.aspx?action=retrieverss&mkt=en-us&cid=74B8FFD299EDD840!10`

<georss:point>37.865523 -122.258492</georss:point>

## 프로그래밍하지 않고 지도를 생성하는 방법

## 데이터 교환 포맷

- CSV
- 마이크로포맷과 기타 메타태그
- GeoRSS
- KML

### CSV

<http://rpc.geocoder.us/service/csv?address=2855+Telegraph+Ave.,+Berkeley,+CA>

37.858276,-122.260070,2855 Telegraph Ave,Berkeley,CA,94705

## HTML에서 사용되는 마이크로포맷과 메타태그

<http://www.flickr.com/photos/raymondye/18389540/>

**! CAUTION!** 비록 위도와 경도가 정확하다고 해도, 버클리가 오클랜드의 일부로 포함되지 않는다면 캘리포니아주 오클랜드 내의 주소의 배치는 정확하지 않다. 이러한 부정확함이 adr 예제의 문장상의 정확성을 해치지 않는다.<sup>19</sup>

## GeoRSS

- GeoRSS GML 인코딩(<http://georss.org/gml>)은 gml:pos 엘리먼트를 georss:where에 포함된 gml:Point 엘리먼트로 감싼다. 여기서 gml 접두어는 <http://www.opengis.net/gml>을 참조하고 georss 접두어는 <http://www.georss.org/georss>을 참조한다.

---

<sup>19</sup> (역자주) 이 예제에서 다루는 사진의 종탑은 UC버클리 캠퍼스 내에 위치한 것으로 UC 버클리는 실제로는 버클리 지역이 아닌 오클랜드에 위치하고 있다.

- GeoRSS Simple 인코딩(<http://georss.org/simple>)은 하나의 georss:point 엘리먼트를 사용하여 위도와 경도를 담는다. 여기서 georss는 GeoRSS GML 인코딩에서와 동일한 네임스페이스인 <http://www.georss.org/georss>를 참조한다.
- W3C Basic Geo 인코딩(<http://georss.org/w3c>)은 geo:Point를 사용하여 geo:lat와 geo:long 엘리먼트를 감싸며 여기서 geo 네임스페이스 접두어는 [http://www.w3.org/2003/01/geo/wgs84\\_pos#.\\_gml](http://www.w3.org/2003/01/geo/wgs84_pos#._gml)을 참조한다.
- Compact W3C Basic Geo 인코딩은 세 가지 인코딩 방식의 공통점을 이용하는 변형된 형태를 띠며 geo:Point로 감싸는 태그를 제거하였다.

[http://api.flickr.com/services/feeds/geo/?id={user-nsid}&lang=en-us&format=rss\\_200](http://api.flickr.com/services/feeds/geo/?id={user-nsid}&lang=en-us&format=rss_200)

[http://api.flickr.com/services/feeds/geo/?id=35468159852@N01&lang=en-us&format=rss\\_200](http://api.flickr.com/services/feeds/geo/?id=35468159852@N01&lang=en-us&format=rss_200)

[http://api.flickr.com/services/feeds/geo/?g={group-nsid}&lang=en-us&format=rss\\_200](http://api.flickr.com/services/feeds/geo/?g={group-nsid}&lang=en-us&format=rss_200)

[http://api.flickr.com/services/feeds/geo/?g=34427469792@N01&lang=en-us&format=rss\\_200](http://api.flickr.com/services/feeds/geo/?g=34427469792@N01&lang=en-us&format=rss_200)

## 야후!의 GeoRSS 사용과 YMaps 확장기능

- <http://developer.yahoo.com/maps/simple/V1/reference.html> (야후!맵의 Simple API를 위한 문서)
- <http://developer.yahoo.com/maps/georss/index.html> (Ajax API를 위한 문서)

[http://api.maps.yahoo.com/Maps/V1/annotatedMaps?appid=YahooDemo&xmlsrc=http://  
developer.yahoo.com/maps/sample.xml](http://api.maps.yahoo.com/Maps/V1/annotatedMaps?appid=YahooDemo&xmlsrc=http://developer.yahoo.com/maps/sample.xml)

[http://collections.local.yahooapis.com/LocalSearchService/V1/getCollection?appid={appid-  
id}&collection\\_id={collection-id}](http://collections.local.yahooapis.com/LocalSearchService/V1/getCollection?appid={appid}&collection_id={collection-id})

[http://collections.local.yahooapis.com/LocalSearchService/V1/getCollection?appid={appid-  
id}&collection\\_id=1000014156](http://collections.local.yahooapis.com/LocalSearchService/V1/getCollection?appid={appid}&collection_id=1000014156)

<http://examples.mashupguide.net/ch13/bookstores.yahoo.local.xml>

<http://examples.mashupguide.net/ch13/yahoo.local.to.georss.xml>





▶ **TIP!** 온라인에서 XSLT 변환을 하려면 <http://www.w3.org/2001/05/xslt> 와 [http://www.w3.org/2005/08/online\\_xslt/](http://www.w3.org/2005/08/online_xslt/) (XSLT2.0용) 에서 W3C XSLT 서비스를 이용하면 된다.

<http://api.maps.yahoo.com/Maps/V1/annotatedMaps?appid={app-id}&xmlsrc=http://examples.mashupguide.net/ch13/bookstores.georsss.xml>

1. UC 버클리 캠퍼스를 중심으로 지도를 위치시키고 야후!가 제공하는 예제<sup>22</sup>를 여러분의 야후! API 키<sup>23</sup>를 사용하여 수정하라.
2. 예제를 로딩하라. 이 예제를 로딩한 상태에서 자바스크립트 셸<sup>24</sup>을 호출한 후 다음의 코드를 입력하여 RSS 파일(YMaps 확장기능을 포함한)을 지도로 로딩하라.



▼ **CAUTION!** 만일 GeoRSS를 야후! 맵 API에 전달할 계획이라면 야후! 맵의 문서에 포함된 Compact W3C Basic Geo 인코딩을 사용하도록 하라. 야후!맵의 Simple API 의 에러메시지가 불충분하기 때문에 GeoRSS를 디버깅할 때는 Feedvalidator.org가 유용할 것이다.

## 버추얼 어스에서의 GeoRSS

<http://dev.live.com/virtualearth/sdk/Ref/HTML/WorkingWithLayers.htm>

<sup>21</sup> <http://developer.yahoo.com/maps/georss/index.html>

<sup>22</sup> <http://developer.yahoo.com/maps/ajax/V3/ajaxexample1.html>

<sup>23</sup> <http://examples.mashupguide.net/ch13/yahoo.map.berkeley.html>

<sup>24</sup> 8장에서 소개한 기술로 <http://www.squarefree.com/shell>을 참조하라.

- 
- 25 <http://msdn2.microsoft.com/en-us/library/bb429606.aspx>
- 26 <http://examples.mashupguide.net/ch13/flickr.geofeed.xml>

<http://examples.mashupguide.net/ch13/virtualearth.flickrgeofeed.v5.html>

## KML

---

27 <http://earth.google.com/kml>

28 <http://blog.pipes.yahoo.com/2007/05/02/pipes-adds-interactive-yahoo-maps-kml-support-and-more/>

29 <http://googleearthuser.blogspot.com/2007/05/feed-validator.html>

30 <http://geotips.blogspot.com/2007/04/kml-ogc.html>

## GeoRSS와 KML과 같은 포맷들 사이의 호환성

## API 프로그래밍으로 지도 생성하기

---

31 [http://www.oreillynnet.com/conferences/blog/2006/06/georss\\_and\\_kml.html](http://www.oreillynnet.com/conferences/blog/2006/06/georss_and_kml.html)

32 [http://www.ogleearth.com/2006/05/georss\\_is\\_here.html](http://www.ogleearth.com/2006/05/georss_is_here.html) 은 이것과 관련된 이슈를 잘 요약하였다.

33 <http://xtech06.usefulinc.com/schedule/paper/56>

34 <http://brainoff.com/gmaps/mgeorss.html>

- 
- 
- 
- 

## 구글 맵 API

**\* NOTE!** 구글 맵 API에 대한 이 튜토리얼은 본질적으로 8장에서 다룬 것과 같으며 여러분의 편의를 위해 반복하여 설명하였다.

## 구글 맵과 자바스크립트 쉘로 시작하기

구글 맵 API를 사용하여 간단한 지도를 생성해보자.

1. 지도를 호스팅할 공개된 웹 디렉터리를 확보하고 해당 디렉터리에 대한 URL을 알아야 한다. 무료 공개 API를 사용하는 구글 맵은 반드시 공개적으로 웹에서 접근할 수 있어야 한다.
2. 구글 맵에 접근하기 위한 키를 발급받기 위해 구글의 등록 페이지에 접속하라.<sup>36</sup> 여러분은 구글 맵을 호스팅하려는 해당 도메인을 위한 키가 필요하다(구글은 구글 맵 API의 사용을 통제하는 용도로 이 키 값을 사용한다).
3. 이용약관<sup>37</sup>을 읽고 동의한다면 테스트 파일이 위치하게 될 호스트의 URL을 입력하도록 하자. 예를 들어, 내 경우에는 `http://examples.mashupguide.net/ch13/`을 사용하였다. 나중에 사용할 수 있도록 키 값을 기억해 두자.
4. HTML 코드를 복사하여 호스팅하려는 디렉터리내의 페이지에 붙여넣기 하라. 아래와 유사한 코드를 생성하게 될 것이다.<sup>38</sup>

---

<sup>36</sup> <http://www.google.com/apis/maps/signup.html>

<sup>37</sup> <http://www.google.com/apis/maps/terms.html>

<sup>38</sup> <http://www.google.com/maps/api/signup?url=http%3A%2F%2Fexamples.mashupguide.net%2Fch13%2F>

5. 이제 map이라는 변수의 앞에 위치한 var 키워드를 제거하여 이 변수를 전역 변수로 설정하여 자바스크립트 셸에서 접근할 수 있도록 예제코드를 수정 하라. 즉,

을 다음과 같이 변경하여

자바스크립트 셸 유틸리티에서 map 객체에 접근할 수 있도록 하라.<sup>39</sup>

6. 지도를 로딩한 상태에서 자바스크립트 셸 북마크릿을 클릭하여 자바스크립트 셸을 호출하도록 하라. 아래 단계들의 코드를 타이핑하고 어떤 일이 발생하는지 관찰하라(다른 방법으로는 코드를 직접 수정하여 페이지를 재로딩할 수 있다). 이러한 과정은 버전 2의 구글 맵 API를 사용한다.<sup>40</sup>
7. 지도의 현재 확대수준(0부터 17의 값을 가지며 17이 가장 상세한 수준이다)으로 되돌아 가려면, 다음의 명령어를 입력하자(자바스크립트 셸의 응답은 코드 바로 다음에 표시하였다).

---

---

8. 지도의 중앙에 대한 위도와 경도를 얻으려면 다음과 같이 하라.

---

---

9. UC 버클리의 종탑을 중심으로 지도를 위치시키려면 다음과 같이 하라

<sup>39</sup> <http://examples.mashupguide.net/ch13/google.map.1.html>

<sup>40</sup> <http://www.google.com/apis/maps/documentation/reference.html#GMap2>



10. 다음과 같은 방법으로 위의 위치로 이동해 갈 수도 있다.

11. 작은 지도 컨트롤을 추가하려면, 아래 두 개의 명령어를 실행하라

12. GMap 키보드 내비게이션 기능을 활성화시키려면, 다음을 사용하라

---

---

13. 지도를 완전히 축소하려면 다음을 사용하라

14. 지도를 최대로 확대(최대 확대수준은 15에서 17정도가 될 것이다)하려면 다음을 사용하라

15. `maptypes` 변수를 세 개의 오브젝트를 가진 배열로 설정하려면 다음과 같이 하라.

---

---

16. `maptypes`에서 첫 번째 항목의 이름을 가져오려면 다음과 같이 하라

---

---

17. 현재 사용 중인 지도의 유형을 알아내려면, 오브젝트를 얻어서 해당 오브젝트의 유형에 대한 이름을 가져올 수 있다.

---

---

---

---

18. Maptype를 위성으로 설정하려면 다음과 같이하라.

19. 현재 최대나 최소 확대수준에 있지 않다면 한 단계 확대하고 축소할 수 있다.

20. 오버레이를 생성하려면 다음과 같이 하라.

---

---

---

---

21. 마커를 클릭했을 때 어떤 일을 수행하고 싶다면, 다음과 같이 하면 된다.

---

---

22. GeoRSS 피드로부터 읽은 데이터를 표시하는 핀에 연결된 새로운 레이어를 추가하려면, 다음과 같이 하라.

선으로 이루어진 도형과 오버레이, 그리고 드래그 가능한 점들과 같이 살펴볼 만한 것들이 아주 많이 있다. 보다 자세한 내용을 배우려면, 'Google Maps API : Introduction' 이란 문서<sup>42</sup>를 추천

---

42 <http://www.google.com/apis/maps/documentation/#Introduction>

한다. “이 문서는 자바스크립트 프로그래밍과 객체지향 프로그래밍 개념에 익숙한 사람들을 위해 설계되었다. 또한 사용자의 관점에서 구글 맵에 익숙해야 한다”는 것에 주의하자.

---

## 야후! 맵 API

**\* NOTE!** 만일 이 책에서 다루는 범위를 넘어서는 야후! 맵의 플래시 API에 대해 알고 싶다면, <http://developer.yahoo.net/maps/flash/index.html> 의 공식문서를 참고하라.

### 야후! 맵과 자바스크립트 쉘로 시작하기

1. 야후! 애플리케이션 키를 발급받도록 하라.<sup>46</sup> 야후! 문서상의 예제코드에 사용된 appid(YahooDemo이다)를 사용하면 안 된다.
2. 여러분의 웹사이트에 다음 코드를 복사하여 붙여넣기하라.<sup>47</sup>

---

<sup>43</sup> <http://developer.yahoo.net/maps/>

<sup>44</sup> <http://developer.yahoo.net/maps/simple/index.html>

<sup>45</sup> <http://developer.yahoo.net/maps/ajax/index.html>

<sup>46</sup> <https://developer.yahoo.com/wsregapp/index.php>

<sup>47</sup> 내 코드를 <http://examples.mashupguide.net/ch13/yahoo.map.simple.1.html>에 올려두었다.

4. 확대수준을 확인하려면 다음을 사용하라

---

---

5. 중심위치의 위도, 경도를 구하려면 다음을 사용하라

---

---

---

---

---

---

---

---

6. 확대수준(10이 가장 확대된 값이고 15가 가장 축소된 값이다)을 설정하려면, 다음을 사용하라.

7. 지도의 중심을 새로운 위치(이 경우에는 UC 버클리 캠퍼스이다)로 이동하려면, 다음과 같이 하라.

---

---

8. 내비게이션 컨트롤과 확대 컨트롤을 추가하려면, 다음을 사용하라

9. H라는 이름의 마커를 추가하려면, 다음을 사용하라

---

---

10. 클릭 이벤트가 팝업을 호출하도록 하려면, 다음과 같이 하라.



11. 캘리포니아 버클리의 텔레그래프가 2855 번지에 마커를 추가하려면, 다음의 세 가지 방법 중 하나를 사용하면 된다.

또는

또는

12. 지도에 야후! YMaps 확장기능이나 Compact W3C Basic Geo 인코딩을 사용하는 GeoRSS 피드를 추가하려면, 다음과 같이 하라.



위 예제에서는 폴리라인과 보다 완전한 오버레이 기능을 제공하는 야후! Ajax API를 과도하게 사용하지는 않았다.

---

## 마이크로소프트의 라이브 서치 맵/버추얼 어스

■

■

### 버추얼 어스와 자바스크립트 웹로 시작하기

1. 다음 코드(이 코드는 버추얼 어스 인터랙티브 SDK가 제공하는 가장 간단한 코드이다)를 복사하여 붙여넣기 하라.<sup>52</sup>

50 <http://dev.live.com/virtualearth>

51 <http://dev.live.com/virtualearth/sdk/>

52 <http://examples.mashupguide.net/ch13/ve.map.1.html>

---

---

4. 확대, 축소를 하려면 다음을 사용하라.

5. 지도의 유형을 확인하려면, 다음과 같이 하라.

---

---

6. 지도의 유형(a=항공사진, r=도로사진, h=하이브리드, 그리고 o=내려다보는 사진)을 설정 하려면, 다음을 사용하라.

### 버추얼 어스의 3D 기능

<http://msdn2.microsoft.com/en-us/library/bb429547.aspx>

다음 명령어를 자바스크립트 웹에서 실행하여 지도를 3D 모드로 설정할 수 있다.

```
map.SetMapMode(VEMapMode.Mode3D);
```

버추얼 어스의 셰이프(shape) 다루기와 운전방향 설정과 같은 다른 기능들에 대해 배우려면 버추얼 어스 인터랙티브 SDK(<http://dev.live.com/virtualearth/sdk/>)를 사용하도록 하라.

---



## 주소와 좌표 간의 변환

**▼ CAUTION!** 모호한 주소를 처리할 때 발생하는 API의 정확성 문제와 주어진 위치에 어떤 주소-좌표변환기가 최적인지에 관한 문제에는 미묘한 차이점들이 존재하며 여기서는 다루지 않는다.

### 야후! 맵

<http://developer.yahoo.com/maps/rest/V1/geocode.html>

- street, city, state, zip을 조합하여 사용한다.

`http://api.local.yahoo.com/MapsService/V1/geocode?appid={app-id}&street=2855+Telegraph+Ave.&city=Berkeley&state=CA`

- street, city, state, zip의 조합으로 구성된 하나의 문자열을 사용하며 아래의 예에서 보듯이 location 문자열이 URL에서 가장 앞부분에 위치한다.

`http://api.local.yahoo.com/MapsService/V1/geocode?appid={app-id}&location=2855+Telegraph+Ave.%2C+Berkeley%2C+CA`

■

■

■

■

■

■

■

# Geocoder.us

---

53 <http://api.local.yahoo.com/MapsService/V1/geocode?appid={app-id}&city=Berkeley>

54 <http://api.local.yahoo.com/MapsService/V1/geocode?appid={app-id}&location=350+5th+Ave,+New+York,+NY&output=php>

55 <http://api.local.yahoo.com/MapsService/V1/geocode?appid={app-id}&location=350000+main+Street,+Berkeley,+CA>

<http://geocoder.us/help/>

- Geocoder.us의 사용자 인터페이스에서 다음을 호출하면

<http://geocoder.us/demo.cgi?address=2855+Telegraph+Ave.%2C+Berkeley%2C+CA>

(37.858276, -122.260070)

- CSV 인터페이스를 사용하여 다음의 URL을 호출하면

<http://rpc.geocoder.us/service/csv?address=2855+Telegraph+Ave.%2C+Berkeley%2C+CA>

37.858276,-122.260070,2855 Telegraph Ave,Berkeley,CA,94705

- 다음의 REST 인터페이스를 사용하면

<http://geocoder.us/service/rest/?address=2855+Telegraph+Ave.%2C+Berkeley%2C+CA>

## 구글 Geocoder

[http://www.google.com/apis/maps/documentation/#Geocoding\\_Examples](http://www.google.com/apis/maps/documentation/#Geocoding_Examples)

## REST 인터페이스

- q는 주소-좌표 변환될 주소이다.
- key는 여러분의 API 키 값이다.<sup>56</sup>
- output은 xml, kml, csv, 또는 json중 하나의 값을 갖는 결과 값의 포맷이다.

---

<sup>56</sup> <http://www.google.com/apis/maps/signup.html> 에서 여러분의 키를 얻을 수 있으며 ABQIAAAAdjIS7YH6Pzk2Nrli02b5xxR10RG5t-vK3TwPKbpNUO2c5sYb4RTmySs\_TEFzYvIZrCaYJKlmTzJ5IA 이 <http://examples.mashupguide.net/ch13/>을 위한 키 값이다.

<sup>57</sup> [http://maps.google.com/maps/geo?q=2855+Telegraph+Ave.%2C+Berkeley%2C+CA.&output=xml&key=ABQIAAAAdjIS7YH6Pzk2Nrli02b5xxR10RG5t-vK3TwPKbpNUO2c5sYb4RTmySs\\_TEFzYvIZrCaYJKlmTzJ5IA](http://maps.google.com/maps/geo?q=2855+Telegraph+Ave.%2C+Berkeley%2C+CA.&output=xml&key=ABQIAAAAdjIS7YH6Pzk2Nrli02b5xxR10RG5t-vK3TwPKbpNUO2c5sYb4RTmySs_TEFzYvIZrCaYJKlmTzJ5IA)

**⚠ CAUTION!** XML 결과 값이 KML 엘리먼트인 <Placemark>를 포함하고 있지만 결과 값에 <Response> 엘리먼트가 포함되어 있기 때문에 이 문서는 유효한 KML이 아니다

## 자바스크립트 인터페이스

1. 이전 절에서 다룬 간단한 구글 맵을 로딩하라.<sup>60</sup>
2. 자바스크립트 셸에서, 주소를 할당하고 GClientGeocoder의 인스턴스를 생성하라.

```
address = "2855 Telegraph Ave., Berkeley, CA"
```

---

---

```
geocoder = new GClientGeocoder();
```

---

---

3. 주소와 콜백함수를 파라미터로 받는 GClientGeocoder.getLatLng에 대한 호출 코드를 작성하라. 이 콜백함수는 주소에 대한 좌표변환 값을 파라미터로 받는다.

아래와 같이 함수에 포함된 공백을 제거하면, 자바스크립트 셸에서 쉽게 호출할 수 있을 것이다.

---

<sup>59</sup> [http://www.google.com/apis/maps/documentation/#Geocoding\\_JavaScript](http://www.google.com/apis/maps/documentation/#Geocoding_JavaScript)

<sup>60</sup> <http://examples.mashupguide.net/ch13/google.map.1.html>

4. 이제 그림 13-1에서 보는 것과 같이 구글 맵에 2855 Telegraph Ave., Berkeley, CA를 표시하는 오버레이가 추가된 것을 확인할 수 있을 것이다.



## 버추얼 어스

1. 기본 버추얼 어스 예제를 로딩하라.<sup>62</sup>

61 <http://msdn2.microsoft.com/en-US/library/bb429645.aspx>

62 <http://examples.mashupguide.net/ch13/ve.map.1.html>

2. 주소-좌표변환 기능을 통해 찾아진 위도/경도를 팝업창으로 보여주는 콜백함수를 생성하도록 하자.

3. 자바스크립트 셸에서 다음을 입력하라.

---

---

아래와 같이 공백을 제거하면 자바스크립트 셸에서 호출하기 편리할 것이다.

**\* NOTE!** 마지막 라인의 많은 null 파라미터 때문에 놀랐을 수도 있을 것이다. 하지만 이것이 <http://msdn2.microsoft.com/en-us/library/bb545008.aspx>에서 공식적으로 추천하는 호출방법이다.

4. 주소의 위도와 경도를 표시하는 팝업창을 볼 수 있을 것이다.



## 미국 내 주소가 아닌 경우의 주소-좌표변환

- 구글의 주소-좌표 변환기능은 아래에 목록으로 정리된 국가들의 주소를 변환할 수 있다.

<http://code.google.com/support/bin/answer.py?answer=62668&topic=12266>

위 목록은 현재 오스트리아, 오스트레일리아, 벨기에, 브라질, 캐나다, 체코 공화국, 덴마크, 핀란드, 프랑스, 독일, 홍콩, 헝가리, 인도, 아일랜드, 이탈리아, 일본, 룩셈부르크,

네덜란드, 뉴질랜드, 폴란드, 포르투갈, 싱가포르, 스페인, 스웨덴, 스위스, 대만, 영국, 미국, 그리고 한국을 포함하고 있다. 결과의 정확도가 국가별로 다를 수 있다는 것에 주의하도록 하라.

- 야후!가 변환할 수 있는 국가의 목록을 가장 잘 정리한 페이지는 다음과 같다.

<http://www.ysearchblog.com/archives/000563.html>

이 블로그 항목은 오스트리아, 벨기에, 덴마크, 핀란드, 독일, 영국, 룩셈부르크, 네덜란드, 아일랜드, 노르웨이, 포르투갈, 스페인, 스웨덴, 스위스, 프랑스, 그리고 이탈리아 등의 서유럽의 국가들에 대해 전체를 변환할 수 있다고 명시하고 있다.

- 기타 주소-좌표변환 기능에 대해서는 다음의 목록을 참고하라.

<http://groups.google.com/group/Google-Maps-API/web/resources-non-google-geocoders>

<http://mapki.com/wiki/FAQs#Geocoding>

## 구글 어스와 KML

■

■

■

---

64 [http://en.wikipedia.org/wiki/Virtual\\_globe](http://en.wikipedia.org/wiki/Virtual_globe)

65 [http://www.technologyreview.com/read\\_article.aspx?ch=specialsections&sc=personal&id=17537](http://www.technologyreview.com/read_article.aspx?ch=specialsections&sc=personal&id=17537) 는 구글 어스가 시장 지배적인 플랫폼이 될 것이라고 주장한다.

## 최종 사용자로서 KML을 화면에 표시하고 다루기

- 구글 어스 갤러리(<http://earth.google.com/gallery/index.html>)
- 구글 어스 커뮤니티(<http://bbs.keyhole.com/ubb/ubbthreads.php/Cat/0>)

<http://maps.google.com/maps/ms?f=q&hl=en&geocode=&ie=UTF8&msa=0&msid=116029721704976049577.0000011345e68993fc0e7&z=14&om=1>

<http://maps.google.com/maps/ms?f=q&msa=0&output=kml&msid=116029721704976049577.0000011345e68993fc0e7>

### KML 파일을 구글 어스로 다운로드하기

1. 구글 어스를 설치하도록 하라. 아래 URL에서 다운로드할 수 있다.

<http://earth.google.com/>

2. 구글 어스가 설치되면, 인터페이스를 어떻게 내비게이션하는지 사용해보도록 하라. 적어도, 주소나 사업체 이름을 입력하여 구글 인터페이스가 해당 장소를 표시하도록 하는 것에는 익숙해져야만 한다. 또한 각 항목에 대한 컬렉션을 생성하고 각 항목의 위도, 경도, 아이콘, 그리고 뷰와 같은 프로퍼티들을 변경하기 위해 Save to My Places 기능

구글 My Maps에 생성해둔 서점들을 표시하는 지도에 접속하자.

<http://maps.google.com/maps/ms?f=q&msa=0&msid=116029721704976049577.0000011345e68993fc0e7&z=14&om=1>

4. 다음과 같은 KML 링크를 클릭하라.

<http://maps.google.com/maps/ms?f=q&om=1&ie=UTF8&msa=0&output=nl&msid=116029721704976049577.0000011345e68993fc0e7>

5. 만약 브라우저와 구글 어스가 일반적인 환경설정(구글 어스가 Content-Type 헤더의 값이 "application/vnd.google-earth.kml+xml"인 파일을 처리하도록 등록되었다면)으로 설치되었다면, 구글 어스를 사용하여 다운로드된 KML을 열 것인가를 묻는 창을 보게 될 것이다. 만약 수락한다면, 서점의 위치를 나타내는 지점들의 컬렉션이 임시 공간에 로딩된다. 컬렉션의 링크를 더블 클릭하면 구글 어스의 화면이 회전하면서 마커들이 나타나게 된다. 구글 어스에 표시된 구글 맵 컬렉션을 보려면 그림 13-2을 살펴보라.

---

66 프로그램을 구동하기 위해, 저장된 KML파일의 기본위치를 알아야만 한다. 윈도우계열에서는 C:\Documents and Settings\W[USERNAME]\Application Data\Google\GoogleEarth\myplaces.kml 에 저장되며 맥 OS X에서는 ~/Library/Google Earth/myplaces.kml 에 저장된다.



<http://maps.google.com/maps?q=http:%2F%2Fmaps.google.com%2Fmaps%2Fms%3Ff%3Dq%26om%3D1%26ie%3DUTF8%26msa%3D0%26output%3Dnl%26msid%3D116029721704976049577.0000011345e68993fc0e7>



## 플리커에서 KML 생성하기

<http://geobloggers.com/archives/2007/05/31/flickr-kml-and-a-stroll-down-memory-lane/>

- 가장 최근의 사진을 보여주기 위해 주기적으로 갱신을 하는 KML 네트워크 링크를 사용하도록 format=kml\_ni를 선택할 수 있다(잠시 후에 KML의 <NetworkLink> 엘리먼트에 대해 논의할 것이다).
- 위치 데이터를 포함하는 정적인 KML을 생성하기 위해 format=kml을 선택할 수 있다.

[http://api.flickr.com/services/feeds/geo/?format=kml\\_nl](http://api.flickr.com/services/feeds/geo/?format=kml_nl)

[http://maps.google.com/maps?q=http:%2F%2Fapi.flickr.com%2Fservices%2Ffeeds%2Fgeo%2F%3Fformat%3Dkml\\_nl](http://maps.google.com/maps?q=http:%2F%2Fapi.flickr.com%2Fservices%2Ffeeds%2Fgeo%2F%3Fformat%3Dkml_nl)

[http://api.flickr.com/services/feeds/geo/?id={user-nsid}&format=kml\\_nl](http://api.flickr.com/services/feeds/geo/?id={user-nsid}&format=kml_nl)

[http://api.flickr.com/services/feeds/geo/?id=35468159852@N01&format=kml\\_nl](http://api.flickr.com/services/feeds/geo/?id=35468159852@N01&format=kml_nl)

[http://api.flickr.com/services/feeds/geo/?g={group-nsid}&format=kml\\_nl](http://api.flickr.com/services/feeds/geo/?g={group-nsid}&format=kml_nl)

[http://api.flickr.com/services/feeds/geo/?g=94823070@N00&format=kml\\_nl](http://api.flickr.com/services/feeds/geo/?g=94823070@N00&format=kml_nl)

[http://api.flickr.com/services/feeds/geo/&tags=flower&format=kml\\_nl](http://api.flickr.com/services/feeds/geo/&tags=flower&format=kml_nl)

[http://api.flickr.com/services/feeds/geo/us/?format=kml\\_nl](http://api.flickr.com/services/feeds/geo/us/?format=kml_nl)

[http://api.flickr.com/services/feeds/geo/us/ca/berkeley/?format=kml\\_nl](http://api.flickr.com/services/feeds/geo/us/ca/berkeley/?format=kml_nl)

[http://api.flickr.com/services/feeds/geo/uk/london/?format=kml\\_nl](http://api.flickr.com/services/feeds/geo/uk/london/?format=kml_nl)

[http://api.flickr.com/services/feeds/geo/ca/on/toronto/?format=kml\\_nl](http://api.flickr.com/services/feeds/geo/ca/on/toronto/?format=kml_nl)

[http://api.flickr.com/services/feeds/geo/FR/%C3%8Ele-de-France/Paris?format=kml\\_nl](http://api.flickr.com/services/feeds/geo/FR/%C3%8Ele-de-France/Paris?format=kml_nl)

[http://api.flickr.com/services/feeds/geo/cn/beijing/?format=kml\\_nl](http://api.flickr.com/services/feeds/geo/cn/beijing/?format=kml_nl)

[http://api.flickr.com/services/feeds/geo/cn/%E5%8C%97%E4%BA%AC/?format=kml\\_nl](http://api.flickr.com/services/feeds/geo/cn/%E5%8C%97%E4%BA%AC/?format=kml_nl)

[http://api.flickr.com/services/feeds/geo/us/ca/berkeley/?id=48600101146@N01&tags=flower&format=kml\\_  
nl](http://api.flickr.com/services/feeds/geo/us/ca/berkeley/?id=48600101146@N01&tags=flower&format=kml_nl)

**▼ CAUTION!** KML이 초기 단계를 넘어 어떻게 발전해갈지가 궁금하다면 플리커에서 KML에 대한 공식적인 문서를 찾아보라.

## KML

<http://code.google.com/apis/kml/documentation/>

[http://code.google.com/apis/kml/documentation/KML\\_Samples.kml](http://code.google.com/apis/kml/documentation/KML_Samples.kml)

[http://maps.google.com/maps?q=http%3A%2F%2Fkmlscribe.googlepages.com%2FSamplesInMaps.  
kml](http://maps.google.com/maps?q=http%3A%2F%2Fkmlscribe.googlepages.com%2FSamplesInMaps.kml)



<http://maps.google.com/maps?q=http:%2F%2Fexamples.mashupguide.net%2Fch13%2Fberkeley.simple.kml>

<http://googlemapsapi.blogspot.com/2007/06/validate-your-kml-online-or-offline.html>

<http://feedvalidator.org/check.cgi?url=http%3A%2F%2Fexamples.mashupguide.net%2Fch13%2Fberkeley.simple.kml>

<http://code.google.com/apis/kml/schema/kml21.xsd>

## Placemark에 LookAt과 Camera 뷰 추가하기

<http://code.google.com/apis/kml/documentation/cameras.html>

<http://examples.mashupguide.net/ch13/berkeley.campanile.evans.kml>

<http://maps.google.com/maps?f=q&hl=en&geocode=&q=http:%2F%2Fexamples.mashupguide.net%2Fch13%2Fberkeley.campanile.evans.kml&ie=UTF8&ll=37.872507,-122.257565&spn=0.006581,0.01133&z=17&om=1>

<http://tinyurl.com/38oawy>

Web Images Video News Maps Gmail more

raymond.yee@gmail.com | Saved Locations | Help | Web History | My Account | Sign out

Google Maps

Search the map | Find businesses

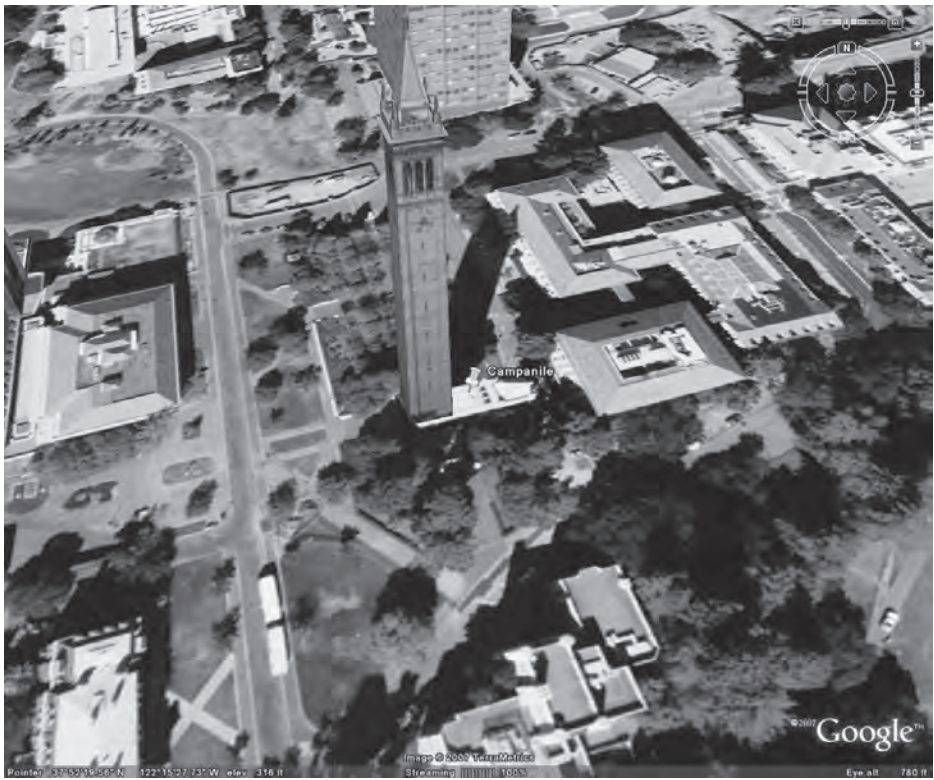
Search Results | My Maps

Save to My Maps

**Contents**

- Campanile and Evans Hall, UC Berkeley  
This file illustrates the use of and in KML, using two buildings on the UC Berkeley campus. The Campanile is almost due south of Evans Hall. Note: turn on 3D Warehouse layer to see the buildings in 3D in Google Earth.
- Campanile  
Campanile viewed from the south, using
- On the roof of Evans Hall Looking at Campanile  
Looking south from Evans to Campanile.
- On the roof of Evans Hall Looking at Campanile (with a 3D)  
Looking south from Evans to Campanile.

©2007 Google - Map data ©2007 NAVTEQ



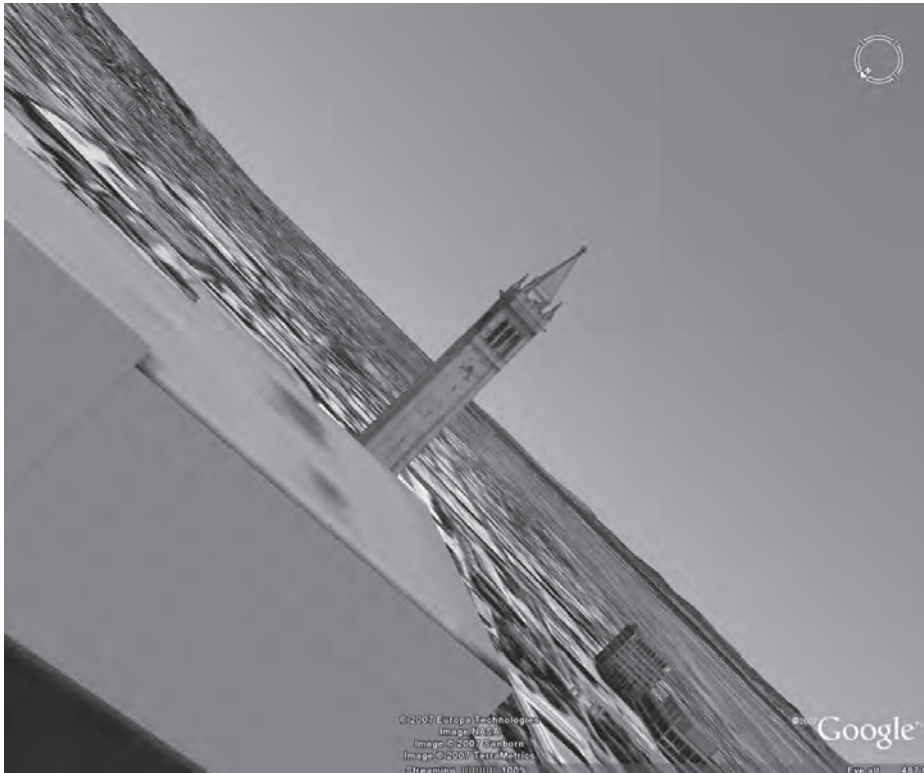
■

■



- roll은 기본값인 0도로 설정된다. 아래 URL을 로딩하면 세 번째 placemark를 볼 수 있다.

<http://examples.mashupguide.net/ch13/berkeley.campanile.evans.kml>



COM과 애플스크립트를 사용하여 구글 어스 프로그래밍하기

## COM 인터페이스

<http://earth.google.com/comapi/index.html>

구글 어스에 대한 애플스크립트 인터페이스

<http://appscript.sourceforge.net/>



↳

↳

맵스트랙션과 오픈레이어즈

## 통합 예제: 플리커 사진을 구글 어스에서 보여주기

<http://examples.mashupguide.net/ch13/flickrgeo.php>

<http://examples.mashupguide.net/ch13/flickrgeo.php.txt>

■

- flickrgeo는 결과 값의 포맷을 설정하기 위해 `o_format`이라는 파라미터를 사용한다. 허용되는 값은 `html`(간단한 사용자 인터페이스를 위해), `rest`(REST 포맷으로 플리커 API의 결과 값을 반환하기 위해), `json`(결과 값을 JSON형태로 반환하기 위해), `nl`(KML 네트워크를 반환하기 위해), `kml`(사진 데이터를 KML로 반환하기 위해)이며 `format`에 대한 적절한 값이 `flickr.photos.search`로 전달된다.

[http://examples.mashupguide.net/ch13/flickrgeo.php?user\\_id=48600101146%40N01&tags=&text=&lat0=37.817785166068&lon0=-122.34375&lat1=37.926190569376&lon1=-122.17208862305&page=1&per\\_page=10&min\\_upload\\_date=820483200&extras=geo&o\\_format=html](http://examples.mashupguide.net/ch13/flickrgeo.php?user_id=48600101146%40N01&tags=&text=&lat0=37.817785166068&lon0=-122.34375&lat1=37.926190569376&lon1=-122.17208862305&page=1&per_page=10&min_upload_date=820483200&extras=geo&o_format=html)

<http://tinyurl.com/2nbjbb>

[http://examples.mashupguide.net/ch13/flickrgeo.php?user\\_id=48600101146%40N01&tags=&text=&lat0=37.817785166068&lon0=-122.34375&lat1=37.926190569376&lon1=-122.17208862305&page=1&per\\_page=10&min\\_upload\\_date=820483200&extras=geo&o\\_format=kml](http://examples.mashupguide.net/ch13/flickrgeo.php?user_id=48600101146%40N01&tags=&text=&lat0=37.817785166068&lon0=-122.34375&lat1=37.926190569376&lon1=-122.17208862305&page=1&per_page=10&min_upload_date=820483200&extras=geo&o_format=kml)

<http://tinyurl.com/36hu2j>

[http://maps.google.com/maps?f=q&hl=en&geocode=&q=http%3F%2Fexamples.mashupguide.net%2Fch13%2Fflickrgeo.php%3Fuser\\_id%3D48600101146%2540N01%26tags%3D%26text%3D%26lat0%3D37.817785166068%26lon0%3D-122.34375%26lat1%3D37.926190569376%26lon1%3D-122.17208862305%26page%3D1%26per\\_page%3D10%26min\\_upload\\_date%3D820483200%26extras%3Dgeo%26o\\_format%3Dkml&ie=UTF8&z=14&om=1](http://maps.google.com/maps?f=q&hl=en&geocode=&q=http%3F%2Fexamples.mashupguide.net%2Fch13%2Fflickrgeo.php%3Fuser_id%3D48600101146%2540N01%26tags%3D%26text%3D%26lat0%3D37.817785166068%26lon0%3D-122.34375%26lat1%3D37.926190569376%26lon1%3D-122.17208862305%26page%3D1%26per_page%3D10%26min_upload_date%3D820483200%26extras%3Dgeo%26o_format%3Dkml&ie=UTF8&z=14&om=1)



1. 위치정보가 태그된 사진을 플리커에 질의한다.
2. flickr.photos.search로부터 받은 플리커 결과 값을 이에 상응하는 KML로 변환한다.
3. 구글 어스의 동작을 flickrgeo로 전달하기 위해 KML 네트워크링크를 생성한다.
4. o\_format에 다양한 값을 설정할 수 있도록 하나의 스크립트 내에 잘 정리한다.

## KML 네트워크 링크

<http://examples.mashupguide.net/ch13/hello.world.networklink.kml>

<http://examples.mashupguide.net/ch13/hello.world.networklink.kml>





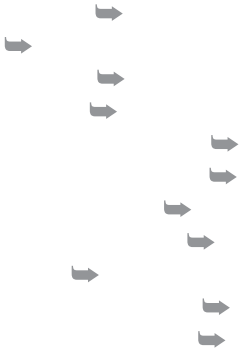
<http://examples.mashupguide.net/ch13/hello.world.networklink.php>







# 사진을 표시하는 KML 생성하기



■

■

■

flickrgeo.php 코드





















요약

■

■

■

■

■

■

■

■

■

■

■



# 14

---

## 소셜 북마킹과 도서목록 시스템에 대한 탐구

---

■

■

---

1 <http://www.educause.edu/ir/library/pdf/ELI7001.pdf>

2 <http://kftf.ischool.washington.edu/>

■

■

■

■

■

---

3 [http://en.wikipedia.org/wiki/List\\_of\\_social\\_software#Social\\_bookmarking](http://en.wikipedia.org/wiki/List_of_social_software#Social_bookmarking)  
4 <http://3spots.blogspot.com/2006/01/all-social-that-can-bookmark.html>



## 소셜 북마킹 사이트의 시장현황

- 

- 

Programmableweb.com을 사용하여 API의 인기를 알아보기

<http://www.programmableweb.com/apilist/bycat>

- 딜리셔스(del.icio.us)
- 심피(simpify)
- 블로그마크스(Blogmarks)
- 스크라이블(Scribble)
- 쉐도우즈(Shadows)
- 젯스(Jots)
- 로브(Rove)
- 온리와이어(OnlyWire)

- 링크가고고(linkaGoGo)
- 매그놀리아(Ma.gnolia)



---

\*위 표는 2007년 8월 11에 생성되었다.

## 딜리셔스

<http://del.icio.us/rdhyee>

`http://del.icio.us/rdhyee/NYTimes`

`http://del.icio.us/tag/NYTimes`

`http://del.icio.us/popular/NYTimes`

`http://del.icio.us/popular/`

`http://del.icio.us/popular/?new`

`http://harpers.org/TheEcstasyOfInfluence.html`

`http://del.icio.us/url/53113b15b14c90292a02c24b55c316e5`

`md5.new("http://harpers.org/TheEcstasyOfInfluence.html").hexdigest()`

`http://del.icio.us/url?url=http://harpers.org/TheEcstasyOfInfluence.html`

`http://del.icio.us/url/53113b15b14c90292a02c24b55c316e5`

**딜리셔스 API 사용하기**

- 모든 딜리셔스 API 호출은 반드시 HTTPS를 통해 전달되어야 한다.
- 모든 API 호출은 HTTP 인증을 요구하며, 이것은 API 키 값이 본질적으로 존재하지 않는다는 의미이다. 하지만 모든 API 호출은 특정 사용자 계정에 연결되어 동작한다.
- 503 HTTP 에러(서버의 부하를 줄이기 위해 호출이 조절되고 있으며 호출의 빈도수를 늦춰야 한다는 것을 의미한다)에 대해서 처리를 해야만 한다.

✱

## 업데이트

<http://del.icio.us/help/api/update>

- 웹 브라우저를 사용하여, 다음 URL에 접속하고 입력창이 화면에 표시되면 딜리셔스 사용자 아이디와 비밀번호를 입력하라.

`https://api.del.icio.us/v1/posts/update`

■

`curl -u USER:PASSWORD https://api.del.icio.us/v1/posts/update`

- 마지막 방법으로 URL에 사용자 아이디와 비밀번호를 포함시켜서 사용할 수 있다.

`curl https://{user}:{password}@api.del.icio.us/v1/posts/update`

---

---

**\* NOTE!** 다음 예제들에서, 우리는 두 번째 방법만을 사용할 것이다.

## 태그

`curl -u USER:PASSWORD https://api.del.icio.us/v1/tags/get`

`curl -u USER:PASSWORD https://api.del.icio.us/v1/tags/rename?old=FEDORA&new=fedora`

## 포스트

## get 메서드

`https://api.del.icio.us/v1/posts/get?`

`curl -u USER:PASSWORD https://api.del.icio.us/v1/posts/get?tag=mashup`

`http://www.ala.org/ala/acrl/acrlissues/future/changingroles.htm`

`curl -u USER:PASSWORD https://api.del.icio.us/v1/posts/get?➡`  
`url=http://www.ala.org/ala/acrl/acrlissues/future/changingroles.htm`

---

---

---

---

`http://del.icio.us/rss/url?url={url}`

`http://del.icio.us/rss/url?url=http://www.ala.org/ala/acrl/acrlissues/future/changingroles.  
htm`

## recent 메서드

`https://api.del.icio.us/v1/posts/recent?`

`curl -u USER:PASSWORD https://api.del.icio.us/v1/posts/recent?count=5`

## all 메서드

`curl -u USER:PASSWORD https://api.del.icio.us/v1/posts/all?tag=architecture`

## add 메서드

- &url(필수)은 아이템의 URL이다.
- &description(필수)은 아이템에 대한 설명이다.
- &extended(선택)는 아이템에 대한 노트이다.
- &tags(선택)는 아이템에 대한 태그이다(공백으로 각각 구분된다).
- &dt(선택)는 CCYY-MM-DDThh:mm:ssZ의 포맷을 가진 아이템에 대한 날짜 정보이다. 이 값은 <http://www.cl.cam.ac.uk/~mgk25/iso-time.html>의 ISO8601에 명시된 것과 같이 문자 T와 Z를 필요로 한다. 예를 들면 1984-09-01T14:21:31Z의 형태이다.
- &replace=no(선택)는 만약 주어진 URL이 이미 포스팅되었다면 post를 대체하지 않는다.
- &shared=no(선택)는 아이템을 비공개로 만든다.

```
curl -u USER:PASSWORD "https://api.del.icio.us/v1/posts/add?url=http://www.ala.org/ala/acrl/acrlissues/future/changingroles.htm&description=ALA+%7C+Changing+Roles+of+Academic+and+Research+Libraries&tags=library+2.0+academic+ACRL+technology"
```

---



---

## dates 메서드

```
curl -u USER:PASSWORD https://api.del.icio.us/v1/posts/dates?tag=mashup
```



## delete 메서드

```
curl -u USER:PASSWORD https://api.del.icio.us/v1/posts/delete?url={url}
```

## 번들

```
curl -u USER:PASSWORD "https://api.del.icio.us/v1/tags/bundles/set?bundle=Google&tags=GoogleMaps+GoogleEarth"
```

```
curl -u USER:PASSWORD https://api.del.icio.us/v1/tags/bundles/delete?bundle=Google
```

## RSS와 JSON

<http://del.icio.us/help/rss>

<http://del.icio.us/rss/>

<http://del.icio.us/rss/recent>

<http://del.icio.us/rss/popular>

<http://del.icio.us/rss/{user}>

<http://del.icio.us/rss/tag/{tag}>

<http://del.icio.us/rss/tag/mashup>

<http://del.icio.us/rss/tag/mashup+computer>

<http://del.icio.us/rss/{user}/{tag}>

<http://del.icio.us/rss/rdhyee/mashp+computer>

<http://del.icio.us/rss/url?url={url}>

[http://del.icio.us/rss/url?url=http://www.ala.org/ala/acrl/acrlissues/future/changingroles.  
htm](http://del.icio.us/rss/url?url=http://www.ala.org/ala/acrl/acrlissues/future/changingroles.htm)

<http://del.icio.us/rss/url/fa5be4b4401acf147ff6c8634b55cdda>

<http://del.icio.us/rss/subscriptions/{user}>

<http://del.icio.us/rss/network/{user}?private={private-key}>

<http://del.icio.us/rss/for/{user}?private={private-key}>

<http://del.icio.us/help/json/>

- 사용자의 네트워크에 존재하는 다른 사용자들을 <http://del.icio.us/feeds/json/network/{user}> 에서 볼 수 있다. 즉, 사용자가 관심 있어하는 사람들의 목록이다.
- 사용자의 팬을 <http://del.icio.us/feeds/json/fans/{user}>에서 볼 수 있다. 즉, 사용자에게 관심이 있는 사람들의 목록이다.

<http://www.twoantennas.com/projects/delicious-network-explorer/>

## 딜리셔스를 위한 서드파티 툴

<http://del.icio.us/help/thirdpartytools>

- 딜리셔스를 위한 공식 파이어폭스 애드온(<http://del.icio.us/help/firefox/extension>)은 여러분의 북마크와 태그를 브라우저의 사이드바를 통해 접근할 수 있게 해준다.
- PHP 라이브러리인 MySQLicious(<http://nanovivid.com/projects/mysqlicious/>)는 북마크를 MySQL 데이터베이스로 복사할 수 있게 해준다. 코드를 다운로드해서 지침대로 진행하면 된다. 그 결과로 북마크에 필요한 모든 데이터를 MySQL 데이터베이스에 추가할 수 있다. 문서에는 PHP4를 지원한다고 되어 있지만 내 경험상 PHP5에서도 동작한다.
- freshDel.icio.us(<http://freshdelicious.googlepages.com/>)는 링크가 유효한지를 체크하고 불필요한 북마크를 제거해주는 유틸리티이다.

## 서드파티 API 키트

- PHPDelicious(<http://www.ejeliot.com/pages/5>)

- Pydelicious(<http://code.google.com/p/pydelicious/>)
- Cocoalicious(좋은 데스크톱 툴로 활용할 수 있는 맥 OS X용 Cocoa 딜리셔스 클라이언트)
- 딜리셔스를 로컬 데이터베이스에 복제해둘 필요가 있다면, MySQLicious를 사용하여 복제할 수 있다.



## 야후! 북마크와 마이웹

- <http://bookmarks.yahoo.com/>
- <http://myweb2.search.Yahoo.com/>

■

■

■

- 사용자(rdhyee)의 태그목록을 가져오려면 다음을 사용하라.<sup>9</sup>

<http://api.search.yahoo.com/MyWebService/V1/tagSearch?appid=mashupguide.net&yahoodid=rdhyee&results=50>

- 특정 태그(mashup)를 가진 URL을 검색하려면 다음을 사용하라.

<http://search.yahooapis.com/MyWebService/V1/urlSearch?appid=mashupguide.net&tag=mashup>

- 주어진 태그(예를 들어, mashup)와 연관된 태그를 검색하려면 다음을 사용하라.<sup>10</sup>

<http://search.yahooapis.com/MyWebService/V1/relatedTags?appid=mashupguide.net&tag=mashup&results=50>

## 캐노티

<http://www.connotea.org/>

<http://www.connotea.org/wiki/WebAPI>

---

<sup>9</sup> <http://developer.yahoo.com/search/myweb/V1/tagSearch.html>

<sup>10</sup> <http://developer.yahoo.com/search/myweb/V1/relatedTags.html>



`http://www.connotea.org/user/{user}`

`http://www.connotea.org/user/timo`

`curl -v -u USER:PASSWORD http://www.connotea.org/data/noop`

- 기본 URL인 `http://www.connotea.org/data`
- 가져오기를 원하는 데이터의 유형(bookmarks, tags 또는 posts)을 나타내는 식별자. 즉, `/bookmarks` 나 `/tags` 를 사용할 수 있으며 빈 문자열을 명시하는 경우는 posts를 의미한다.
- 필터는 모두 선택 파라미터이며 user, tag, date, uri의 순서로 명시한다(즉, `/usr/{username}/tag/{tagname}/date/{YYYY-MM-DD}/uri/{uri-or-hash}`).
- 검색관련 파라미터 역시 선택 항목이며 반환할 결과 값의 개수와 시작 인덱스를 명시한다.(즉, `?q={free-text-string}&num={number-of-results}&star={starting-index}`)

`curl -u USER:PASSWORD http://www.connotea.org/data/tags/user/timo`

```
curl -u USER:PASSWORD http://www.connotea.org/data/bookmarks/user/timo
```

```
curl -u USER:PASSWORD http://www.connotea.org/data/user/timo
```





```
curl -u USER:PASSWORD http://www.connotea.org/data/bookmarks/user/timo/tag/chemistry
```

```
http://www.connotea.org/data/add
```

```
curl -v -u USER:PASSWORD --data-binary "uri=http://www.ala.org/ala/acrl/acrlissues/future/changingroles.htm&tags=library2.0+academic+ACRL+technology&usertitle=Essay%3A+Changing+Roles+of+Academic+and+Research+Libraries" http://www.connotea.org/data/edit
```

```
curl -v -u USER:PASSWORD --data-binary "uri=http://www.ala.org/ala/acrl/acrlissues/future/changingroles.htm" http://www.connotea.org/data/remove
```

## 플리커와 딜리셔스의 매쉬업





All your items (105)

del.icio.us

earlier | later > page 1 of 2



**title** My sick day (on Flickr)

**url** <http://www.flickr.com/photos/43472158@N00/1204981/>

**notes**

**tags** FlickrFavorite art

full-screen edit



**alfred (on Flickr)** add favorite  
to flickr/alfred - 12 hours ago



**faded pepper (on Flickr)** edit favorite  
to flickr/fadedpepper - 12 hours ago



**lest we forget (on Flickr)** add favorite  
to flickr/lestweforget - 12 hours ago



**Learning by drawing (on Flickr)** edit favorite  
to flickr/learningbydrawing - 12 hours ago



**C03-47-03 (on Flickr)** edit favorite  
to flickr/c03-47-03 - 12 hours ago



**Zuardi's Photoboard (on Flickr)** add favorite  
to flickr/zuardisphotoboard - 12 hours ago



**Series Lonely #1 (on Flickr)** edit favorite  
to flickr/serieslonely - 12 hours ago

**tags** 1001fr 2003 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 2521 2522 2523 2524 2525 2526 2527 2528 2529 2530 2531 2532 2533 2534 2535 2536 2537 2538 2539 2540 2541 2542 2543 2544 2545 2546 2547 2548 2549 2550 2551 2552 2553 2554 2555 2556 2557 2558 2559 2560 2561 2562 2563 2564 2565 2566 2567 2568 2569 2570 2571 2572 2573 2574 2575 2576 2577 2578 2579 2580 2581 2582 2583 2584 2585 2586 2587 2588 2589 2590 2591 2592 2593 2594 2595 2596 2597 2598 2599 2600 2601 2602 2603 2604 2605 2606 2607 2608 2609 2610 2611 2612 2613 2614 2615 2616 2617 2618 2619 2620 2621 2622 2623 2624 2625 2626 2627 2628 2629 2630 2631 2632 2633 2634 2635 2636 2637 2638 2639 2640 2641 2642 2643 2644 2645 2646 2647 2648 2649 2650 2651 2652 2653 2654 2655 2656 2657 2658 2659 2660 2661 2662 2663 2664 2665 2666 2667 2668 2669 2670 2671 2672 2673 2674 2675 2676 2677 2678 2679 2680 2681 2682 2683 2684 2685 2686 2687 2688 2689 2690 2691 2692 2693 2694 2695 2696 2697 2698 2699 2700 2701 2702 2703 2704 2705 2706 2707 2708 2709 2710 2711 2712 2713 2714 2715 2716 2717 2718 2719 2720 2721 2722 2723 2724 2725 2726 2727 2728 2729 2730 2731 2732 2733 2734 2735 2736 2737 2738 2739 2740 2741 2742 2743 2744 2745 2746 2747 2748 2749 2750 2751 2752 2753 2754 2755 2756 2757 2758 2759 2760 2761 2762 2763 2764 2765 2766 2767 2768 2769 2770 2771 2772 2773 2774 2775 2776 2777 2778 2779 2780 2781 2782 2783 2784 2785 2786 2787 2788 2789 2790 2791 2792 2793 2794 2795 2796 2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813 2814 2815 2816 2817 2818 2819 2820 2821 2822 2823 2824 2825 2826 2827 2828 2829 2830 2831 2832 2833 2834 2835 2836 2837 2838 2839 2840 2841 2842 2843 2844 2845 2846 2847 2848 2849 2850 2851 2852 2853 2854 2855 2856 2857 2858 2859 2860 2861 2862 2863 2864 2865 2866 2867 2868 2869 2870 2871 2872 2873 2874 2875 2876 2877 2878 2879 2880 2881 2882 2883 2884 2885 2886 2887 2888 2889 2890 2891 2892 2893 2894 2895 2896 2897 2898 2899 2900 2901 2902 2903 2904 2905 2906 2907 2908 2909 2910 2911 2912 2913 2914 2915 2916 2917 2918 2919 2920 2921 2922 2923 2924 2925 2926 2927 2928 2929 2930 2931 2932 2933 2934 2935 2936 2937 2938 2939 2940 2941 2942 2943 2944 2945 2946 2947 2948 2949 2950 2951 2952 2953 2954 2955 2956 2957 2958 2959 2960 2961 2962 2963 2964 2965 2966 2967 2968 2969 2970 2971 2972 2973 2974 2975 2976 2977 2978 2979 2980 2981 2982 2983 2984 2985 2986 2987 2988 2989 2990 2991 2992 2993 2994 2995 2996 2997 2998 2999 3000 3001 3002 3003 3004 3005 3006 3007 3008 3009 3010 3011 3012 3013 3014 3015 3016 3017 3018 3019 3020 3021 3022 3023 3024 3025 3026 3027 3028 3029 3030 3031 3032 3033 3034 3035 3036 3037 3038 3039 3040 3041 3042 3043 3044 3045 3046 3047 3048 3049 3050 3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061 3062 3063 3064 3065 3066 3067 3068 3069 3070 3071 3072 3073 3074 3075 3076 3077 3078 3079 3080 3081 3082 3083 3084 3085 3086 3087 3088 3089 3090 3091 3092 3093 3094 3095 3096 3097 3098 3099 3100 3101 3102 3103 3104 3105 3106 3107 3108 3109 3110 3111 3112 3113 3114 3115 3116 3117 3118 3119 3120 3121 3122 3123 3124 3125 3126 3127 3128 3129 3130 3131 3132 3133 3134 3135 3136 3137 3138 3139 3140 3141 3142 3143 3144 3145 3146 3147 3148 3149 3150 3151 3152 3153 3154 3155 3156 3157 3158 3159 3160 3161 3162 3163 3164 3165 3166 3167 3168 3169 3170 3171 3172 3173 3174 3175 3176 3177 3178 3179 3180 3181 3182 3183 3184 3185 3186 3187 3188 3189 3190 3191 3192 3193 3194 3195 3196 3197 3198 3199 3200 3201 3202 3203 3204 3205 3206 3207 3208 3209 3210 3211 3212 3213 3214 3215 3216 3217 3218 3219 3220 3221 3222 3223 3224 3225 3226 3227 3228 3229 3230 3231 3232 3233 3234 3235 3236 3237 3238 3239 3240 3241 3242 3243 3244 3245 3246 3247 3248 3249 3250 3251 3252 3253 3254 3255 3256 3257 3258 3259 3260 3261 3262 3263 3264 3265 3266 3267 3268 3269 3270 3271 3272 3273 3274 3275 3276 3277 3278 3279 3280 3281 3282 3283 3284 3285 3286 3287 3288 3289 3290 3291 3292 3293 3294 3295 3296 3297 3298 3299 3300 3301 3302 3303 3304 3305 3306 3307 3308 3309 3310 3311 3312 3313 3314 3315 3316 3317 3318 3319 3320 3321 3322 3323 3324 3325 3326 3327 3328 3329 3330 3331 3332 3333 3334 3335 3336 3337 3338 3339 3340 3341 3342 3343 3344 3345 3346 3347 3348 3349 3350 3351 3352 3353 3354 3355 3356 3357 3358 3359 3360 3361 3362 3363 3364 3365 3366 3367 3368 3369 3370 3371 3372 3373 3374 3375 3376 3377 3378 3379 3380 3381 3382 3383 3384 3385 3386 3387 3388 3389 3390 3391 3392 3393 3394 3395 3396 3397 3398 3399 3400 3401 3402 3403 3404 3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415 3416 3417 3418 3419 3420 3421 3422 3423 3424 3425 3426 3427 3428 3429 3430 3431 3432 3433 3434 3435 3436 3437 3438 3439 3440 3441 3442 3443 3444 3445 3446 3447 3448 3449 3450 3451 3452 3453 3454 3455 3456 3457 3458 3459 3460 3461 3462 3463 3464 3465 3466 3467 3468 3469 3470 3471 3472 3473 3474 3475 3476 3477 3478 3479 3480 3481 3482 3483 3484 3485 3486 3487 3488 3489 3490 3491 3492 3493 3494 3495 3496 3497 3498 3499 3500 3501 3502 3503 3504 3505 3506 3507 3508 3509 3510 3511 3512 3513 3514 3515 3516 3517 3518 3519 3520 3521 3522 3523 3524 3525 3526 3527 3528 3529 3530 3531 3532 3533 3534 3535 3536 3537 3538 3539 3540 3541 3542 3543 3544 3545 3546 3547 3548 3549 3550 3551 3552 3553 3554 3555 3556 3557 3558 3559 3560 3561 3562 3563 3564 3565 3566 3567 3568 3569 3570 3571 3572 3573 3574 3575 3576 3577 3578 3579 3580 3581 3582 3583 3584 3585 3586 3587 3588 3589 3590 3591 3592 3593 3594 3595 3596 3597 3598 3599 3600 3601 3602 3603 3604 3605 3606 3607 3608 3609 3610 3611 3612 3613 3614 3615 3616 3617 3618 3619 3620 3621 3622 3623 3624 3625 3626 3627 3628 3629 3630 3631 3632 3633 3634 3635 3636 3637 3638 3639 3640 3641 3642 3643 3644 3645 3646 3647 3648 3649 3650 3651 3652 3653 3654 3655 3656 3657 3658 3659 3660 3661 3662 3663 3664 3665 3666 3667 3668 3669 3670 3671 3672 3673 3674 3675 3676 3677 3678 3679 3680 3681 3682 3683 3684 3685 3686 3687 3688 3689 3690 3691 3692 3693 3694 3695 3696 3697 3698 3699 3700 3701 3702 3703 3704 3705 3706 3707 3708 3709 3710 3711 3712 3713 3714 3715 3716 3717 3718 3719 3720 3721 3722 3723 3724 3725 3726 3727 3728 3729 3730 3731 3732 3733 3734 3735 3736 3737 3738 3739 3740 3741 3742 3743 3744 3745 3746 3747 3748 3749 3750 3751 3752 3753 3754 3755 3756 3757 3758 3759 3760 3761 3762 3763 3764 3765 3766 3767 3768 3769 3770 3771 3772 3773 3774 3775 3776 3777 3778 3779 3780 3781 3782 3783 3784 3785 3786 3787 3788 3789 3790 3791 3792 3793 3794 3795 3796 3797 3798 3799 3800 3801 3802 3803 3804 3805 3806 3807 3808 3809 3810 3811 3812 3813 3814 3815 3816 3817 3818 3819 3820 3821 3822 3823 3824 3825 3826 3827 3828 3829 3830 3831 3832 3833 3834 3835 3836 3837 3838 3839 3840 3841 3842 3843 3844 3845 3846 3847 3848 3849 3850 3851 3852 3853 3854 3855 3856 3857 3858 3859 3860 3861 3862 3863 3864 3865 3866 3867 3868 3869 3870 3871 3872 3873 3874 3875 3876 3877 3878 3879 3880 3881 3882 3883 3884 3885 3886 3887 3888 3889 3890 3891 3892 3893 3894 3895 3896 3897 3898 3899 3900 3901 3902 3903 3904 3905 3906 3907 3908 3909 3910 3911 3912 3913 3914 3915 3916 3917 3918 3919 3920 3921 3922 3923 3924 3925 3926 3927 3928 3929 3930 3931 3932 3933 3934 3935 3936 3937 3938 3939 3940 3941 3942 3943 3944 3945 3946 3947 3948 3949 3950 3951 3952 3953 3954 3955 3956 3957 3958 3959 3960 3961 3962 3963 3964 3965 3966 3967 3968 3969 3970 3971 3972 3973 3974 3975 3976 3977 3978 3979 3980 3981 3982 3983 3984 3985 3986 3987 3988 3989 3990 3991 3992 3993 3994 3995 3996 3997 3998 3999 4000 4001 4002 4003 4004 4005 4006 4007 4008 4009 4010 4011 4012 4013 4014 4015 4016 4017 4018 4019 4020 4021 4022 4023 4024 4025 4026 4027 4028 4029 4030 4031 4032 4033 4034 4035 4036 4037 4038 4039 4040 4041 4042 4043 4044 4045 4046 4047 4048 4049 4050 4051 4052 4053 4054 4055 4056 4057 4058 4059 4060 4061 4062 4063 4064 4065 4066 4067 4068 4069 4070 4071 4072 4073 4074 4075 4076 4077 4078 4079 4080 4081 4082 4083 4084 4085 4086 4087 4088 4089 4090 4091 4092 4093 4094 4095 4096 4097 4098 4099 4100 4101 4102 4103 4104 4105 4106 4107 4108 4109 4110 4111 4112 4113 4114 4115 4116 4117 4118 4119 4120 4121 4122 4123 4124 4125 4126 4127 4128 4129 4130 4131 4132 4133 4134 4135 4136 4137 4138 4139 4140 4141 4142 4143 4144 4145 4146 4147 4148 4149 4150 4151 4152 4153 4154 4155 4156 4157 4158 4159 4160 4161 4162 4163 4164 4165 4166 4167 4168 4169 4170 4171 4172 4173 4174 4175 4176 4177 4178 4179 4180 4181 4182 4183 4184 4185 4186 4187 4188 4189 4190 4191 4192 4193 4194 4195 4196 4197 4198 4199 4200 4201 4202 4203 4204 4205 4206 4207 4208 4209 4210 4211 4212 4213 4214 4215 4216 4217 4218 4219 4220 4221 4222 4223 4224 4225 4226 4227 4228 4229 4230 4231 4232 4233 4234 4235 4236 4237 4238 4239 4240 4241 4242 4243 4244 4245 4246 4247 4248 4249 4250 4251 4252 4253 4254 4255 4256 4257 4258 4259 4260 4261 4262 4263 4264 4265 4266 4267 4268 4269 4270 4271 4272 4273 42



- 딜리셔스의 태그를 기준으로 하여 플리커 사진을 그룹으로 묶고 사진들을 슬라이드쇼로 제공하도록 딜리셔스 API를 사용할 수 있다.
- 공개된 '좋아하는 사진'뿐만 아니라 모든 사용자의 '좋아하는 사진'을 가져오도록 스크립트를 수정할 수 있다.
- '좋아하는 사진' 외에, 개별적인 사진이나 여러분이 가지고 있는 사진 그룹 또는 임의적으로 플리커에서 수집한 사진들을 딜리셔스에 추가할 수 있다.
- 플리커와 딜리셔스 간에 '좋아하는 사진'을 동기화할 수 있다.

## 요약



# 15

---

## 온라인 캘린더와 이벤트 수집기

---

---

1 SyncML은 이제 Open Mobile Alliance Data Synchronization and Device Management로 알려져 있다.

2 <http://www.coldsync.org/description.html>

- 이번 장은 프로그래밍 하지 않고 캘린더에서 어떤 데이터를 추가하고 가져올 수 있는지에 대해 iCalendar와 다양한 XML 피드를 예로 들어서 설명한다.
- 구글 캘린더와 30boxes.com을 이용하여 어떻게 개인 캘린더를 프로그래밍하는지, 데이터를 어떻게 이벤트 데이터의 소스로부터 캘린더로 이동시키는지, 어떻게 upcoming.yahoo.com과 eventful.com과 같은 이벤트 수집기에 이벤트 정보를 추가하는지 다룬다.

## 구글 캘린더

## 최종 사용자로서 구글 캘린더 설정하기

<http://www.google.com/calendar>

■

■

■

■

■

■

---

4 <http://www.google.com/calendar/render?mode=gallery&cat=POPULAR>

5 <http://www.google.com/support/calendar/bin/answer.py?answer=34577&hl=en>

6 <http://www.google.com/support/calendar/bin/answer.py?answer=53231&topic=8556>

7 <http://www.google.com/support/calendar/bin/answer.py?answer=61527&topic=8556>

## 구글 캘린더를 이용하는 몇 가지 사용패턴

- 완전히 개인적인 이벤트를 기록하는 캘린더. 이 캘린더는 어느 누구와도 공유하지 않도록 설정하였다.
- 가까운 친구들을 위한 가족/친구용 캘린더. 이 캘린더 또한 어느 누구와도 공유하지 않도록 설정하였지만 친구들과 가족들의 이메일 주소를 추가하였다.
- 참석할 이벤트가 표기된 '레이몬드 이(Raymond Yee)의 공개 이벤트'라는 이름의 캘린더. 이 캘린더는 모든 사람과 공유하도록 설정하였다.
- 이번 장에서 구글 캘린더를 어떻게 프로그램 하는지 설명하기 위해 설정한 공개 캘린더인 Mashup Guide Demo Calendar라는 이름의 캘린더.
- 이 캘린더를 누구와 공유할 것인가?(즉, 이 캘린더가 나 자신을 위한 것인지, 특정 그룹의 사용자를 위한 것인지, 또는 세상의 모두를 위한 것인지)
- 해당 캘린더의 주제

## 캘린더를 공유하기

- HTML
- iCalendar(또한 구어체로는 iCal로도 사용된다)<sup>8</sup>
- XML(보다 엄밀히 말하면, Atom 피드이다)

---

8 <http://www.google.com/support/calendar/bin/answer.py?answer=34578&hl=en> and <http://www.google.com/support/calendar/bin/answer.py?answer=37104&ctx=sibling>

9 <http://en.wikipedia.org/wiki/ICalendar>

9imfjk71chkcs66t1i436je0s0%40group.calendar.google.com

<http://www.google.com/calendar/embed?src={userID}>

[http://www.google.com/calendar/embed?src=9imfjk71chkcs66t1i436je0s0%40group.  
calendar.google.com](http://www.google.com/calendar/embed?src=9imfjk71chkcs66t1i436je0s0%40group.calendar.google.com)

<http://www.google.com/calendar/ical/{userID}/{visibility}/{projection}.ics>

[http://www.google.com/calendar/ical/9imfjk71chkcs66t1i436je0s0%40group.calendar.  
google.com/public/full.ics](http://www.google.com/calendar/ical/9imfjk71chkcs66t1i436je0s0%40group.calendar.google.com/public/full.ics)

[http://www.google.com/calendar/ical/9imfjk71chkcs66t1i436je0s0%40group.calendar.  
google.com/public/basic.ics](http://www.google.com/calendar/ical/9imfjk71chkcs66t1i436je0s0%40group.calendar.google.com/public/basic.ics)

<http://www.google.com/calendar/feeds/{userID}/{visibility}/{projection}>

[http://www.google.com/calendar/feeds/9imfjk71chkcs66t1i436je0s0%40group.calendar.  
google.com/public/basic](http://www.google.com/calendar/feeds/9imfjk71chkcs66t1i436je0s0%40group.calendar.google.com/public/basic)

## 구글 캘린더가 제공하는 피드 포맷에 대해 알아보자

### iCalendar/iCal

<http://tools.ietf.org/html/rfc2445>

- Events, BusyTime, To-dos와 Journal Entries(RFC2446)를 스케줄링하는 ‘iCalendar 통신레이어 비의존 호환 프로토콜(iCalendar Transport-Independent Interoperability Protocol, iTIP)’은 어떻게 캘린더 서버가 이벤트 일정관리 정보를 교환하는지 설명한다.<sup>13</sup>

---

10 <http://www.google.com/support/calendar/bin/answer.py?answer=34576&hl=en>

11 <http://code.google.com/apis/calendar/overview.html>

12 [http://code.google.com/apis/calendar/developers\\_guide\\_protocol.html](http://code.google.com/apis/calendar/developers_guide_protocol.html)

13 <http://tools.ietf.org/html/rfc2446>



- ‘iCalendar 메시지 기반 호환 프로토콜(iCalendar Message-Based Interoperability Protocol, iMIP) (RFC 2447)’은 이벤트 일정관리 정보를 이메일로 교환하는 방법을 설명한다.<sup>14</sup>

- iCalendar는 VCALENDAR라는 최상위 오브젝트를 갖는다.
- 그 하위로 VEVENT, VTODO, VJOURNAL, VFREEBUSY와 같은 하위 오브젝트가 있다.

---

14 <http://tools.ietf.org/html/rfc2447>

15 <http://en.wikipedia.org/wiki/ICalendar>

16 <http://www.calconnect.org/calendarstandards.shtml>

17 <http://en.wikipedia.org/wiki/XCal>

18 <http://tools.ietf.org/html/rfc2445#section-4.4>

```
curl "http://www.google.com/calendar/ical/9imfjk71chkcs66t1i436je0s0%40group.  
calendar.google.com/public/basic.ics"
```





- ‘인터넷 기반 캘린더 서비스 구현에 대한 지침(Guide to Internet Calendaring, <http://www.ietf.org/rfc/rfc3283.txt>)’ 을 읽도록 하라.
- 많은 표준들(<http://www.calconnect.org/calendaringstandards.shtml>)이 존재하지만 특히 RFC 2445를 숙지하라.
- iCalendar는 기능이 아주 많기 때문에 iCalendar와 호환된다고 주장하는 캘린더, 서버, 또는 라이브러리들이 이러한 모든 기능을 동일한 수준으로 구현하지 않는다는 것을 주의하라.
- 커뮤니티에서 많은 문제들을 해결하기 위해 노력하고 있다. 그 결과로 반복적으로 발생하는 이벤트와 타임존의 처리방법을 고민하는 CalConnect와 같은 조직이 만들어졌다 (<http://calconnect.org/recommendations.shtml>).

- iCalendar 구현체들 간의 상호 호환성은 아직 해결해야 할 숙제로 남아 있으므로<sup>19</sup> 하나의 시스템을 사용하여 다른 시스템에서 생성한 iCalendar 파일을 해석할 때 문제가 발생할 수도 있다.
- iCalendar 라이브러리의 품질을 파악하는 것이 어렵기는 하지만 좋은 프로그래밍 라이브러리를 선택하여 iCalendar를 파싱하고 생성하라.
- 표준을 업데이트하는 작업이 진행중(<http://www.ietf.org/html.charters/calsify-charter.html>)이므로 참고하도록 하자.

<http://severinghaus.org/projects/icv/?url=http%3A%2F%2Fwww.google.com%2Fcalendar%2Fical%2F9imfjk71chkcs66t1i436je0s0%2540group.calendar.google.com%2Fpublic%2Fbasic.ics>


## Atom 포맷의 구글 캘린더 데이터

`curl http://www.google.com/calendar/feeds/9imfjk71chkcs66t1i436je0s0%40group.calendar.google.com/public/basic`



<sup>19</sup> <http://www.calconnect.org/iop/testing.shtml> and <http://www.calconnect.org/interop/uc%20berkeley%20interop%20testing.pdf>



- 
- 피드는 Atom 포맷(이미 4장에서 배운)으로 표현된다.
  - 이 데이터는 GData 확장 엘리먼트<sup>20</sup>를 사용하며 확장 엘리먼트에는 OpenSearch와 구글 캘린더 확장기능과 같은 것이 있다.<sup>21</sup>

## GData 기반의 캘린더 API를 직접 사용하기

---

<sup>20</sup> <http://code.google.com/apis/gdata/elements.html>

<sup>21</sup> <http://code.google.com/apis/calendar/reference.html#Elements>

<sup>22</sup> [http://code.google.com/apis/calendar/developers\\_guide\\_protocol.html](http://code.google.com/apis/calendar/developers_guide_protocol.html)

<http://code.google.com/apis/gdata/clientlibs.html>

[http://code.google.com/apis/calendar/developers\\_guide\\_protocol.html](http://code.google.com/apis/calendar/developers_guide_protocol.html)

<http://code.google.com/apis/calendar/reference.html>

## 사용자 인증 토큰 획득하기

<http://code.google.com/apis/gdata/auth.html>

<https://www.google.com/accounts/ClientLogin>

Email : 구글 이메일(예를 들어, raymond.yee@gmail.com)

Password : 구글 패스워드

source : 프로그램을 식별하기 위한 '회사명-애플리케이션명-버전ID' 형태의 문자열(예를 들면, mashupguide.net-Chap15-v1)

service : 구글 서비스의 이름으로 이 경우에는 cl 이다.

```
curl -v -X POST -d "Passwd={passwd}&source=mashupguide.net-Chap15-v1&Email=raymond.yee%40gmail.com&service=cl" https://www.google.com/accounts/ClientLogin
```

Authorization: GoogleLogin auth=[AUTH-TOKEN]

▶ **TIP!** curl 에서, -H 옵션을 사용하여 헤더에 추가할 수 있다. 즉 아래와 같이 사용하면 된다.  
-H "Authorization: GoogleLogin auth=[AUTH-TOKEN]"

▶ **TIP!** HTTP GET요청에서, -L 옵션을 사용하면 curl이 자동으로 리다이렉트를 처리하도록 할 수 있다.

구글 캘린더에서 사용할 수 있는 피드



<http://code.google.com/apis/calendar/reference.html#Feeds>

## visibility와 projection

<http://code.google.com/apis/calendar/reference.html#Projection>

- full(읽기와 쓰기가 가능)
- free-busy(항상 읽기만 가능). 이 피드는 이벤트에 대한 최소한의 정보만을 보여주지만 이벤트 기간에 대한 정보를 포함하고 있다(다른 말로 하면, <gd:when> 엘리먼트를 포함한다).
- basic(항상 읽기만 가능). basic은 어떠한 확장 엘리먼트도 가지지 않는 Atom 피드를 생성한다. 즉 <atom:summary>와 <atom:content> 엘리먼트가 이벤트 관련 데이터를 포함하는 HTML description을 가지게 된다.

## 캘린더 피드

meta-feed

<http://www.google.com/calendar/feeds/default>

```
curl -L -X GET -H "Authorization: GoogleLogin auth=[AUTH-TOKEN]" http://www.  
google.com/calendar/feeds/default
```



- rel="alternate"인 link의 href 값은 다음과 같다.

<http://www.google.com/calendar/feeds/raymond.yee%40gmail.com/private/full>

만약 사용자 인증 후에 GET 요청을 위 URL로 전송하면 기본 캘린더의 모든 이벤트를 포함하는 이벤트 피드를 수신하게 한다.

이 피드의 URL이 아래 형태에 부합된다는 것에 주의하자.

<http://www.google.com/calendar/feeds/{userID}/{privacy}/{projection}>

여기서 user ID는 raymond.yee%40gmail.com이며 visibility는 private, 그리고 projection은 full이다.

- rel=http://schemas.google.com/acl/2007#accessControlList 인 link의 href 값은 아래와 같으며 다음 피드는 해당 캘린더에 대한 접근제어 목록을 제공한다.

<http://www.google.com/calendar/feeds/raymond.yee%40gmail.com/acl/full>

이 캘린더에는 아래에서 볼 수 있듯이 하나의 entry가 존재한다(내 기본 캘린더에 권한을 가지고 있는 사용자는 나뿐이다).



- rel="self" 인 link의 href 값은 아래와 같으며

<http://www.google.com/calendar/feeds/default/ramond.yee%40gmail.com>

사용자(raymond.yee@gmail.com)가 접근할 수 있는 모든 캘린더 대신에 이 피드는 기본 캘린더에 대해 하나의 entry를 반환한다.

## allcalendars

<http://www.google.com/calendar/feeds/default/allcalendars/full>

```
curl -L -X GET -H "Authorization: GoogleLogin auth=[AUTH-TOKEN]" http://www.google.com/calendar/feeds/default/allcalendars/full
```

**\* NOTE!** meta-feed와 allcalendars 모두 사용자가 접근할 수 있는 모든 캘린더의 목록을 제공하기 때문에 meta-feed와 allcalendars의 차이점이 궁금할 것이다. meta-feed가 읽기만 가능한 반면에 projection 값이 full인 allcalendars는 읽고 쓰기가 가능하다. 만약 projection 값이 basic인 allcalendars 피드에 접근을 시도하면(meta-feed와 유사한 결과를 얻기 위해서), 'unknown visibility found' 에러가 발생한다.

<http://www.google.com/calendar/embed?src=ht3jlfac5lfd6263ulfh4tql8%40group.calendar.google.com>

ht3jlfaac5lfd6263ulfh4tql8%40group.calendar.google.com

```
curl -v -X POST --data-binary "@phases_of_moon_entry.xml" -H "Content-Type: application/atom+xml" -H "Authorization: GoogleLogin auth=[AUTH-TOKEN]" http://www.google.com/calendar/feeds/default/allcalendars/full
```

<http://www.google.com/calendar/feeds/default/allcalendars/full?gsessionid={gsessionid}>

<http://www.google.com/calendar/feeds/default/allcalendars/full?gsessionid=GUWxgPh61GQ>

```
curl -v -X POST --data-binary "@phases_of_moon_entry.xml" -H "Content-Type: application/atom+xml" -H "Authorization: GoogleLogin auth=[AUTH-TOKEN]" http://www.google.com/calendar/feeds/default/allcalendars/full?gsessionid={gsessionid}
```





```
curl -v -X DELETE -H "Authorization: GoogleLogin auth=[AUTH-TOKEN]" http://www.  
google.com/calendar/feeds/default/allcalendars/full/ht3jlfaac5lfd6263ulfh tq18%40group.  
calendar.google.com?gsessionid={gsessionid}
```

**owncalendars**

<http://www.google.com/calendar/feeds/default/owncalendars/full>

```
curl -v -L -X GET -H "Authorization: GoogleLogin auth=[AUTH-TOKEN]" http://www.google.com/
calendar/feeds/default/owncalendars/full
```

```
curl -v -X POST --data-binary "@book_writing_calendar_entry.xml" -H "Content-Type:
application/atom+xml " -H "Authorization: GoogleLogin auth=[AUTH-TOKEN]" http://
www.google.com/calendar/feeds/default/owncalendars/full?gsessionid={gsession-id}
```

```
http://www.google.com/calendar/feeds/default/owncalendars/full/{userID}
```

```
http://www.google.com/calendar/feeds/default/owncalendars/full/{userID}
```

## Event 피드

```
9imfjk71chkcs66t1i436je0s0%40group.calendar.google.com
```

`http://www.google.com/calendar/feeds/{userID}/{privacy}/{projection}`

`http://www.google.com/calendar/feeds/{userID}/public/full`

`http://www.google.com/calendar/feeds/9imfjk71chkcs66t1i436je0s0%40group.calendar.google.com/public/full`

`http://www.google.com/calendar/feeds/{userID}/private/full`

`http://www.google.com/calendar/feeds/9imfjk71chkcs66t1i436je0s0%40group.calendar.google.com/private/full`

`curl -v -X POST --data-binary "@project_showcase_event.xml" -H "Content-Type: application/atom+xml" -H "Authorization: GoogleLogin auth=[AUTH-TOKEN]" http://www.google.com/calendar/feeds/{userID}/private/full?gsessionid={gsessionid}`



## 구글 캘린더용 PHP API 키트 사용하기

[http://code.google.com/apis/calendar/developers\\_guide\\_php.html](http://code.google.com/apis/calendar/developers_guide_php.html)

<http://framework.zend.com/>

<http://framework.zend.com/manual/en/zend.gdata.calendar.html>

<http://examples.mashupguide.net/lib/ZendFramework/>





## 구글 캘린더용 파이썬 API 키트 사용하기

[http://code.google.com/apis/calendar/developers\\_guide\\_python.html](http://code.google.com/apis/calendar/developers_guide_python.html)

<http://code.google.com/p/gdata-python-client/downloads/list>

<http://gdata-python-client.googlecode.com/svn/trunk/>

**\* NOTE!** 다음 코드는 파이썬 2.5 이후의 버전에 함께 배포되는 ElementTree 라이브러리에 의존성이 있다. <http://effbot.org/zone/element-index.htm>에서 ElementTree를 다운로드하는 방법을 찾아 볼 수 있다.

30boxes.com

- <http://30boxes.com/developers>
- <http://30boxes.com/api/>

## 최종 사용자 튜토리얼

1. 계정이 없다면 아래 URL에서 가입을 하도록 하자.

<http://30boxes.com/signup>

2. 계정을 가지고 있다면 다음 URL에서 로그인하라.

<http://30boxes.com/login>

3. help 섹션을 참고하면 어떻게 30boxes.com에서 다양한 작업을 할 수 있는지 배울 수 있다(<http://30boxes.com/help>).

- 친구들은 이벤트를 비공개로 표시하지 않는다면 전체 캘린더를 볼 수 있다.
- 친구들은 특정 태그로 표시된 이벤트를 볼 수 있다.
- 친구들은 친구 페이지buddy page의 내용만을 볼 수 있다.

## 30boxes.com의 API

<http://30boxes.com/api/>

<http://30boxes.com/api/api.php?method=getKeyForUser>

■ test.ping:<sup>23</sup>

`http://30boxes.com/api/api.php?method=test.Ping&apiKey={APIKEY}`

■ user.FindByEmail:

`http://30boxes.com/api/api.php?method=user.FindByEmail&apiKey={APIKEY}&email=yee@berkeley.edu`

- user.Authorize: 많은 메서드들은 사용자 인증을 필요로 하며, 사용자 인증 과정에서 사용자 인증 토큰이 발급된다. 이번 예제에서는 내가 업로드한 작은 사진을 애플리케이션의 아이콘으로 사용한다.<sup>24</sup> user.FindByEmail을 호출할 때, 선택 파라미터인 returnUrl을 누락시켰다는 것에 주목하라.

`http://30boxes.com/api/api.php?method=user.Authorize&apiKey={APIKEY}&applicationName={application-name}&applicationLogoUrl={url}`

예를 들면, 아래와 같이 호출하면

`http://30boxes.com/api/api.php?method=user.Authorize&apiKey={APIKEY}&applicationName=Raymond+Yee&applicationLogoUrl=http%3A%2F%2Ffarm1.static.flickr.com%2F4%2F5530475_48f80eece8_s.jpg`

사용자 인증 토큰을 발급받게 될 것이고 여기서는 {AUTHTOKEN}으로 표시한다.

■ user.GetAllInfo:

`http://30boxes.com/api/api.php?method=user.GetAllInfo&apiKey={APIKEY}&authorizedUserToken={AUTHTOKEN}`

위 명령어를 실행하면 다음과 유사한 결과 값을 반환받게 된다.

<sup>23</sup> <http://30boxes.com/api/#t>

<sup>24</sup> [http://farm1.static.flickr.com/4/5530475\\_48f80eece8\\_s.jpg](http://farm1.static.flickr.com/4/5530475_48f80eece8_s.jpg)

■ events.Get:

`http://30boxes.com/api/api.php?method=events.Get&apiKey={APIKEY}&authorizedUserToken={AUTHTOKEN}&start=2007-01-01&end=2007-09-01`

위 명령어를 실행하면 아래와 같은 결과를 반환받게 된다.



**\* NOTE!** end 파라미터의 값은 시작일로부터 180일을 넘을 수 없다.

- event.GetDisplayList(이벤트 목록을 정렬된 형태로 반환받기 위해):

`http://30boxes.com/api/api.php?method=events.GetDisplayList&apiKey={APIKEY}&authorizedUserToken={AUTHTOKEN}&start=2007-01-01&end=2007-09-01`

- todos.Get:

`http://30boxes.com/api/api.php?method=todos.Get&apiKey={APIKEY}&authorizedUserToken={AUTHTOKEN}`

- todos.Add:

`http://30boxes.com/api/api.php?method=todos.Add&apiKey={APIKEY}&authorizedUserToken={AUTHTOKEN}&text=Eat+more+veggies&externalUID=123456x`

- todos.Update:

`http://30boxes.com/api/api.php?method=todos.Update&apiKey={APIKEY}&authorizedUserToken={AUTHTOKEN}&text=Eat+more+veggies+and+fruit&todoId=123110&externalUID=123456x`

- todos.Delete:

`http://30boxes.com/api/api.php?method=todos.Delete&apiKey={APIKEY}&authorizedUserToken={AUTHTOKEN}&text=Eat+more+veggies+and+fruit&todoId=123110`

- events.AddByOneBox:

`http://30boxes.com/api/api.php?method=events.AddByOneBox&apiKey={APIKEY}&authorizedUserToken={AUTHTOKEN}&event=eat+some+sushi+tomorrow+at+7pm`

**\* NOTE!** 30boxes.com의 파이썬 API 래퍼는 <http://trentm.com/projects/thirtyboxes/>에서 찾을 수 있다.

## 이벤트 수집기

### Upcoming.yahoo.com

`http://upcoming.yahoo.com/`

`http://upcoming.yahoo.com/event/{event-id}/`

<http://upcoming.yahoo.com/event/76140>

## 검색결과로부터 피드 얻기

<http://upcoming.yahoo.com/search/?type=Events&rt=1&q=bach&loc=Berkeley%2C+California%2C+United+States>

<http://upcoming.yahoo.com/search/?type=Events&rt=1&q={q}&loc={location}&sort={sort}>

[http://upcoming.yahoo.com/syndicate/v2/search\\_all/?q=bach&loc=Berkeley%2C+California%2C+United+States&rt=1](http://upcoming.yahoo.com/syndicate/v2/search_all/?q=bach&loc=Berkeley%2C+California%2C+United+States&rt=1)



webcal://upcoming.yahoo.com/calendar/v2/search\_all/?q=bach&loc=Berkeley%2C+California%2C+United+States&rt=1

http://upcoming.yahoo.com/calendar/v2/search\_all/?q=bach&loc=Berkeley%2C+California%2C+United+States&rt=1

http://pipes.yahoo.com/pipes/pipe.info?\_id=G1qEg8WA3BGZNw9ELO2fWQ

**\* NOTE!** 파이프가 날짜 범위나 카테고리 와 같은 upcoming.yahoo.com의 검색 옵션을 포함 하도록 확장시킬 수도 있다.

[http://pipes.yahoo.com/pipes/pipe.info?q=Bach&loc=Berkeley%2C+CA&sort=r&\\_cmd=Run+Pipe&\\_id=G1qEg8WA3BGZNw9ELO2fWQ&\\_run=1](http://pipes.yahoo.com/pipes/pipe.info?q=Bach&loc=Berkeley%2C+CA&sort=r&_cmd=Run+Pipe&_id=G1qEg8WA3BGZNw9ELO2fWQ&_run=1)

[http://pipes.yahoo.com/pipes/pipe.run?\\_id=G1qEg8WA3BGZNw9ELO2fWQ&\\_render=rss&loc=Berkeley%2C+CA&q=Bach&sort=r](http://pipes.yahoo.com/pipes/pipe.run?_id=G1qEg8WA3BGZNw9ELO2fWQ&_render=rss&loc=Berkeley%2C+CA&q=Bach&sort=r)

[http://pipes.yahoo.com/pipes/pipe.run?\\_id=G1qEg8WA3BGZNw9ELO2fWQ&\\_render=kml&loc=Berkeley%2C+CA&q=Bach&sort=r](http://pipes.yahoo.com/pipes/pipe.run?_id=G1qEg8WA3BGZNw9ELO2fWQ&_render=kml&loc=Berkeley%2C+CA&q=Bach&sort=r)

[http://maps.google.com/maps?q=http:%2F%2Fpipes.yahoo.com%2Fpipes%2Fpipe.run%3F\\_id%3DG1qEg8WA3BGZNw9ELO2fWQ%26\\_render%3Dkml%26loc%3DBerkeley%252C%2BCA%26q%3DBach%26sort%3Dr&ie=UTF8](http://maps.google.com/maps?q=http:%2F%2Fpipes.yahoo.com%2Fpipes%2Fpipe.run%3F_id%3DG1qEg8WA3BGZNw9ELO2fWQ%26_render%3Dkml%26loc%3DBerkeley%252C%2BCA%26q%3DBach%26sort%3Dr&ie=UTF8)

## 읽기만 가능한 API

<http://upcoming.yahoo.com/services/api/>

<http://upcoming.yahoo.com/services/api/keygen.php>

<http://upcoming.yahooapis.com/services/rest/>

<http://upcoming.yahoo.com/services/api/event.search.php>

[http://upcoming.yahooapis.com/services/rest/?api\\_key={api\\_key}&method=event.search&search\\_text=bach&location=Berkeley%@C+California](http://upcoming.yahooapis.com/services/rest/?api_key={api_key}&method=event.search&search_text=bach&location=Berkeley%@C+California)



- event\_id를 가지고 공개 이벤트에 대한 정보를 event.getInfo를 사용하여 가져올 수 있다. 예를 들면 아래와 같이 WWW2008 컨퍼런스(<http://upcoming.yahoo.com/event/205875>)의 event\_id 값을 사용하여 이벤트를 요청할 수 있다.

[http://upcoming.yahooapis.com/services/rest/?method=event.getInfo&api\\_key={api-key}&event\\_id=205875](http://upcoming.yahooapis.com/services/rest/?method=event.getInfo&api_key={api-key}&event_id=205875)

위 요청은 다음과 같은 결과 값을 반환한다.



- metro.getForLatLon을 사용하여 주어진 위도와 경도에 해당하는 위치정보를 가져올 수도 있다. 캘리포니아, 버클리에 있는 UC 버클리 캠퍼스의 건물에 해당하는 아래 위도와 경도

37.869111,-122.260634

를 사용하여 다음의 요청을 구성해보자.

`http://upcoming.yahooapis.com/services/rest/?method=metro.getForLatLon&api_key={api-key}&latitude=37.869111&longitude=-122.260634`

위 요청은 다음과 같은 결과 값을 반환한다.

## 사용자 인증을 요구하는 API

`http://upcoming.yahoo.com/services/api/token_auth.php`

### 토큰 획득하기

`http://upcoming.yahoo.com/services/auth/?api_key={api-key}`

`http://upcoming.yahooapis.com/services/rest/?method=auth.getToken&api_key={api-key}&frob={frob}`



## API를 사용하여 이벤트를 추가하기

<http://upcoming.yahoo.com/services/api/event.add.php>

- api\_key(필수)
- token(필수)
- name(필수)
- venue\_id(숫자, 필수)
- category\_id(숫자, 필수)
- start\_date(YYYY-MM-DD, 필수)
- end\_date(YYYY-MM-DD, 선택)
- start\_time(HH:MM:SS, 선택)
- end\_time(HH:MM:SS, 선택)
- description(선택)
- url(선택)
- personal(1=친구들에게만 보이기, 0=공개, 선택, 기본값은 0)
- selfpromotion(1=자기홍보, 0=정상, 선택, 기본값은 0)

<http://www.jcdl2008.org/>

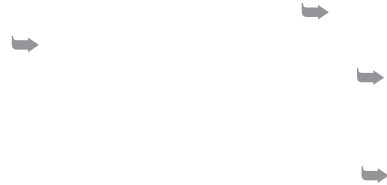
<http://upcoming.yahoo.com/event/add/>

[http://upcoming.yahooapis.com/services/rest/?api\\_key={api\\_key}&method=venue.search&search\\_text=Omni+William+Penn+Hotel&location=Pittsburgh%40C+PA](http://upcoming.yahooapis.com/services/rest/?api_key={api_key}&method=venue.search&search_text=Omni+William+Penn+Hotel&location=Pittsburgh%40C+PA)

[http://upcoming.yahooapis.com/services/rest/?api\\_key={api\\_key}&method=category.getList](http://upcoming.yahooapis.com/services/rest/?api_key={api_key}&method=category.getList)







```
curl -v -X POST -d "venue_id=56189&name=Joint+Conference+on+Digital+Libraries+%28JCDL%29+2008&end_date=2008-06-20&url=http%3A%2F%2Fwww.jcdl2008.org%2F&description{description}&start_date=2008-06-15&token=[TOKEN]&api_key=[API-KEY]&method=event.add&category_id=5" http://upcoming.yahooapis.com/services/rest/
```

<http://upcoming.yahoo.com/event/300826/>

**upcoming.yahoo.com 용 API 키**

[http://upcoming.yahoo.com/help/w/Language-specific\\_Libraries](http://upcoming.yahoo.com/help/w/Language-specific_Libraries)

<http://code.google.com/p/upcoming-python-api/>

svn checkout <http://upcoming-python-api.googlecode.com/svn/trunk/> upcoming-

python-api



**▼ CAUTION!** 이 글을 집필하는 시점에는 UPCOMING\_API 변수 값을 `http://upcoming.yahooapis.com/services/rest/` 로 설정하기 위해 `upcoming_api.py`를 수정해야만 했다.

## Eventful.com

<http://eventful.com/>

<http://api.eventful.com/>

<http://api.eventful.com/docs/>

<http://api.eventful.com/keys/>

<http://api.evdb.com/rest/{path for methods}>

<http://api.evdb.com/rest/events/search>

### 피드를 사용하여 이벤트 검색하기

[http://eventful.com/events?page\\_size=50&sort\\_order=Date&within=5&units=mi&q=berkeley%2C+ca&t=Future&c=](http://eventful.com/events?page_size=50&sort_order=Date&within=5&units=mi&q=berkeley%2C+ca&t=Future&c=)

[http://pipes.yahoo.com/pipes/pipe.info?\\_id=IJPPcvWA3BGvrWbY6kjTQA](http://pipes.yahoo.com/pipes/pipe.info?_id=IJPPcvWA3BGvrWbY6kjTQA)

## API를 사용하여 이벤트 검색하기

<http://api.eventful.com/docs/events/search>

[http://api.eventful.com/rest/events/search?app\\_key={api-key}&keywords=Bach&location=Berkeley%2C%20CA&within=5&units=5&page\\_size=50](http://api.eventful.com/rest/events/search?app_key={api-key}&keywords=Bach&location=Berkeley%2C%20CA&within=5&units=5&page_size=50)



<http://api.eventful.com/docs/events/ical>

[http://api.eventful.com/rest/events/ical?app\\_key={api-key}&keywords=Bach&location=Berkeley%2C%20CA&within=5&units=5&page\\_size=50](http://api.eventful.com/rest/events/ical?app_key={api-key}&keywords=Bach&location=Berkeley%2C%20CA&within=5&units=5&page_size=50)



## eventful.com을 위한 PHP API 키트

<http://api.eventful.com/>

[http://api.eventful.com/libs/Services\\_EVDB](http://api.eventful.com/libs/Services_EVDB)

<http://eventful.com/events/categories/technology?l=Berkeley%2C%20California%2C%20USA>

[http://api.evdb.com/rest/events/search?category=technology&location=Berkeley%2C%20California%2C%20USA&within=25&page\\_size=5&app\\_key={api-key}](http://api.evdb.com/rest/events/search?category=technology&location=Berkeley%2C%20California%2C%20USA&within=25&page_size=5&app_key={api-key})

<http://examples.mashupguide.net/ch15/evdb1.php?id=E0-001-004433237-3>

**eventful.com을 위한 파이썬 API 키트**

<http://api.eventful.com/libs/python/>

<http://api.eventful.com/libs/python/eventfulpy-0.3.tar.gz>

## 윈도우 버전의 파이썬에 simplejson과 httpLib2 설치하기

eventfulpy는 두 가지의 다른 라이브러리, simplejson([http://undefined.org/python/#simple\\_json](http://undefined.org/python/#simple_json))과 httpLib2(<http://bitworking.org/projects/httpLib2/>)에 의존성을 가지고 있다. 윈도우 버전의 파이썬 2.5를 사용하기 위해 simplejson를 설치하려면, 다음과 같은 작업을 해줘야 한다(기본 디렉터리로는 C:\Python25를 사용했다).

1. setuptools()를 설치한다. 가장 쉬운 방법은 .exe 인스톨러(예를 들어, setuptools-0.6c7.win32-py2.5.exe)를 실행하는 것이다.
2. 서브버전의 svn을 사용하여 <http://svn.red-bean.com/bob/simplejson/trunk/> 로 부터 simplejson을 체크아웃한다.
3. 비주얼 스튜디오가 설치되어 있지 않기 때문에 mingw32(<http://www.mingw.org/>)를 설치한다.
4. 다음 명령어를 사용하여 simplejson을 빌드한다.

```
c:\python25\python.exe setup.py build -c mingw32 -force
```

5. 빌드의 결과로 생성된 build\lib.win32-2.5\simplejson을 C:\WPython25\Lib\site-packages로 복사한다(표준 설치명령어- c:\Wpython25\python.exe setup.py install -가 작동하도록 하는 방법을 찾지 못해 수동으로 디렉터리를 복사했다).

아래에서 이 방법을 대체할 수 있는 다른 방법을 찾아볼 수 있다.

<http://maurus.net/weblog/2007/10/02/simplejson-17x-activestate-python-and-the-visual-studio-2003-compiler/>

httpLib2의 설치 좀 더 직관적이다. 예를 들면, 최신 배포판을 <http://code.google.com/p/httpLib2/downloads/list> 에서 다운로드해서 c:\python25\python.exe setup.py install 을 실행하면 된다.



## iCalendar로 프로그래밍하기

**\* NOTE!** hCalendar 마이크로포맷은 iCalendar와 동일한 정보를 표시하기 위해 설계되었지만 형태 측면에서 보면 hCalendar는 HTML과 RSS 내에 포함될 수 있다. hCalendar를 어떻게 사용하고 생성하는지 알아보려면 마이크로포맷을 다루는 18장을 읽어보도록 하라.

### 파이썬과 iCalendar

<http://codespeak.net/icalendar/>

<http://codespeak.net/icalendar/iCalendar-1.2.tgz>

D:\Document\Docs\2007\05\iCal-20070508-082112.ics

<http://vobject.skyhouseconsulting.com/usage.html>

## PHP와 iCalendar

<http://www.kigkonsult.se/iCalcreator/index.php>

---

25 <http://codespeak.net/icalendar/>, <http://codespeak.net/icalendar/example.html>, <http://codespeak.net/icalendar/small.html>,  
과 <http://codespeak.net/icalendar/groupscheduled.html>

<http://www.kigkonsult.se/iCalcreator/docs/using.html>



## 이벤트 캘린더를 iCalendar와 구글 캘린더로 내보내기

- iCalendar 피드
- 구글 캘린더

**소스 : UC 버클리 이벤트 캘린더**

<http://events.berkeley.edu/>

<http://events.berkeley.edu/documentation/user/rss.html>

- HTML

[http://events.berkeley.edu/?event\\_ID={event\\_ID}](http://events.berkeley.edu/?event_ID={event_ID})

- RSS 2.0

[http://events.berkeley.edu/index.php/rss/sn/pubaff/?event\\_ID={event\\_ID}](http://events.berkeley.edu/index.php/rss/sn/pubaff/?event_ID={event_ID})

- iCalendar

[http://events.berkeley.edu/index.php/ical/event\\_ID/{event\\_ID}/.ics](http://events.berkeley.edu/index.php/ical/event_ID/{event_ID}/.ics)

- live\_export XML

[http://events.berkeley.edu/index.php/live\\_export/sn/pubaff/?event\\_ID={event\\_ID}](http://events.berkeley.edu/index.php/live_export/sn/pubaff/?event_ID={event_ID})

[http://events.berkeley.edu/index.php/critics\\_choice\\_rss.html](http://events.berkeley.edu/index.php/critics_choice_rss.html)

**파이썬을 사용하여 ‘비평가의 선택’에서 iCalendar 피드 생성하기**

1. 아래의 피드에서 event\_ID를 파싱한다.

[http://events.berkeley.edu/index.php/critics\\_choice\\_rss.html](http://events.berkeley.edu/index.php/critics_choice_rss.html)

2. 각각의 iCalendar 항목을 읽어서 '비평가의 선택' 피드를 iCalendar로 통합하기 위한 항목에 추가한다.



[http://examples.mashupguide.net/ch15/critics\\_choice.ics](http://examples.mashupguide.net/ch15/critics_choice.ics)

**구글 캘린더에 이벤트 추가하기**

[n7irauk3nns30fuku1anh43j5s@group.calendar.google.com](mailto:n7irauk3nns30fuku1anh43j5s@group.calendar.google.com)

<http://www.google.com/calendar/embed?src=n7irauk3nns30fuku1anh43j5s@group.calendar.google.com>

- [http://code.google.com/apis/calendar/developers\\_guide\\_php.html#CreatingRecurring](http://code.google.com/apis/calendar/developers_guide_php.html#CreatingRecurring)
- <http://code.google.com/apis/gdata/elements.html#gdRecurrence>
  
- <http://www.w3.org/2002/12/cal/rfc2445#sec4.3.10> (RECUR에 대한 4.3.10 섹션)
- <http://www.w3.org/2002/12/cal/rfc2445#sec4.8.5.1> (EXDATE[exception dates/times]에 대한 4.8.5.1 섹션)
- <http://www.w3.org/2002/12/cal/rfc2445#sec4.8.5.4> (Recurrence Rule에 대한 4.8.5.4 섹션)















요약

■

■

■

■

■

■

■

■

■

# 16

---

## 온라인 스토리지 서비스 사용하기

---

- 아마존 S3(<http://aws.amazon.com/s3>)
- 자체 API를 제공하는 box.net(<http://box.net/>)<sup>1</sup>
- 자체 API를 제공하는 MediaMax(<http://www.mediamax.com/>)<sup>2</sup>
- 자체 API를 제공하는 Omnidrive(<http://www.omnidrive.com/>)<sup>3</sup>

---

1 <http://enabled.box.net/>

2 <http://www.mediamax.com/webservices/>

3 <http://dev.omnidrive.com/HomePage>

## 아마존 S3에 대한 소개

### S3를 사용하는 근본적인 이유

- S3는 다른 솔루션에 비해 가격이 저렴하다. 예를 들어, 플리커와 유사한 사진 호스팅 서비스인 스머그머그(SmugMug)의 돈 맥어스킬(Don MacAskill)은 동일한 양의 스토리지를 구매하고 유지하는 대신에 S3를 사용함으로써 2006년에 약 \$692,000를 절감했

---

4 <http://aws.amazon.com/s3>에 접속하면 <http://www.amazon.com/gp/browse.html?node=l6427261>로 리다이렉트 된다.

다고 추정하고 있다<sup>5</sup>(이 비교 데이터는 아마존의 새로운 가격모델 도입과 함께 변경되었을 수도 있다<sup>6</sup>).

- S3는 높은 확장성을 약속하므로 사용자들은 필요한 스토리지의 용량을 자유롭게 변경할 수 있다. 사용한 만큼에 대해서만 지불하면 되고, 이것은 필요할 것이라고 예상하는 최대 스토리지의 양을 구매하기 위해 사전에 돈을 지불해야 할 필요가 없다는 것을 의미한다. 이렇게 실용적인 모델은 상대적으로 영세한 개인이 웹2.0 애플리케이션을 개발할 수 있는 수준으로 진입장벽을 낮춰준다.
- 아마존이 자신들의 인프라를 S3에서 운영한다고 주장하고 있으므로 견고함 역시 제공된다. 따라서 아마존 자체와 같은 신뢰도를 기대할 수 있다.<sup>7</sup>
- <http://jeremy.zawodny.com/blog/archives/007641.html> 에서 S3를 사용하는 백업 서비스와 툴의 목록을 찾아볼 수 있다.
- <http://solutions.amazonwebservices.com/connect/kbcategory.jspa?categoryID=66> 에서 개발자들이 이미 개발해놓은 솔루션들을 찾아볼 수 있다.

## 아마존 S3의 개념적인 구조

---

5 <http://blogs.smugmug.com/don/files/ETech-SmugMug-Amazon-2007.pdf>

6 <http://blogs.smugmug.com/don/2007/05/01/amazon-s3-new-pricing-model/>

7 [http://www.amazon.com/S3-FAQs-AWS-home-page/b/ref=sc\\_fe\\_c\\_0\\_16427261\\_9?&node=16427271&no=16427261#as9](http://www.amazon.com/S3-FAQs-AWS-home-page/b/ref=sc_fe_c_0_16427261_9?&node=16427271&no=16427261#as9)

■

■

■

■

■

■

■

■

- 
- 8 <http://docs.amazonwebservices.com/AmazonS3/2006-03-01/UsingBucket.html>
- 9 <http://docs.amazonwebservices.com/AmazonS3/2006-03-01/UsingObjects.html>
- 10 <http://docs.amazonwebservices.com/AmazonS3/2006-03-01/ListingKeysHierarchy.html>
- 11 <http://docs.amazonwebservices.com/AmazonS3/2006-03-01/UsingMetadata.html>
- 12 <http://docs.amazonwebservices.com/AmazonS3/2006-03-01/UsingAccessControl.html>
- 13 <http://docs.amazonwebservices.com/AmazonS3/2006-03-01/S3TorrentRetrieve.html>
- 14 <http://docs.amazonwebservices.com/AmazonS3/2006-03-01/VirtualHosting.html>



`http://s3.amazonaws.com/{bucket}/{key}`

`http://{bucket}.s3.amazonaws.com/{key}`

`http://s3.amazonaws.com/raymondye/858Xtoc____.pdf`

`http://raymondye.s3.amazonaws.com/858Xtoc____.pdf`

#### 아마존 AWS에 계정 생성하기

S3를 사용하기 위해서는 아마존 웹서비스(AWS)의 키와 시크릿(secret) 키가 필요하다. AWS에 대한 홈페이지는 아래에 위치하고 있다.

`http://aws.amazon.com`

AWS의 이커머스 서비스에 대한 문서는 아래에서 찾을 수 있다.

`http://www.amazon.com/gp/browse.html?node=12738641`

키를 발급받으려면 AWS 계정을 생성해야만 한다.

`http://www.amazon.com/gp/aws/registration/registration-form.html`

이미 회원이라면 아래에 접속하면 키를 얻을 수 있다.

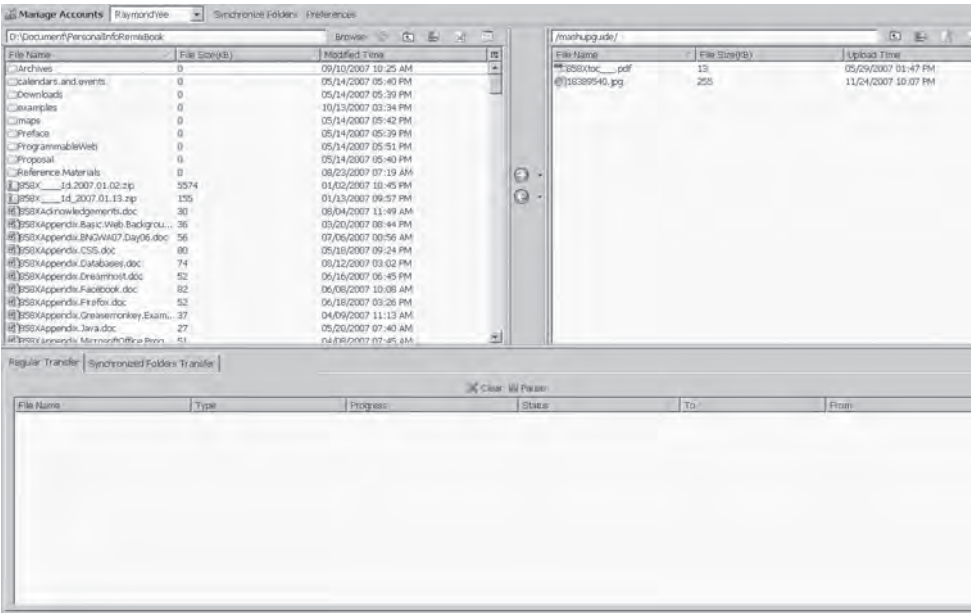
`http://aws- ➡`

`portal.amazon.com/gp/aws/developer/account/index.html?ie=UTF8&action=access-key`

키와 시크릿 키를 발급받고 나면, 또한 X.509 인증서를 사용할 수도 있다.

## 파이어폭스 S3 확장기능을 사용하여 S3 시작하기

<http://www.rjonna.com/ext/s3fox.php> (<http://www.s3fox.net/>으로 변경되었다)



## S3의 REST 인터페이스 사용하기

<http://docs.amazonwebservices.com/AmazonS3/2006-03-01/RESTAPI.html>

- 사용자 인증의 처리방법, 보다 엄밀하게 말하자면 서명값의 연산방법<sup>15</sup>
- 권한인증을 처리하기 위한 권한 제어 목록의 사용<sup>16</sup>
- 메타데이터의 구현방법

<http://host.s3.amazonaws.com>

- **AWSSecretAccessKey**: 접근 키 값
- **Expires**: 협정 세계시각(UTC, Universal Time Coordinated) 기준으로 1970년 1월 1일이후부터의 시간을 초단위 숫자로 명시한 값으로 언제 서명이 만료되는지를 나타내는 값
- **Signature**: **StringToSign**(뒷부분에서 설명하고 있다)을 HMAC-SHA1으로 암호화하고 이 값을 Base64로 인코딩한 후 다시 URL 인코딩한 값.

---

<sup>15</sup> <http://docs.amazonwebservices.com/AmazonS3/2006-03-01/RESTAuthentication.html>

<sup>16</sup> <http://docs.amazonwebservices.com/AmazonS3/2006-03-01/RESTAccessPolicy.html>

<http://docs.amazonwebservices.com/AmazonS3/2006-03-01/RESTAuthentication.html>





Signature=rucSbH0yNEcP9oM2XNlouVI3BH4%3D

- Crypt\_HMAC<sup>17</sup>
- HTTP\_Request<sup>18</sup>

`http://johnsmith.s3.amazonaws.com/photos/puppy.jpg?AWSAccessKeyId=0PN5J17HBG  
ZHT7JJ3X82&Signature=rucSbH0yNEcP9oM2XNlouVI3BH4%3D&Expires=1175139620`

## REST 인터페이스를 사용하여 버킷 목록 가져오기

---

<sup>17</sup> [http://pear.php.net/package/Crypt\\_HMAC](http://pear.php.net/package/Crypt_HMAC)

<sup>18</sup> [http://pear.php.net/package/HTTP\\_Request](http://pear.php.net/package/HTTP_Request)



`http://s3.amazonaws.com?AWSAccessKeyId={AWSAccessKeyId}&Expires=1196114919&Signature={Signature}`

## S3의 SOAP 인터페이스 사용하기

- REST 인터페이스의 경우와 마찬가지로, AWS에 계정을 등록하면 얻을 수 있는 아마존의 AWS 키와 시크릿 키를 필요로 한다(7장을 참고하거나 좀 더 자세한 내용을 보고 싶다면 ‘아마존 AWS에 계정생성하기’ 란을 참고하자).
- 비록 S3가 REST와 SOAP 인터페이스를 통해 접근할 수 있지만, 이 코드는 SOAP과 WSDL을 사용한다. 이 글을 집필하는 시점에서 서비스를 위한 WSDL은 <http://s3.amazonaws.com/doc/2006-03-01/AmazonS3.wSDL>에 위치하고 있다. 가장 최근의 WSDL에 대한 URL은 <http://aws.amazon.com/s3>에서 조회할 수 있다.
- 예제 코드에서는 ListMyBuckets와 CreateBuckets 등 두 개의 메서드가 사용된다. 모든 메서드의 목록을 <http://www.awszone.com/scratchpads/aws/s3.us/index.aws>에서 찾아볼 수 있다. (기술적인 문서는 <http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=48>에 위치하고 있으며 이 페이지에서



docs.amazonwebservices.com/AmazonS3/2006-03-01/를 찾을 수 있다.)

- 복잡한 내용 중의 하나가 이전 절에서 배운 서명값 계산이라는 것에 주의하라.

## 아마존의 S3 API 키트

### PHP

- `php-aws`<sup>19</sup>
- Neurofuzzy.net에서 제공하는 `s3.class.zip`. 자주 사용되는 클래스를 구현한 것처럼 보인다.<sup>20</sup>
- edoceo의 `php3tk`<sup>21</sup>

<http://php-aws.googlecode.com/svn/trunk/class.s3.php>

<http://code.google.com/p/php-aws/wiki/S3Class>

<http://sitening.com/blog/introducing-php-aws/>

- 이 라이브러리가 `curl`을 사용한다는 것은 이 라이브러리가 보다 용량이 큰 파일들을 처

---

<sup>19</sup> <http://code.google.com/p/php-aws/> and specifically <http://php-aws.googlecode.com/svn/trunk/class.s3.php>

<sup>20</sup> <http://neurofuzzy.net/2006/08/26/amazon-s3-php-class-update/> and <http://www.neurofuzzy.net/wp-content/2006/03/s3.class.zip>

<sup>21</sup> <http://www.edoceo.com/creo/phps3tk/>

리할 수 있도록 개발되었다는 것을 의미한다.

- 공개된 리소스를 읽거나 비공개 접근제어목록(Access Control List, ACL)을 제어하는 기능만이 현재 구현되어 있다.
- 오브젝트의 사용자 메타데이터에 대해서는 구현되어 있지 않다.

1. <http://php-aws.googlecode.com/svn/trunk/class.s3.php>를 원하는 로컬 PHP 디렉터리에 다운로드하라(내 경우에는 /home/rdhyee/phplib/php-aws/class.s3.php에 다운로드하였다).
2. 다음 코드를 실행시켜 보자(이 코드는 먼저 S3 버킷 목록을 가져오는 작업을 수행하고 만약 버킷이 존재하지 않는다면 mashupguidetest라는 이름으로 버킷을 생성한다).

**\* NOTE!** php-aws를 사용하려면, curl이 커멘드 라인에서 실행될 수 있도록 설치가 되어 있어야 한다. 또한 class.s3.php에 \$\_pathToCurl 파라미터가 설정되어 있어야 php-aws가 curl을 찾을 수 있다.

## 파이썬

- Boto<sup>22</sup>
- HanzoiArchive의 S3 tools<sup>23</sup>
- BitBucket<sup>24</sup>

[http://boto.googlecode.com/svn/trunk/doc/s3\\_tut.txt](http://boto.googlecode.com/svn/trunk/doc/s3_tut.txt)

- S3 버킷 목록을 가져와서 버킷의 이름, 생성일자를 출력하고 접근제어목록(ACL)을 XML 형태로 화면에 표시한다.
- 수정된 최종시간을 표시하는 타임스탬프, 오브젝트의 메타데이터와 함께 특정 버킷에 포함된 오브젝트의 목록을 읽는다.
- 버킷으로 파일을 업로드하고 새롭게 업로드된 파일의 메타데이터를 다시 읽어온다.

---

22 <http://code.google.com/p/boto/>

23 <http://www.hanzoarchives.com/development-projects/s3-tools/>

24 <http://cheeseshop.python.org/pypi/BitBucket/0.4a>

요약

# 17

---

## 데스크톱과 웹 기반 오피스 스위트 매쉬업하기

---

■

---

1 (오피스) 사무 등에 흔히 쓰이는 엑셀이나 워드 프로그램 등을 총칭하는 의미이다.

- ODF와 Open XML을 사용하여 어떻게 간단한 문서를 생성하는지 설명할 것이다.
- 오픈오피스와 마이크로소프트 오피스를 사용하는 간단한 스크립팅에 대해 알아볼 것이다.
- 공개된 문서 포맷을 사용하여 어떤 것들이 가능한지 개략적으로 살펴볼 것이다.
- 어떻게 구글 스프레드시트를 프로그래밍하고 어떻게 아마존 이커머스 서비스와 같은 다른 API와 매쉬업할 수 있는지 보여줄 것이다.

## 오피스 스위트의 매쉬업 시나리오

파워포인트의 재사용: 그림, 약도, 슬라이드와 같은 공통 디지털 자산을 재활용하여 마이크로소프트 파워포인트 파일들을 작성할 수 있는가? 개인정보 관리 시스템을 개발하여 개인이 작성한 PPT로부터 슬라이드를 추출하고 이렇게 추출된 슬라이드를 동적으로 자산화할 수 있는가?

한번 작성하고 어느 곳으로든 출판한다: 현재 마이크로소프트 오피스 2007에서 작성 중인 이 책을 (X)HTML, Docbook(<http://www.docbook.org/whatis> 를 참고하라), PDF, 그리고 위키 마크업으로 재출판하고 싶다. 어떻게 마이크로소프트 워드로 작성된 원고를 이러한 포맷들로 변경할 수 있을까?

데이터 변환하기: 정적인 데이터뿐만 아니라 동적인 시뮬레이션 데이터가 스프레드시트 안으로 다운로드되어 교육 참고자료로 사용될 수 있도록 구현된 교육사이트를 개발할 수도 있을 것이다. 세상에는 방대한 양의 데이터가 존재한다. 이러한 데이터를 많은 사람이 사용하는 데이터 분석 툴인 스프레드시트(데스크톱에서 실행되던 클라우드 플랫폼에서 실행되던)에서 사용 가능한 형태로 변환하는 프로그램을 작성할 수 있는가?

플리커를 사용하여 즉석 파워포인트 프레젠테이션 생성하기: 나는 플리커의 사진 세트를 파워포인트 파일로 다운로드하기를 원한다. 이 시나리오는 파워포인트가 시장 지배적인 프레젠테이션 프로그램인 현재 상황에 적당할 듯하다. 에드워드 터프트(Edward Tufte, [http://en.wikipedia.org/wiki/Edward\\_Tufte](http://en.wikipedia.org/wiki/Edward_Tufte), 예일대 교수로 그의 저서에서 파워포인트의 단점에 대해서 비판하였다)가 비록 싫어할지라도, 플리커와 PPT 간의 변환기는 여러분의 휴가사진을 다음 번 회사 프레젠테이션에서 발표하기 쉽게 만들어줄 것이다.

## 문서 마크업의 세계

- [http://en.wikipedia.org/wiki/Document\\_markup\\_language](http://en.wikipedia.org/wiki/Document_markup_language)
- [http://en.wikipedia.org/wiki/List\\_of\\_document\\_markup\\_languages](http://en.wikipedia.org/wiki/List_of_document_markup_languages)
- [http://en.wikipedia.org/wiki/Comparison\\_of\\_document\\_markup\\_languages](http://en.wikipedia.org/wiki/Comparison_of_document_markup_languages)

# OpenDocument 포맷

**\* NOTE!** 이번 절에서는, 오픈오피스 버전 2.2(번역시점의 최신 버전은 3.1)를 설치하였다고 가정하고 진행한다.

1. 오픈오피스 라이터(Writer)를 실행하고 **Hello World**를 입력한 뒤 helloworld.odt란 이름으로 저장하도록 하라.<sup>5</sup>
2. 파일을 ZIP 유틸리티(WinZip와 같은)를 사용하여 열도록 하자. 파일의 확장자를 .odt에서 .zip으로 변경하여 운영체제가 이 파일을 ZIP 파일로 인식하도록 해주면 이 작업을 좀 더 편하게 할 수 있다. 파일의 압축을 해제해보면 실제로 ZIP 포맷의 파일이라는 것을 볼 수 있을 것이다(그림 17-1의 파일 목록을 보도록 하라).

---

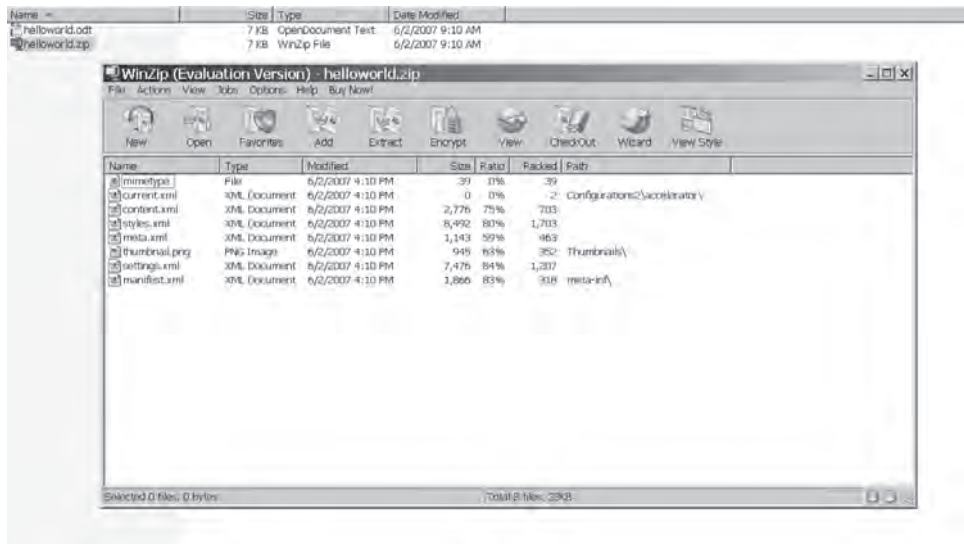
2 <http://www.oasis-open.org/committees/office/>

3 [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=43485](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43485)

4 [http://books.evc-cit.info/OD\\_Essentials.pdf](http://books.evc-cit.info/OD_Essentials.pdf) 또는 <http://develop.opendocumentfellowship.com/book/>

5 <http://examples.mashupguide.net/ch17/helloworld.odt>





- content.xml
- styles.xml
- meta.xml
- settings.xml
- META-INF/manifest.xml
- mimetypes
- Configuration2/accelerator/
- Thumbnails/thumbnail.png

---

---

---

---

- ‘명세서의 1장부터 16장까지의 내용에서 발췌한’ 오피스 문서를 위한 스키마 - 버전 1.0<sup>7</sup>
- ‘OpenDocument 패키지 포맷에 사용되는 메니페스트(manifest) 파일을 위한 표준 스키마’ - 버전 1.0<sup>8</sup>
- ‘이 명세서에 포함된 메타 정보와 포매팅 프로퍼티만을 허용하는 오피스 문서를 위한 엄격한 스키마’ - 버전 1.0<sup>9</sup>

실제로 필수적인 파일은 단지 **content.xml**과 **META-INF/manifest.xml** 파일뿐이다. 만일 워드프로세서 엘리먼트를 **zip** 포맷으로 포함하고 메니페스트 파일이 이 파일을 가리키도록 생성한다면, 오픈오피스는 이 파일을 문제 없이 열 수 있을 것이다. 이 결과로 생성된 문서는 서식이 적용되지 않는 일반 텍스트만 사용된 문서일

---

7 <http://www.oasis-open.org/committees/download.php/12571/OpenDocument-schema-v1.0-os.rng>

8 <http://www.oasis-open.org/committees/download.php/12570/OpenDocument-manifest-schema-v1.0-os.rng>

9 <http://www.oasis-open.org/committees/download.php/12569/OpenDocument-strict-schema-v1.0-os.rng>

것이다. 여러분은 누가 파일을 생성하였는지 또는 언제 마지막으로 편집되었는지 등 어떠한 메타 정보도 알 수 없을 것이며 프린터 설정, 표시 영역, 그리고 확대수준 등은 오픈오피스의 기본값으로 설정될 것이다.

---

10 [http://examples.mashupguide.net/ch17/helloworld\\_min\\_odt\\_1.odt](http://examples.mashupguide.net/ch17/helloworld_min_odt_1.odt)

11 <http://opendocumentfellowship.com/odfviewer>

**\* NOTE!** OpenDocumentValidator<sup>12</sup>를 다운로드해서 설치하거나 온라인 버전을 사용하면 된다.<sup>13</sup>

---

---

[http://examples.mashupguide.net/ch17/helloworld\\_min\\_odt\\_2.odt](http://examples.mashupguide.net/ch17/helloworld_min_odt_2.odt)

---

<sup>12</sup> <http://opendocumentfellowship.com/projects/odftools>

<sup>13</sup> <http://opendocumentfellowship.com/validator>

<sup>14</sup> <http://testsuite.opendocumentfellowship.com/testcases/General/DocumentStructure/SingleDocumentContents/testDoc/testDoc.odt> via <http://testsuite.opendocumentfellowship.com/testcases/General/DocumentStructure/SingleDocumentContents/TestCase.html>







## 기본적인 ODF 태그에 대해 배워보자

[http://examples.mashupguide.net/ch17/odt\\_example\\_4.odt](http://examples.mashupguide.net/ch17/odt_example_4.odt)

[http://examples.mashupguide.net/ch17/odt\\_example\\_4/](http://examples.mashupguide.net/ch17/odt_example_4/)

## Purpose (Heading 1)

The following sections illustrate various possibilities in ODF Text.

### ***A simple series of paragraphs (Heading 2)***

This section contains a series of paragraphs.

This is a second paragraph.

And a third paragraph.

### ***A section with lists (Heading 2)***

Elements to illustrate:

- hyperlinks
- italics and bold text
- lists (ordered and unordered)

How to figure out ODF

1. work out the content.xml tags
2. work styles into the mix
3. figure out how to apply what we learned to spreadsheets and presentations

The *URL* for Flickr is <http://www.flickr.com>. The **API** page is <http://www.flickr.com/services/api/>

### **A Table (Heading 1)**

Website	Description	URL
<u>Flickr</u>	A social photo sharing site	<u><a href="http://www.flickr.com">http://www.flickr.com</a></u>
Google Maps	An <u>online</u> map	<u><a href="http://maps.google.com">http://maps.google.com</a></u>

### **Footnotes (Heading 1)**

This sentence has an accompanying footnote.<sup>1</sup> Where does the text after a footnote go?

### **An Image**

- 레벨1과 레벨2의 제목
- 몇 개의 문단

- 정렬된 목록과 정렬되지 않은 목록
- 이탤릭과 볼드체, 그리고 변경된 폰트로 표기된 텍스트
- 테이블
- 이미지

1. ODF 요소에 어떠한 서식도 적용하지 않은 ODF 텍스트 문서를 생성하기
2. 텍스트 문단에 서식을 적용하기
3. 정렬된 목록과 정렬되지 않은 목록을 구분하기 위한 서식을 목록에 적용하기
4. 텍스트에 볼드체, 이탤릭체, 폰트 변경 그리고 색상 변경을 적용하기

## ODF 요소에 어떠한 서식도 적용하지 않은 ODF 텍스트 문서 생성하기

[http://examples.mashupguide.net/ch17/odt\\_example\\_1.odt](http://examples.mashupguide.net/ch17/odt_example_1.odt)

[http://examples.mashupguide.net/ch17/odt\\_example\\_1/](http://examples.mashupguide.net/ch17/odt_example_1/)

[http://examples.mashupguide.net/ch17/odt\\_example\\_1/content.xml](http://examples.mashupguide.net/ch17/odt_example_1/content.xml)

헤더와 단락

목록

텍스트 스타일의 일괄적용을 위해 text:a와 text:span 사용하기

테이블

각주

포함된 이미지



## 문단의 텍스트를 text-body로 설정하기

---

15 [http://examples.mashupguide.net/ch17/odt\\_example\\_1/META-INF/manifest.xml](http://examples.mashupguide.net/ch17/odt_example_1/META-INF/manifest.xml)

16 [http://examples.mashupguide.net/ch17/odt\\_example\\_2/styles.xml](http://examples.mashupguide.net/ch17/odt_example_2/styles.xml)

17 [http://examples.mashupguide.net/ch17/odt\\_example\\_2/content.xml](http://examples.mashupguide.net/ch17/odt_example_2/content.xml)

[http://examples.mashupguide.net/ch17/odt\\_example\\_2.odt](http://examples.mashupguide.net/ch17/odt_example_2.odt)

[http://examples.mashupguide.net/ch17/odt\\_example\\_2/](http://examples.mashupguide.net/ch17/odt_example_2/)

**정렬된 목록과 정렬되지 않은 목록을 구분하기 위해 포매팅하기**



## <text:span> 엘리먼트에 볼드체, 이탤릭체, 폰트 변경, 그리고 색상 변경 적용하기

- Arial 스타일을 선언하는 <style:font-face>를 포함하는 <office:font-face-decls> 엘리먼트를 추가하라.
- 스타일이 적용될 세 개의 <text:span> 엘리먼트의 스타일을 표현하는 각각 T1, T2, T5 라는 이름의 <style:style> 엘리먼트를 생성하라.
- T1, T2 그리고 T5 스타일을 text:style-name 어트리뷰트를 사용하여 <text:span> 엘리먼트와 연결시켜라.



[http://examples.mashupguide.net/ch17/odt\\_example\\_4.odt](http://examples.mashupguide.net/ch17/odt_example_4.odt)

## ODF를 다루는 API 키트

[http://en.wikipedia.org/wiki/OpenDocument\\_software](http://en.wikipedia.org/wiki/OpenDocument_software)

<http://opendocumentfellowship.com/development/tools>

- Odfpy(<http://opendocumentfellowship.com/projects/odfpy>)에 대해 알아볼 것이다. Odfpy가 제공하는 문서에 따르면, Odfpy는 파이썬에서 사용할 수 있으며 OpenDocument를 완벽하게 지원하는 API가 되는 것을 목표로 하고 있다. 보다 편리한 다른 API들과는 다르게, 본질적으로 Odfpy는 단지 XML 포맷 상위에 존재하는 추상 레이어일뿐이다. 이 API는 프로그래머들이 유효하지 않은 문서를 생성하지 않도록 하는 것에 초점을 맞추고 있다. 이 API는 프로그래머가 유효하지 않은 엘리먼트를 추가하거나, 문법상 알려져 있지 않은 어트리뷰트를 추가하거나, 필수적인 엘리먼트를 추가하지 않거나, 허용되지 않는 텍스트를 엘리먼트에 추가하면 예외를 발생시킨다.
- 아직 개발의 초기단계에 있는 OpenDocumentPHP(<http://opendocumentphp.org/>)에 대해 설명할 것이다.

## Odfpy

<http://opendocumentfellowship.com/files/api-for-odfpy.odt>

`svn export http://opendocumentfellowship.com/repos/odfpy/trunk odfpy`

---

---

예제 ODF 텍스트 문서를 재생성하기





















[http://examples.mashupguide.net/ch17/odfpy\\_gen\\_example.odt](http://examples.mashupguide.net/ch17/odfpy_gen_example.odt)

## **OpenDocumentPHP**

<http://downloads.sourceforge.net/opendocumentphp/OpenDocumentPHP-0.5.2.zip>

<http://opendocumentphp.org/index.php/home/11-new-pear-server-for-opendocumentphp>

<http://opendocumentphp.org/static/apidoc/svn/>



**\* NOTE!** PHP에서 OpenDocumentPHP를 구동하기 위해서는 ZipArchive 기능을 활성화시켜줘야 한다. 리눅스 시스템에서는 `-enable-zip` 옵션을 사용하여 PHP를 컴파일하도록 하고 윈도우 시스템에서는 `php.ini` 파일 내에 `php_zip.dll`을 활성화시키도록 하라.





[http://examples.mashupguide.net/ch17/odp\\_gen\\_example.odt](http://examples.mashupguide.net/ch17/odp_gen_example.odt)

## 오픈오피스를 사용하여 ODF 생성하기

- JODConverter 자바 라이브러리(<http://www.artofsolving.com/opensource/jodconverter>)
- 오픈오피스를 사용하는 펄과 파이썬 라이브러리인 OOoLib(<http://sourceforge.net/projects/oolib/>).



## ECMA 표준인 Office Open XML(OOXML)

[http://en.wikipedia.org/wiki/Office\\_Open\\_XML](http://en.wikipedia.org/wiki/Office_Open_XML)

<http://www.ecma-international.org/publications/standards/Ecma-376.htm>

[http://www.ecma-international.org/news/TC45\\_current\\_work/OpenXML%20White%20Paper.pdf](http://www.ecma-international.org/news/TC45_current_work/OpenXML%20White%20Paper.pdf)

- '새로운 오피스 2007 파일 포맷에 대해 반드시 알아야 할 5가지 장점'이라는 글이 아마 도움이 될 것이다(<http://www.devx.com/MicrosoftISV/Article/30907/2046>).
- <http://openxmldeveloper.org/default.aspx> 는 이 주제에 관련한 튜토리얼을 제공한다.

---

19 <http://openxmldeveloper.org/articles/LearningOnline.aspx>

20 [http://examples.mashupguide.net/ch17/helloworld\\_onedoc\\_2003.xml](http://examples.mashupguide.net/ch17/helloworld_onedoc_2003.xml)

<http://blogs.msdn.com/dmahugh/archive/2006/06/27/649007.aspx>

<http://schemas.openxmlformats.org/wordprocessingml/2006/3/main>

<http://schemas.openxmlformats.org/wordprocessingml/2006/main>

---

---



- 'ECMA Office Open XML 포맷 가이드'는 OOXML을 고수준의 개념과 마케팅 관점에서 개략적으로 정리한 공식문서이다.
- <http://openxmldeveloper.org/articles/directory.aspx> 는 OOXML 커뮤니티에 의해 수집된 튜토리얼 목록을 제공한다.
- <http://openxmldeveloper.org/articles/OpenXMLsamples.aspx>는 예제 OOXML 문서를 제공한다.
- <http://msdn2.microsoft.com/en-us/library/bb187361.aspx>는 마이크로소프트 오피스 2007의 오브젝트 모델에 대한 내용을 다룬다.

- [http://en.wikipedia.org/wiki/User:Flemingr/Microsoft\\_Office\\_2003\\_XML\\_formats](http://en.wikipedia.org/wiki/User:Flemingr/Microsoft_Office_2003_XML_formats) 는 계보상 OOXML과 유사한 이전 버전인 Office 2003 XML 포맷에 대한 내용을 정리하고 있다. 하지만 이해하기가 쉽지 않다.
- 마이크로소프트의 브라이언 존스(Brian Jones)가 OOXML로 스프레드시트를 생성하는 방법에 대해 명쾌한 튜토리얼을 작성하였으며 [http://blogs.msdn.com/brian\\_jones/archive/2007/05/29/simple-spreadsheetml-file-part-3-formatting.aspx](http://blogs.msdn.com/brian_jones/archive/2007/05/29/simple-spreadsheetml-file-part-3-formatting.aspx) 에서 찾아볼 수 있다.

## OOXML 뷰어와 검증기

- Open XML 패키지를 브라우징, 수정할 수 있고 ECMA의 최종 스키마와 비교하여 검증할 수 있게 해주는 Open XML 패키지 익스플로러(<http://www.codeplex.com/PackageExplorer>)
- 마이크로소프트 오피스 XP와 2003을 사용한다면 워드2007, 엑셀2007, 그리고 파워포인트 2007 용 마이크로소프트 오피스 호환 팩을 다운로드 받아서 OOXML을 읽고 작성할 수 있다.<sup>21</sup> 또 이 호환 팩은 워드2007과 엑셀2007파일을 읽을 수 있는 마이크로소프트 오피스 워드 뷰어 2003과 마이크로소프트 오피스 엑셀 뷰어 2003을 무료로 사용할 수 있게 해준다.<sup>22</sup>

## ODF와 OOXML의 비교

21 <http://www.microsoft.com/downloads/details.aspx?FamilyId=941b3470-3ae9-4aee-8f43-c6bb74cd1466&displaylang=en>

22 <http://support.microsoft.com/kb/925180>

- [http://en.wikipedia.org/wiki/Comparison\\_of\\_OpenDocument\\_and\\_Office\\_Open\\_XML\\_formats](http://en.wikipedia.org/wiki/Comparison_of_OpenDocument_and_Office_Open_XML_formats) 은 이 포맷들을 비교한다.
- [http://weblog.infoworld.com/realitycheck/archives/2007/05/odf\\_vs\\_openxml.html](http://weblog.infoworld.com/realitycheck/archives/2007/05/odf_vs_openxml.html) 은 정치, 경제, 마케팅 측면 및 기술과 관련된 이슈들을 정리하여 설명한다.

## 온라인 오피스 스위트

[http://en.wikipedia.org/wiki/List\\_of\\_online\\_spreadsheets](http://en.wikipedia.org/wiki/List_of_online_spreadsheets)

<http://code.google.com/apis/spreadsheets/overview.html>

## 프로그래밍 가능한 온라인 스프레드시트의 사용 시나리오

- 몸무게, 재정상태 또는 시간을 추적하고 이 정보를 가족이나 친구와 공유한다.
- 봇Bot이 스프레드시트에 추가한 데이터를 연산하도록 한 뒤 이것을 분석한다. 예를 들어, 주식 포트폴리오를 추적하고 싶다면 StrikeIron 유료 실시간 주식 시세정보 서비스를 사용하여 포트폴리오의 가치를 계산할 수 있다(포트폴리오 정보를 온라인에 저장하는 점이 고민될 수 있겠지만 원칙적으로는 가능한 시나리오이다).
- 성적을 조회하고 배포하는 애플리케이션을 개발한다.
- 결혼 정보 데이터베이스를 관리한다.



- API를 사용하여 읽고 업데이트할 수 있는 프로젝트 관리 툴을 개발한다.
- 스프레드시트에 딜리셔스 북마크 목록을 백업한다.
- 도서관의 책을 추적한다.
- 온라인 차트를 생성한다(<http://imagine-it.org/google/spreadsheets/makechart.html>을 보라).

## 구글 스프레드시트 API

[http://code.google.com/apis/spreadsheets/developers\\_guide\\_protocol.html](http://code.google.com/apis/spreadsheets/developers_guide_protocol.html)

## 파이썬 API 키트

svn checkout <http://gdata-python-client.googlecode.com/svn/trunk/gdata-python-client>

[http://code.google.com/apis/spreadsheets/developers\\_guide\\_python.html](http://code.google.com/apis/spreadsheets/developers_guide_python.html)



---

---

<http://spreadsheets.google.com/feeds/spreadsheets/private/full/{GSheetID}>

<http://spreadsheets.google.com/feeds/spreadsheets/private/full/o0634173711186572809.3585145106901556666>

<http://spreadsheets.google.com/feeds/worksheets/{GSheetID}/private/full/{worksheetID}>

<http://spreadsheets.google.com/feeds/worksheets/o06341737111865728099.3585145106901556666/private/full/od6>



---

---

---

---

---

---



**\* NOTE!** 구글 스프레드시트 API는 활발하게 개발중이며 여전히 안정화의 과정에 있다

## 아마존 위시리스트와 구글 스프레드시트의 매쉬업

### 아마존 ECS 웹서비스를 통해 위시리스트에 접근하기

<http://www.awszone.com/scratchpads/aws/ecs.us/ListLookup.aws>

<http://www.amazon.com/gp/registry/wishlist/{ListID}/>

[http://ecs.amazonaws.com/onca/xml?Service=AWSECommerceService&Version=2007-05-14&AWSAccessKeyId=\[YourAccessKeyId\]&Operation=ListLookup&ListType=WishList&ListId=1U5EXVPVS3WP5](http://ecs.amazonaws.com/onca/xml?Service=AWSECommerceService&Version=2007-05-14&AWSAccessKeyId=[YourAccessKeyId]&Operation=ListLookup&ListType=WishList&ListId=1U5EXVPVS3WP5)  
[http://ecs.amazonaws.com/onca/xml?Service=AWSECommerceService&Version=2007-10-29&AWSAccessKeyId=\[YourAccessKeyId\]&Operation=ListLookup&ListType=WishList&ListId=1U5EXVPVS3WP5](http://ecs.amazonaws.com/onca/xml?Service=AWSECommerceService&Version=2007-10-29&AWSAccessKeyId=[YourAccessKeyId]&Operation=ListLookup&ListType=WishList&ListId=1U5EXVPVS3WP5)

[http://ecs.amazonaws.com/onca/xml?Service=AWSECommerceService&Version=2007-10-29&AWSAccessKeyId=\[YourAccessKeyId\]&Operation=ListLookup&ListType=WishList&ListId=1U5EXVPVS3WP5&ResponseGroup=ListItems,Medium&ProductPage=2](http://ecs.amazonaws.com/onca/xml?Service=AWSECommerceService&Version=2007-10-29&AWSAccessKeyId=[YourAccessKeyId]&Operation=ListLookup&ListType=WishList&ListId=1U5EXVPVS3WP5&ResponseGroup=ListItems,Medium&ProductPage=2)

**아마존과 구글 스프레드시트를 매쉬업하는 파이썬 코드**





↗  
↗

↗  
↗





- GSheetForAmazonList 클래스는 구글 GData 라이브러리를 위한 유틸리티 메서드를 제공한다.
- 모든 위시리스트 아이템이 요청한 데이터를 온전하게 가지고 있지 않을 수도 있기 때문에 에러 처리는 필수다. 데이터가 누락된 경우라도 코드가 지속적으로 동작하는 것은 아주 중요하다.
- 구글 스프레드시트용 파이썬 GData 인터페이스에서는 공백 셀을 추가할 수 없다.
- amazonWishList.ListItems는 파이썬 생성기이며 아마존의 아이템을 한 번에 하나씩 꺼내서 반복연산을 처리할 수 있게 해준다. 파이썬 생성기에 대한 튜토리얼은 <http://>

[www.ibm.com/developerworks/library/l-pycon.html?t=gr,lnxw16=PyIntro](http://www.ibm.com/developerworks/library/l-pycon.html?t=gr,lnxw16=PyIntro)을 참고하도록 하자.

- 이 코드의 수행시간을 배치작업을 통해 향상시킬 수 있으며(<http://code.google.com/apis/gdata/batch.html>) 현재 GData 인터페이스와 자바 API 키트에서 지원된다(파이썬에서는 지원되지 않는다).

## 구글 스프레드시트용 젠드 PHP API 키트

<http://framework.zend.com/>

<http://framework.zend.com/manual/en/zend.gdata.spreadsheets.html>

[http://code.google.com/apis/spreadsheets/developers\\_guide\\_php.html](http://code.google.com/apis/spreadsheets/developers_guide_php.html)



마지막 변경: COM을 이용하여 아마존 위시리스트를 마이크로소프트 엑셀로 전달하기



## 조호 API

<http://writer.zoho.com/public/help/zohoapi/fullpage>

---

25 <http://www.technologyreview.com/Biztech/18816/> 는 조호와 구글 문서도구(Google docs) 간의 전쟁에 대한 분석과 이 두 가지 어떻게 비교되는지에 대해 유용한 분석을 제공한다.

## 요약



# 18

---

## 포함 가능한 데이터 포맷으로 마이크로포맷과 RDFa 사용하기

---

- 
- 
-

# Operator를 사용하여 마이크로포맷에 대해 학습하기

<https://addons.mozilla.org/en-US/firefox/addon/4106>

<http://www.kaply.com/weblog/operator/>

<http://upcoming.yahoo.com/event/144855>

- Addresses(adr)
- Contacts(hCard)
- Events(hCalendar)
- Location(geo)

---

<sup>1</sup> 0.9b에 대해 소개하고 있는 블로그 포스트, <http://www.kaply.com/weblog/2007/12/03/operator-09-beta-available/>를 보도  
록 하라. Operator의 가장 최신 개발자 버전을 <http://www.kaply.com/operator/operator.xpi> 에서 다운로드할 수 있다.



- Tagspaces(tag 또는 rel-tag)<sup>2</sup>
- Bookmarks(xFolk)
- Resources(RDF)

<sup>2</sup> Operator에서 tag라고 불리는 포맷은 <http://microformats.org>에서는 rel-tag로 알려져 있다.

- 이벤트 정보를 다양한 캘린더로 전송할 수 있다.
- 이벤트 정보를 iCalendar 포맷으로 다운로드할 수 있다.
- 15장에서 설명하였듯이 [upcoming.yahoo.com](http://upcoming.yahoo.com)으로부터 이벤트 정보를 추출하기 위해 [upcoming.yahoo.com](http://upcoming.yahoo.com) API를 사용할 수 있다.

<http://upcoming.yahoo.com/event/144855>

▶ **TIP!** Operator의 General 탭에서 Debug Mode 옵션을 체크하여 사용하면 각 마이크로포맷 인스턴스의 Debug 액션에 접근할 수 있으므로 이번 실습에서 도움을 받을 수 있다. Debug 액션은 마이크로포맷 인스턴스를 포함하는 (X)HTML 소스를 보여준다.

**adr(Addresses)**

<http://maps.google.com/maps?q=1401%20N%20Shoreline%20Blvd.,%20California,%20Mountain%20View,%2094043>

**hCard(Contacts)**

<http://address.yahoo.com/?fn=Computer%20History%20Museum&co=Computer%20History%20Museum&ha1=1401%20N%20Shoreline%20Blvd.&hc=Mountain%20View&hs=California&hz=94043&A=C>

**hCalendar(Events)**

geo(Locations)



tag(Tagspaces)

## 마이크로포맷에 대한 정의와 설계목표

인간을 첫 번째로 고려하고 기계를 두 번째로 고려하여 설계된 마이크로포맷은 폭넓게 수용된 기존 표준들의 기반 위에 개발된 간단한 공개 데이터 포맷의 집합이다.

마이크로포맷은 리소스, 툴, 그리고 서비스의 분산 개발이 가능하도록 기존의 상호동작 가능한 표준에 기반한 간략하고 유추하기 쉬운 클래스명을 사용하여 콘텐

---

<sup>3</sup> <http://microformats.org/wiki/what-are-microformats> 이 <http://microformats.org/wiki?title=what-are-microformats&direction=next&oldid=21422> 로 수정되었다.

츠의 사용 패턴을 정규화하며 인간이 읽을 수 있는 (X)HTML/XML 문서, Atom/RSS 피드, 그리고 일반 XML에 특정 문제 도메인을 위한 시멘틱 마크업을 포함시키는 간단한 표기법으로 정의될 수 있다.

- 인간이 읽을 수 있는 (X)HTML/XML 문서, Atom/RSS 피드, 그리고 일반 XML
- 간략하고 유추하기 쉬운 클래스 이름의 사용
- 무한하게 확장이 가능하며 제한이 없다.
- 모든 텍사너미taxonomy, 온톨로지, 그리고 다른 추상화 관련 문제에 대한 만능 해결책이다.

■



## 마이크로포맷 디자인 패턴

rel-design-pattern

### 어트리뷰트 내에 명시된 여러 개의 값

공백으로 구분하여 각각의 값들을 분리함으로써 어트리뷰트에 여러 개의 값을 추가할 수 있다. 다음은 어떻게 HTML 엘리먼트를 여러 개의 클래스와 연결하는지 보여준다. 또한 이것은 rel과 rev와 같은 다른 어트리뷰트에 있어 표준적인 사용법이다.

`http://www.w3.org/TR/html401/struct/links.html#edef-rel`

`http://www.w3.org/TR/html401/struct/links.html#edef-rev`

## class-design-pattern

abbr-design-pattern

include-pattern

---

6 <http://microformats.org/wiki/abbr-design-pattern>  
7 <http://microformats.org/wiki/include-pattern>

## 마이크로포맷 예제

[http://examples.mashupguide.net/ch18/sample\\_microformats.html](http://examples.mashupguide.net/ch18/sample_microformats.html)

### Operator에 사용자 스크립트 추가하기

Operator에 사용자 스크립트를 설치하려면, 스크립트를 여러분의 로컬 드라이브로 다운로드 한 후에 Operator의 Options에 있는 User Scripts 탭으로 이동한다. 그 뒤에 New 버튼을 클릭하고 드라이브에 저장된 스크립트의 경로를 입력한다. Data formats 탭으로 이동하여 새로운 포맷을 로딩(New 버튼을 사용하여)할 수 있으며 Actions 탭으로 이동하여 Operator 탭에 액션들이 노출되도록 설정할 수 있다.

스트립트가 적용될 수 있도록 파이어폭스를 재시작해야 한다는 것을 잊지말자. <http://www.kaply.com/weblog/operator/>에서 보다 자세한 내용을 확인할 수 있다. 다양한Operator 사용자 스크립트를 찾아보려면 <http://www.kaply.com/weblog/operator-user-scripts/> 을 확인해보도록 하자.

rel-license

**rel-tag**

**xfn**

xFolk

<https://secure.del.icio.us/login?url=http%3A%2F%2Fwww.w3.org%2FPeople%2FBerners-Lee%2F&title=Tim%20Berners-Lee&notes=The%20inventer%20of%20the%20WWW&v=4>



geo

<http://flickr.com/photos/raymondye/18389540/>

<http://maps.google.com/maps?ll=37.8721,-122.257704&q=37.8721,-122.257704>

hCard와 adr

- post-office-box
- extended-address
- street-address
- locality
- region
- postal-code
- country-name

## hCalendar

### 다른 마이크로포맷들

- xoxo(<http://microformats.org/wiki/xoxo>)는 계층적인 내용, 즉 중첩된 목록을 나타낸다.
- vote-links(<http://microformats.org/wiki/vote-links>)는 링크가 vote-for(찬성), vote-abstain(기권), 또는 vote-against(반대) 중 어느 것에 해당하는지를 표시한다.
- hReview(<http://microformats.org/wiki/hreview>)는 URL에 대한 리뷰를 나타낸다
- hResume(<http://microformats.org/wiki/hresume>)는 이력서를 나타낸다.

## 실전에서의 마이크로포맷

- upcoming.yahoo.com과 eventful.com은 adr, hCard, hCalendar, tag, 그리고 geo을 사용한다.
- 야후! Local은 adr과 hCard를 사용한다.
- 테크노라티는 hCard와 adr을 사용한다.

## 마이크로포맷으로 프로그래밍하기

---

9 <http://smackman.com/2006/06/01/an-old-idea/> 와 <http://lists.w3.org/Archives/Public/public-rdf-in-xml-tf/2006Jun/0011.html>

10 마이크로포맷의 검증에 대한 보다 자세한 내용은 <http://norman.walsh.name/2006/04/13/validatingMicroformats> 을 참고하라. 에릭 반 더 블리스트(Erik van der Vlist)는 이러한 분석내용을 [http://eric.van-der-vlist.com/blog/2277\\_Validating\\_microformats.item](http://eric.van-der-vlist.com/blog/2277_Validating_microformats.item)에 추가하였다.

## 특정 언어용 라이브러리

- mofo(<http://mofo.rubyforge.org/>)는 hCard, hCalendar 그리고 xfn을 포함하는 다양한 마이크로포맷 지원기능을 가진 루비(Ruby) 라이브러리이다.
  - uformats(<http://rubyforge.org/projects/uformats/>)는 hReview, hCard, hCalendar, rel-tag, rel-license 그리고 include-pattern을 지원하는 또 다른 루비 라이브러리이다.
  - PHP5에서는 hCard를 지원하는 hKit(<http://allinthehead.com/hkit/>)의 사용을 고려해보도록 하라.
  - 아마도 최고의 라이브러리는 파이어폭스 부가기능인 Operator의 핵심부분에 포함된 Microformats.js 일 것이다.<sup>12</sup>
- 
- user-scripts를 다운로드하고 설치하여 새로운 액션과 새로운 마이크로포맷을 추가하라(<http://www.kaply.com/weblog/operator-user-scripts/>).
  - 기존의 스크립트와 문서를 연구하여 새로운 액션이나 새로운 마이크로포맷에 대한 지원 기능을 작성하라.<sup>14</sup>
  - 마이크로포맷을 이용하는 코드상의 미묘한 차이를 이해하기 위해 Operator의 코드를 연구하라.<sup>15</sup>

---

11 <http://norman.walsh.name/2005/09/05/microformats#comment0008>

12 <http://svn.mozilla.org/labs/operator/chrome/operator/content/Microformats/Microformats.js>

13 [http://www.readwriteweb.com/archives/mozilla\\_does\\_microformats\\_firefox3.php](http://www.readwriteweb.com/archives/mozilla_does_microformats_firefox3.php)

14 <http://www.kaply.com/weblog/2007/04/24/operator-action-architecture/> 와 <http://www.kaply.com/weblog/2007/04/18/microformat-objects-in-javascript/>

15 <http://svn.mozilla.org/labs/operator/>

# Operator 스크립트 작성하기

<http://www.kaply.com/weblog/operator-user-scripts/>

<http://www.kaply.com/weblog/operator-user-scripts/creating-a-microformat-action-user-script-basic/>

1. 튜토리얼에 포함된 스크립트를 설치하여 어떻게 동작하는지 이해하라.
2. adr 인스턴스에 대해 주소-좌표 변환을 수행하도록 스크립트를 수정하라.

## 튜토리얼 스크립트에 대한 이해

<http://www.kaply.com/weblog/wp-content/uploads/2007/07/dominos.js>

<http://upcoming.yahoo.com/event/144855>

<http://www.dominos.com/apps/storelocator-EN.jsp?street=1401%20N%20Shoreline%20Bld.&cityStateZip=California,%20Mountain%20View%2094043>

- dominos 자바스크립트 오브젝트는 액션을 정의한다. 액션은 description, shortDescription, scop, 그리고 doAction 등 4가지의 프로퍼티로 구성된다.
- 자바스크립트 오브젝트의 이름, description, shortDescription을 새로 작성하는 스크립트의 목적에 알맞게 수정해야 하다.
- scope 프로퍼티는 액션을 특정한 데이터 포맷에 지정할 때 사용한다. 다음의 코드

는 어떤 adr 인스턴스에도 액션이 적용됨을 나타낸다. 다음과 같이 locality 프로퍼티를 사용하여 adr 인스턴스에만 적용되도록 제한하거나

또는

과 같이 특정 URL에만 적용되도록 할 수 있다.

- doAction 프로퍼티에 연결된 함수는 adr 인스턴스의 다양한 부분들을 연결하여 실제로 도미노 피자에 대한 URL을 생성한다. 이 함수를 수정하기 위해서는 주소를 좌표로 변환하기 위해 사용할 서비스인 <http://geocoder.us>의 URL 구조를 이해해야 한다.
- Operator 스크립트의 가장 간단한 형태의 액션은 브라우저가 로딩하게 되는 URL을 반환하는 것이라는 점에 주목하라(<http://www.kaply.com/weblog/operator-user-scripts/>를 읽어보면 어떻게 Operator 액션들이 다른 동작을 수행하도록 할 수 있는지 배울 수 있다).

## 주소-좌표 변환 스크립트 작성하기



`http://geocoder.us/demo.cgi?address={address}`

`http://geocoder.us/demo.cgi?address=1600+Pennsylvania+Ave%2C+Washington+DC`

`// http://www.kaply.com/weblog/wp-content/uploads/2007/07/dominos.js`을 수정하였다.

<http://geocoder.us/demo.cgi?address=1401%20N%20Shoreline%20Blvd.,%20Mountain%20View,%20California,%2094043>

## 마이크로포맷을 보완해줄 RDFa

- RDF 문서는 리소스들을 주어-술어-목적어(트리플릿이라고 한다) 형태로 표현하는 일련의 문장들로 구성된다. 좀 더 자세히 설명하자면, 리소스(R)가 특정한 값(V)의 프로퍼티(P)를 갖는다는 것을 명시하는 문장들로 이루어진다는 의미이다. 예를 들면, (“Raymond Yee”, “has age of”, 40)과 같이 표현된다.
- RDF에서 사용되는 용어는 리소스의 타입과 프로퍼티를 위한 조건 등을 표현하는 방법을 정의한다. 예를 들어, 가계도를 나타내는 용어들은 “is mother of”와 “is sister of”와 같은 프로퍼티를 정의한다.
- 일단 이렇게 RPV를 구성하고 나면, RPV에 다양한 논리적인 조건을 추가할 수 있다. 만약 P=“has age of” 이고 V가 30보다 크다( $V > 30$ )로 설정된 RPV를 가지고 있다면, (R, “has to trust status”, No)와 같은 RPV를 추가할 수 있다. 그러면 컴퓨터는 이 RPV로부터, 30세가 넘는 사람을 신뢰해서는 안 되기 때문에 30세 이상인 레이몬드 이(Raymond Yee)를 신뢰해선 안 된다는 결론을 도출할 수 있다.

## 향후 연구에 참고할 만한 문서

- <http://microformatique.com/book/>의 마이크로포맷 관련 서적
- 미카 두빈코Micah Dubinko의 “What are Microformats?”<sup>16</sup>
- 우췌 오그부지Uche Ogbuji의 “Microformats in Context”<sup>18</sup>

---

16 [http://examples.mashupguide.net/ch18/sample\\_rdf.html](http://examples.mashupguide.net/ch18/sample_rdf.html)

17 <http://www.xml.com/pub/a/2005/03/23/deviant.html>

18 <http://www.xml.com/pub/a/2006/04/26/microformats-grddl-rdfa-nvdl.html>

## 요약

# 19

---

## 검색 통합하기

---

구글 Ajax 검색

<http://code.google.com/apis/ajaxsearch/signup.html>

## 검색결과 다루기

- 
- 1 <http://examples.mashupguide.net/ch19/google.ajax.1.html>
  - 2 <http://examples.mashupguide.net/ch19/google.ajax.2.html>

- 구글 맵에 지역 검색 추가하기:

<http://www.google.com/uds/solutions/localsearch/index.html>

- 위젯이 제공하는 정리된 형태의 결과가 아닌 가공되지 않은 결과를 검색하기: <http://www.google.com/uds/samples/apidocs/raw-searchers.html>

<http://code.google.com/apis/ajaxsearch/samples.html>

<http://www.google.com/uds/samples/apidocs/raw-searchers.html>

## 야후! 검색

<https://developer.yahoo.com/wsregapp/index.php>

<https://developer.yahoo.com/wsregapp/index.php?view>

<http://developer.yahoo.com/auth/>

<http://examples.mashupguide.net/ch07/yahoo.php>



<http://search.yahooapis.com/WebSearchService/V1/webSearch?appid=YahooDemo&query=madonna&results=2>

[http://search.yahooapis.com/WebSearchService/V1/webSearch?appid=\[YourAppID\]&query=flower&results=1](http://search.yahooapis.com/WebSearchService/V1/webSearch?appid=[YourAppID]&query=flower&results=1)



<http://developer.yahoo.com/search/web/V1/webSearch.html>

<http://search.yahooapis.com/WebSearchService/V1/WebSearchResponse.xsd>

<http://developer.yahoo.com/download/>

## 야후! 이미지

<http://developer.yahoo.com/search/image/V1/imageSearch.html>

<http://search.yahooapis.com/ImageSearchService/V1/imageSearch?appid=YahooDemo&query=Corvette&results=2>

[http://search.yahooapis.com/ImageSearchService/V1/imageSearch?appid=\[YourAppId\]&query=flower&results=2](http://search.yahooapis.com/ImageSearchService/V1/imageSearch?appid=[YourAppId]&query=flower&results=2)





<http://developer.yahoo.com/search/local/V2/localSearch.html>

## 마이크로소프트 Live.com 검색

<http://soap.search.msn.com/webservices.asmx?wsdl>

<http://dev.live.com/blogs/livesearch/archive/2006/03/23/27.aspx>

<http://search.msn.com/developer>

<http://msdn2.microsoft.com/en-us/library/bb251815.aspx>

<http://www.microsoft.com/express/>

**\* NOTE!** 이론적으로는 마이크로소프트의 개발환경을 사용하지 않더라도 WSDL 인터페이스를 통해서 Live.com을 사용할 수 있어야 하지만, 실제로는 이들 툴에 맞도록 설계된 예제와 문서들 때문에 마이크로소프트의 툴을 사용하는 것이 훨씬 쉽다는 것을 알게 될 것이다. 나는 다른 툴을 사용하는 경우에도 중요 파라미터들에 대한 이해를 돕기 위해 마이크로소프트 툴을 참고하곤 한다.

**\* NOTE!** XML Spy와 oXygen 은 무료 소프트웨어가 아니지만 30일간 무료로 사용해 볼 수 있다. 이외에 WSDL과 SOAP를 쉽게 다룰 수 있게 해주는 무료 소프트웨어(아마도 이클립스의 경우는 무료일 것이다)는 찾지 못했다.

<http://msdn2.microsoft.com/en-us/library/bb266177.aspx>

■ 연결 엔드포인트: <http://soap.search.msn.com:80/webservices.asmx>

- SOAPaction HTTP 헤더:

<http://schemas.microsoft.com/MSNSearch/2005/09/fex/Search>

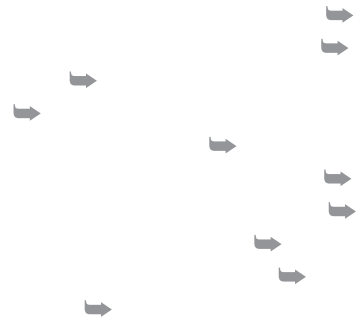
- SOAP 요청을 위한 다음과 같은 템플릿

- AppID: 애플리케이션 키 값
- CultureInfo: 사용 가능한 값의 목록에서 선택되어야 할 언어와 지역 정보<sup>3</sup>(예를 들어, en-US)
- Query: 검색어
- Requests: 사용 가능한 값의 목록(예를 들어, Web, Ads, Image)<sup>4</sup>에서 선택된 SourceRequest 값의 목록
- Flags: None, DisableHostCollapsing, DisableSpellCheckForSpecialWords 또는 MarkQueryWord(None이 기본값이다) 중 하나의 값을 갖는다.
- Location: 검색을 위한 위도, 경도를 나타내며 선택적으로 검색 반경을 명시한다.
- SafeSearch: Strict, Moderate, 또는 Off의 값 중에 하나를 갖는다(Moderate가 기본값이다).

---

<sup>3</sup> <http://msdn2.microsoft.com/en-us/library/bb266177.aspx>

<sup>4</sup> <http://msdn2.microsoft.com/en-us/library/bb266167.aspx>





OpenSearch

<http://blog.mashupguide.net/?s=Yahoo&searchsubmit=Find>



<http://blog.mashupguide.net/?s={searchTerms}&searchsubmit=Find>

- <http://msdn2.microsoft.com/en-us/library/Aa744112.aspx> (for Internet Explorer 7)
- [http://developer.mozilla.org/en/docs/Adding\\_search\\_engines\\_from\\_web\\_pages](http://developer.mozilla.org/en/docs/Adding_search_engines_from_web_pages)  
(for Firefox)

- <https://addons.mozilla.org/en-US/firefox/browse/type:4> (파이어폭스가 관리하는 인기 검색 엔진 플러그인 목록)
- <http://mycroft.mozdev.org/dlstats.html> (가장 많이 다운로드된 검색엔진 플러그인)

모질라 기반 브라우저에서 셜록(Sherlock, 전통적인 애플의 검색 툴)의 구현이 GET 요청만을 지원하는데 반해, OpenSearch의 도입으로 POST 요청도 사용할 수 있게 되었다. 하지만 불행히도 현재 인터넷 익스플로러 7에서는 지원되지 않는다.

<http://inner.geek.nz/projects/wordpress-plugins/mycroft-search-plugin-generator/>

[http://www.opensearch.org/Specifications/OpenSearch/1.1#OpenSearch\\_response\\_elements](http://www.opensearch.org/Specifications/OpenSearch/1.1#OpenSearch_response_elements)

[http://www.opensearch.org/Community/OpenSearch\\_search\\_clients](http://www.opensearch.org/Community/OpenSearch_search_clients)

[http://www.opensearch.org/Community/OpenSearch\\_software](http://www.opensearch.org/Community/OpenSearch_software)

<http://williamsburger.com/wb/archives/opensearch-v-1-1>

## 구글 데스크톱 HTTP/XML 게이트웨이

<http://desktop.google.com/dev/queryapi.html#httpxml>

**\* NOTE!** 또한 윈도우에서 사용할 수 있는 COM 기반 인터페이스도 존재하며 <http://desktop.google.com/dev/queryapi.html#registering> 에서 찾아볼 수 있다. XML 게이트웨이는 맥 OS X의 구글 데스크톱 맥 1.0.3 이상의 버전에서 동작한다. API는 현재 구글 데스크톱의 리눅스 버전에서는 동작하지 않는다.

HKEY\_CURRENT\_USER\Software\Google\Google Desktop\API\search\_url

<http://127.0.0.1:4664/search&s={SECRETKEY}?q=>

<http://127.0.0.1:4664/search&s={SECRETKEY}?q=bach>

요약



# 부록 A

---

## 아이폰 매쉬업 사이트 만들기

---

오픈 API로 작업하기

네이버 오픈 API 키 발급받기

## | 키 이용등록/수정

이 곳은, 오픈 API를 이용해 새로운 사이트 및 프로그램을 개발하시기 위한 키(Key)를 발급받고 확인하시는 곳입니다.

키 발급에는 일반키 발급 및 지도키 발급으로 2가지가 있습니다.

일반키는 지도 API를 제외한 모든 API 서비스를 이용하실 수 있습니다.  
지도 서비스 개발을 위해서는 반드시 지도 API용 키를 별도로 발급 받으시기 바랍니다.

일반키 발급

지도키 발급

등록한 것과 다른 디렉토리에서 동작하려면 추가로 키를 발급 받아 주세요.

번호	키	디렉토리	키 삭제
1	c13a698804efb5e7eac7dc5a43af4c8	p-p.pe.kr:8180/map/	<a href="#">삭제하기</a>
2	998f7ac176356b35a89741c139bd3fdb	purpleware.net:8180/	<a href="#">삭제하기</a>
3	0477890c89341e820f6b917b1406ad7e	imashup.net/	<a href="#">삭제하기</a>
4	3009f151279331a9d80e0e971fcc967a	imashup.net:8889/	<a href="#">삭제하기</a>

디렉토리 지정 (아래 설명문을 잘 읽고, 지도키를 발급 받으시기 바랍니다.)

[지도key추가](#)

지도API를 사용할 웹사이트 도메인 및 디렉토리 지정 방법

· 지도 API를 이용할 디렉토리를 지정하시면, 해당 디렉토리를 포함한 하위 디렉토리에서 동작이 가능합니다.

ex) `http://www.sample.com/directory1` 을 등록해 해당 키로 지도 API를 이용할 경우, 동작가능한 디렉토리는 아래와 같습니다.

→ `http://www.sample.com/directory1/page.html`

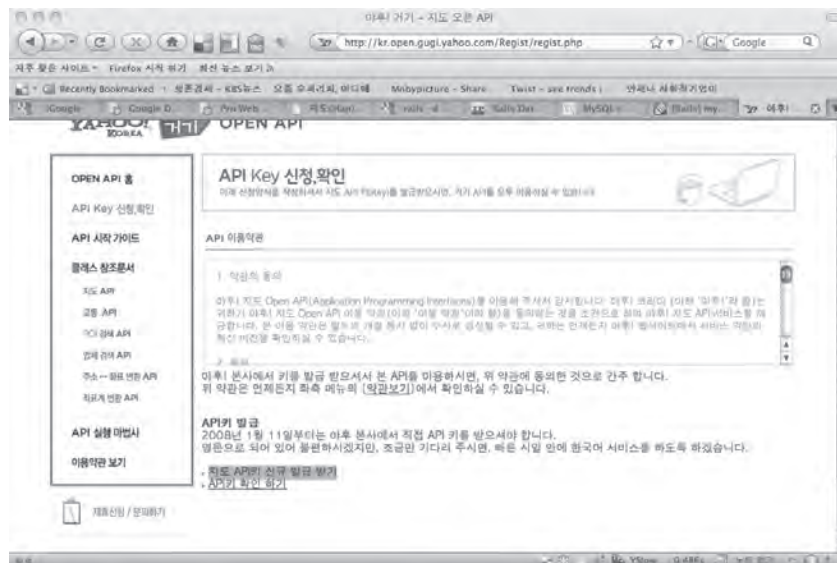
→ `http://www.sample.com/directory1/subdirectory1/page.html`

따라서, 등록되지 않은 아래 디렉토리는 동작이 불가능 합니다.

→ `http://www.sample.com/directory2/page.html`

## 네이버 지도 키 발급받기

## 야후! 거기 오픈 API 이용하기



Register your application on the Yahoo! Developer Network

https://developer.yahoo.com/wsregapp/index.php

If you've already registered for an application ID, you can see them [here](#).

**Developer Registration**

Fields marked with an asterisk \* are required.

\*Yahoo ID:

\*Authentication method: [Click here for more information](#)

☒ **Generic, No user authentication required**  
This applet will allow you to make calls to our non-authenticated web services

☐ **Browser Based Authentication**  
Use this option for browser applications

\*Developer/Company Name:   
For example: 'Joe/Jane Developer' or 'BigCo Inc.'

\*Product name:   
For example: 'My Yahoo! Enabled Web App'

Web Application URL:   
For example: 'http://myapp.com/welcome.html'

\*Contact email:   
For example: 'developer@domain.com'

Phone number:   
For example: '123-456-7890'

\*Description of application:   
(250 characters or less):

\*Required access scopes: The user will be prompted to grant access every time he logs in:

Register your application on the Yahoo! Developer Network

https://developer.yahoo.com/wsregapp/index.php

**YAHOO! DEVELOPER NETWORK** [Yahoo! - Help](#)

Welcome to the Developer Network.

Congratulations, **smithyoo17**, your application is now registered! Yay!

☒ Your application id is  
**YM6XOVV34F26J9M6IAJAWODYOUJKe1YGHzwUmSeTPe1Qnah1MtkKpLnu\_nNDwWQ--**  
 For application endpoint -> <http://imashup.net/mashupmap>

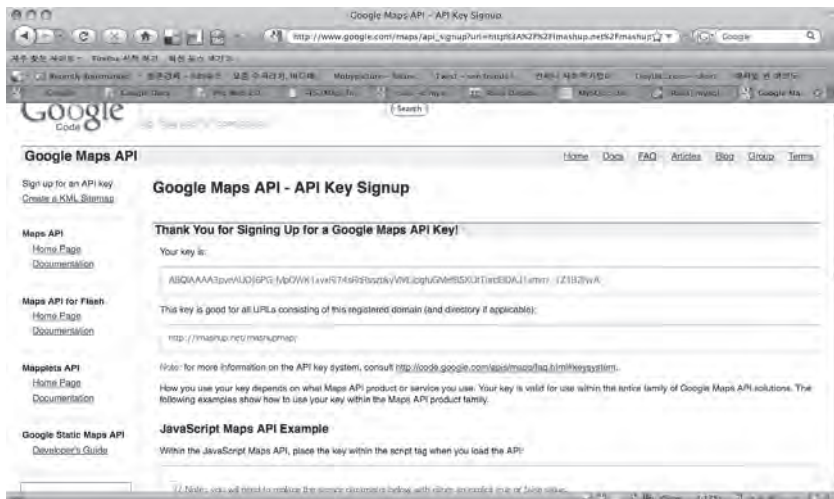
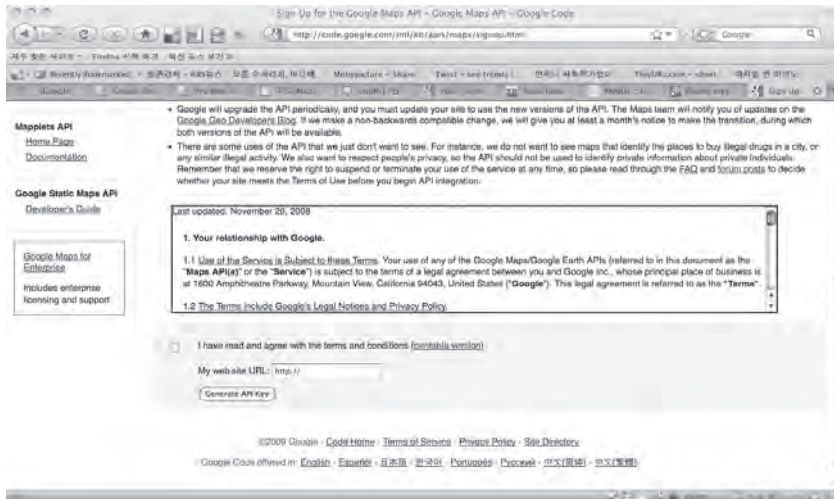
To view your current application ids, click [here](#).  
 You may want to print this [page](#) for your records.  
 Here's what to do next:

[Finished](#)

Copyright © 2009 Yahoo! Inc. All rights reserved. [Copyright/Privacy Policy](#) | [Terms of Service](#) | [Guide to Online Security](#)  
 NOTICE: We collect personal information on this site.  
 To learn more about how we use your information, see our [Privacy Policy](#).

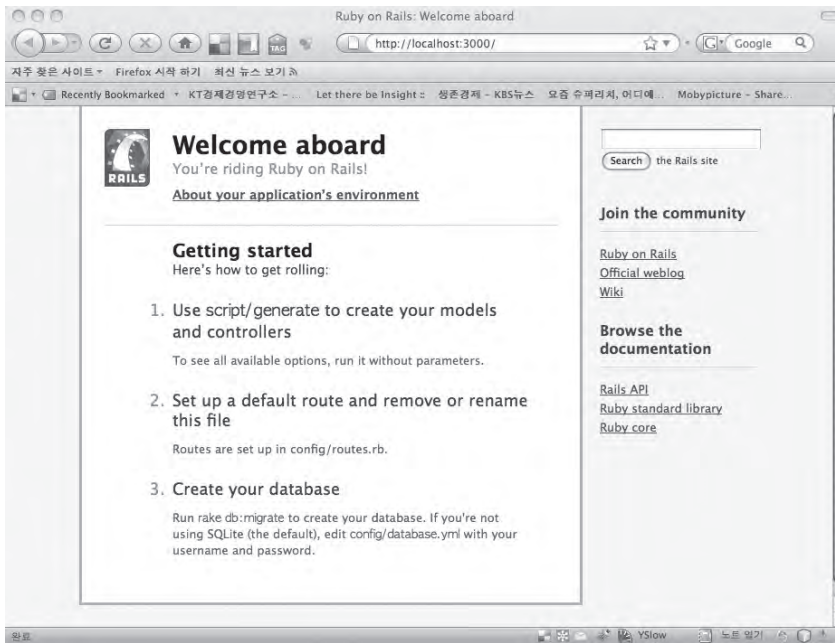


## 구글 오픈 API 이용하기



Rails Project 시작하기

rails 의 webrick 서버 구동하기



## Rails의 MVC (Model, View, Controller)

### Model 생성하기

### 데이터베이스 생성하기

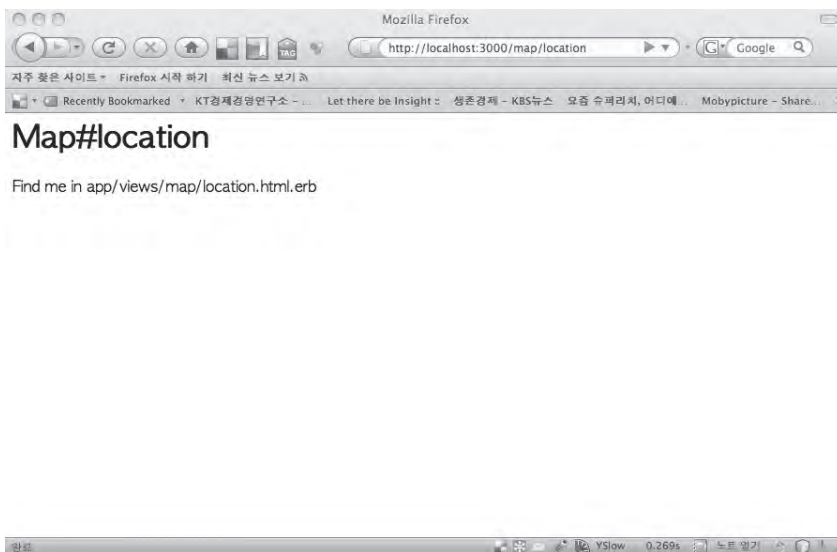


액티브레코드 model 객체 만들기

Controller생성하기

## Controller과 View를 함께 생성하기

```
ruby script/generate controller [controller_name] [view1_name] [view2_name] ... [viewN_name]
```





## 본격적인 Mashupmap 지도 서비스 만들기

Model 객체에 테이블 작성

## Map Controller

### 검색 기능 만들기

#### 주소검색



지역검색

지명검색

location view





검색 (Search)

모두

검색



인기 위치 검색어 (HOTSPOT)

**TODAY** 수내역 강남역 분당 서  
래마을 서울 야후 정자역 삼성  
플라자 수내

**HOTSPOT** 수내역 탐앤탐스 삼  
성플라자 스타벅스 연세대학교  
홍익대 홍익대학교 강남역 분당



검색 결과 보여주기



## 검색결과 (Result)

전체검색결과 : 9

주소검색 : 9

경기도 성남시 분당구 정자동



경기도 성남시 분당구 정자1동



경기도 성남시 분당구 정자2동



경기도 성남시 분당구 정자3동



경기도 수원시 장안구 정자동



경기도 수원시 장안구 정자1동



경기도 수원시 장안구 정자2동









맺음말



## 부록 *B*

---

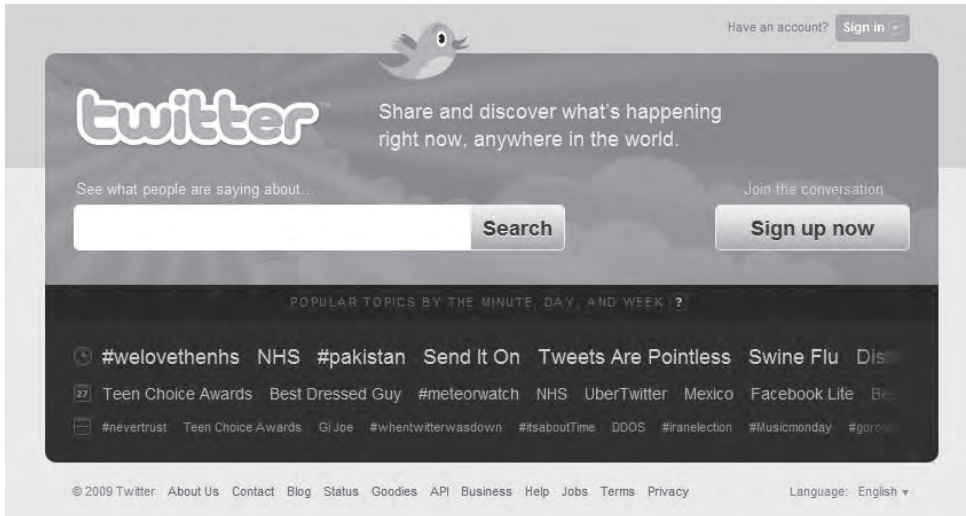
### 트위터 기반 매쉬업

---

트위터란?

- 
- 
- 
-

## A. 트위터 가입하기



Full name : 이름을 한국어나 영어로 입력한다.

Username : 트위터에서 사용할 사용자명을 입력한다.


Password : 비밀 번호를 입력한다.


Email : 이메일 주소를 적어 넣는다.


I want the inside scoop : 이 항목을 체크하면 트위터 소식을 이메일로 보내준다.

Type the words above : 입력창 상단의 영문을 그대로 입력한다. 이 항목은 Capcha라고 하는 것으로 악성 프로그램이 무단으로 계정을 생성하지 못하도록 하기 위한 것이다.


## See if your friends are on Twitter

 Gmail

 Yahoo

 AOL

« We can check if your friends on other services already have a Twitter account. **Choose a service** from the list on the left.







[Skip this step](#)

## Look who else is here. Start following them!

Maybe you've heard of these Twitter users? Select the people you'd like to start following.

☒ Select All

<input checked="" type="checkbox"/>		<b>David Gregory</b>	Host of NBC's Meet the Press
<input checked="" type="checkbox"/>		<b>The New York Times</b>	Where the Conversation Begins. Home page stories from NYTimes.com, special features and more.
<input checked="" type="checkbox"/>		<b>Padmasree</b>	CTO of Cisco
<input checked="" type="checkbox"/>		<b>Biz Stone</b>	Co-founder of Twitter, Inc.

### You'll be following



Finish

[Skip this step](#)



twitter

Home Profile Find People Settings Help Sign out

What are you doing?

140


update

Home



### What to do now:

1. Tell us what **you're** doing in the box above
2. Find some friends and follow what **they're** doing
3. Turn on your mobile phone to update your friends on the go

 **promashup20**

    
following followers tweets

Tweets on tees  
n. awesome crowdsourced  
shirts by Threadless.

Home

@promashup20

Direct Messages

Favorites

Search

## B. 다른 사용자 팔로우하기

The screenshot shows a Twitter profile for a user named 'javadance'. The profile includes a profile picture of a man with glasses, the name 'javadance', and a 'Follow' button. The bio lists 'Name Park CheonGu', 'Location seoul', 'Web http://www.facebook...', and 'Bio Java Developer, IT Book Translator, Service Architect, Social Service'. It also shows 91 following and 120 followers. The 'Tweets' section shows 516 tweets. The 'Following' section shows a grid of 24 user avatars. The main content area displays two tweets: one retweeted by @hwyhfi 33 minutes ago and another by @twihrt about 6 hours ago.

Home Profile Find People Settings Help Sign out

**javadance**

Follow

RT TDD에서는 개발자가 개발전에 테스트를 먼저 작성하죠. 아마 말씀하신 것은 사용자 테스트이신 듯..  
@hjang27: RT @murianwind: 우리는 개발자가 테스트도 다 해요.  
<http://bit.ly/YtbpD>  
33 minutes ago from hwyhfi

RT 많이 미안해 하셔야@hjang27: RT @widelotus 전해들은 요즘 고연전 풍경 연대를 향한 고대의 자랑 "김,연,아, 김,연,아" 그런 고대를 향한 연대의 반박 "이,명,박, 이,명,박" 그에 대한 고대의 반응 "미,안,해, 미 ...."  
about 6 hours ago from twihrt

RT 김민선씨..용기있는 말씀하셨던 거예요.. 힘내세요.. 팬들은 당신

Name Park CheonGu  
Location seoul  
Web <http://www.facebook...>  
Bio Java Developer, IT Book Translator, Service Architect, Social Service

91 following 120 followers

Tweets 516

Favorites

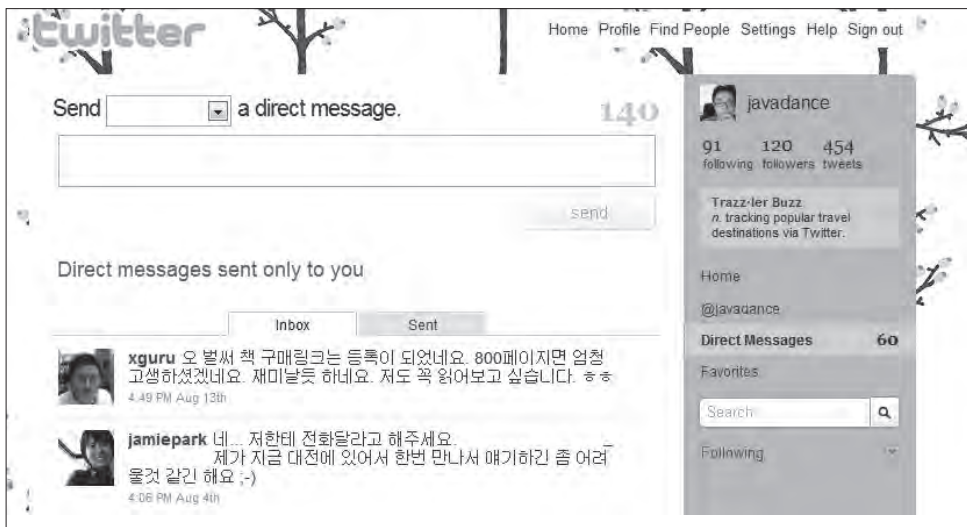
Actions  
and javadance

Following

## C. 글쓰기/댓글 쓰기



#### D. 다이렉트 메시지



## 트위터 기반 매쉬업(트윗봇)

현재 동작중인 Auto Tweet Engine		
@dien_net		<p>IT 관련소식이 가장 빨리 올라오는 커뮤니티 사이트 클리앙 (dien.net) 의 새로운 소식 게시판에 올라오는 글들을 자동으로 트위팅 합니다.</p> <p>bit.ly 단축URL 서비스를 이용하여 해당 글에 대한 바로가기 링크를 제공합니다.</p> <p><b>클리앙 중고장터 모니터링 서비스</b> 트위터 DirectMessage(DM)을 통해서 @dien_net 에 "buy Wii" 와 같은 메시지를 보내면 Wii 판매 같은 글이 올라올때 자동으로 알려줍니다.</p> <ul style="list-style-type: none"> <li>• buy 상품명 : 뭔가를 사고 싶을때 사용합니다. [판매] 글이 올라오면 DM 으로 알려드립니다.</li> <li>• sell 상품명 : 뭔가를 팔고 싶을때 사용합니다. [구입] 글이 올라오면 DM 으로 알려드립니다.</li> </ul>
@info_guru		<p>한국내의 다양한 정보를 매일 지정된 시간대에 자동으로 트위팅 합니다. 다른 정보 추가문의는 @xguru 로</p> <ul style="list-style-type: none"> <li>• 매 일 08:00, 18:00 - 환율정보 : 미국 USD   일본 JPY   유로 EUR   영국 GBP   캐나다 CAD   중국 CNY</li> <li>• 매 일 06:00, 12:00 - 날씨정보 : 서울/경기</li> <li>• 월~금 22:30, 05:00 - 미국증시정보 : 다우, 나스닥, S&amp;P 500 - ( 미국/뉴욕 시간 기준입니다 )</li> <li>• 월~금 09:00, 15:00 - 한국증시정보 : 코스피, 코스닥, 국채</li> <li>• 토요일 22:00 - 로또 추첨 결과</li> </ul> <p><b>DM 기반 개인화 정보전달 서비스</b> DM 을 보내서 날씨/생일/기념일/약속에 대해 정해진 시간마다 정보를 받을수 있습니다. <b>@info_guru DM 명령어</b> 설명을 참고하세요.</p>
@navercast		<p>네이버가 제공하는 <b>네이버캐스트</b> - 오늘의 생활 문화 컨텐츠 정보를 시간별로 트위팅 합니다.</p> <ul style="list-style-type: none"> <li>• 05시 : 지식인의 서재, 06시 : 한국인, 07시 : 오늘의 세계인물, 08시 : 오늘의 책, 09시 : 오늘의 영화</li> <li>• 10시 : 오늘의 포토, 11시 : 오늘의 키친, 12시 : 오늘의 과학, 13시 : 오늘의 뮤직, 14시 : 오늘의 웹툰</li> <li>• 15시 : 아름다운 한국, 16시 : 오늘의 문학, 17시 : 컬처플러스, 18시 : 옛날신문, 19시 : 갤러리N</li> <li>• 20시 : 포토樂보드, 21시 : 스포츠스페셜, 22시 : 영화 스페셜, 23시 : 뮤직스페셜</li> </ul>
@zirim		<p><b>지름도우미</b>가 제공하는 국내 여러 원머데이몰의 오늘 상품을 변경시간마다 즉시 트위팅 합니다. 지름도우미 on Twitter 에 대한 소개글을 참고하세요.</p> <ul style="list-style-type: none"> <li>• 24:00 - 원머데이, 인터파크 모닝커피, 원바이원</li> <li>• 23:00 - 카르페디엠</li> <li>• 10:00 - CJMall 123Shop</li> <li>• 11:00 - 네이버 럭키투데이</li> </ul>



사용 API 소개



## A. REST API와 Search API

### ■ 트위터 REST API

개발자들이 트윗을 포스팅하고 사용자 정보를 수정하는 등의 핵심 데이터에 접근하게 해준다.

### ■ 트위터 Search API

개발자들이 <http://search.twitter.com/> 에서 제공되는 검색 서비스와 상호작용할 수 있도록 해주며 검색 트렌드 관련 데이터에 접근할 수 있도록 해준다.

`statuses/public_timeline`

`http://twitter.com/statuses/public\_timeline.format`

---

---

statuses/friends\_timeline

[http://twitter.com/statuses/friends\\_timeline.format](http://twitter.com/statuses/friends_timeline.format)

---

---

statuses/user\_timeline

[http://twitter.com/statuses/user\\_timeline.format](http://twitter.com/statuses/user_timeline.format)

- id : 선택 파라미터이며 트윗정보를 가져올 특정 사용자의 아이디(시스템에서 부여한 고유한 값)나 스크린명(가입시 입력한 username)을 명시한다. 아래와 같은 형태로 사용하면 된다.

[http://twitter.com/statuses/user\\_timeline/javadance.xml](http://twitter.com/statuses/user_timeline/javadance.xml)

■

```
관리자: 명령 프롬프트
<status>
  <created_at>Fri Aug 07 04:12:22 +0000 2009</created_at>
  <id>3173973897</id>
  <text>@janiepark &#47553;&#53356;&#45208;&#50864;&#47484; &#44060;&#48156;&#54620; &#50629;&#52404;&#51064;&#44163;&#51004;&#47196; &#50508;&#44256; &#51080;&#49845;&#45768;&#45796;;..http://www.linknow.kr</text>
  <source>&lt;a href="http://www.twirl.org/" rel="nofollow"&gt;twirl&lt;/a&gt;</source>
  <truncated>>false</truncated>
  <in_reply_to_status_id>3173329587</in_reply_to_status_id>
  <in_reply_to_user_id>18551127</in_reply_to_user_id>
  <favorited>>false</favorited>
  <in_reply_to_screen_name>janiepark</in_reply_to_screen_name>
  <user>
    <id>14256765</id>
    <name>Park CheonGu</name>
    <screen_name>javadance</screen_name>
    <location>seoul</location>
```

위 응답에서 <user> 엘리먼트내에 <id>엘리먼트의 값으로 표시되는 14256765가 시스템에서 할당된 값으로 이 값을 사용하면 된다. 예를 들면 아래와 같이 사용할 수 있다.

[http://twitter.com/statuses/user\\_timeline.xml?user\\_id=14256765](http://twitter.com/statuses/user_timeline.xml?user_id=14256765)

- screen\_name : 선택 파라미터이며 트윗정보를 가져올 특정 사용자의 스크린명을 명시적으로 지정하기 위해 사용한다. 아래와 같은 형태로 사용하면 된다.

[http://twitter.com/statuses/user\\_timeline.xml?screen\\_name=javadance](http://twitter.com/statuses/user_timeline.xml?screen_name=javadance)

- since\_id : 선택 파라미터로 특정한 아이디값보다 더 큰 값을 갖는 트윗, 즉 더 최근에 포스팅된 트윗을 가져오기 위해 사용하는 파라미터이다. 아래와 같이 사용하면 된다.

[http://twitter.com/statuses/user\\_timeline.xml?since\\_id=12345](http://twitter.com/statuses/user_timeline.xml?since_id=12345)

- max\_id : since\_id 파라미터와는 반대로 특정한 아이디값보다 더 작거나 같은 값을 갖는 트윗, 즉 더 오래전에 포스팅된 트윗만을 가져오기 위해 사용하는 파라미터이다. 아래와 같이 사용하면 된다.

[http://twitter.com/statuses/user\\_timeline.xml?max\\_id=54321](http://twitter.com/statuses/user_timeline.xml?max_id=54321)

- count : 선택 파라미터이며 가져올 트윗의 개수를 명시하며 200개를 넘을 수 없다. 아래와 같이 사용하면 된다.

[http://twitter.com/statuses/user\\_timeline.xml?count=200](http://twitter.com/statuses/user_timeline.xml?count=200)

- page : 선택 파라미터이며 가져올 트윗의 페이지 인덱스를 명시한다.

[http://twitter.com/statuses/user\\_timeline.rss?page=3](http://twitter.com/statuses/user_timeline.rss?page=3)

**statuses/show**

`http://twitter.com/statuses/show/id.format`

- id : 필수 파라미터이며 가져오려는 트윗의 고유 아이디를 명시한다. 아래와 같이 사용할 수 있다

---

---

**statuses/update**

`http://twitter.com/statuses/update.format`

■

■

---

---

## statuses/destroy

<http://twitter.com/statuses/destroy/id.format>

- **id** : 필수 파라미터이며 삭제하려는 트윗의 시스템상의 아이디를 지정한다.

<http://twitter.com/statuses/destroy/12345.xml>

---

---

## direct\_messages

[http://twitter.com/direct\\_messages.format](http://twitter.com/direct_messages.format)

- **since\_id** : 선택 파라미터로 특정한 아이디값보다 더 큰 값을 갖는 다이렉트 메시지, 즉 더 최근에 수신한 다이렉트 메시지를 가져오기 위해 사용하는 파라미터이다. 아래와 같이 사용하면 된다.

[http://twitter.com/direct\\_messages.xml?since\\_id=12345](http://twitter.com/direct_messages.xml?since_id=12345)

■

[http://twitter.com/direct\\_messages.xml?max\\_id=54321](http://twitter.com/direct_messages.xml?max_id=54321)

- **count** : 선택 파라미터이다. 가져올 다이렉트 메시지의 개수를 명시하며 200개를 넘을 수 없다. 아래와 같이 사용하면 된다.

[http://twitter.com/direct\\_messages.xml?count=200](http://twitter.com/direct_messages.xml?count=200)

- **page** : 선택 파라미터이며 가져올 다이렉트 메시지의 페이지 인덱스를 명시한다.

[http://twitter.com/direct\\_messages.xml?page=3](http://twitter.com/direct_messages.xml?page=3)

---

---

## direct\_messages/sent

[http://twitter.com/direct\\_messages/sent.format](http://twitter.com/direct_messages/sent.format)

■

[http://twitter.com/direct\\_messages/sent.xml?since\\_id=12345](http://twitter.com/direct_messages/sent.xml?since_id=12345)

■

[http://twitter.com/direct\\_messages/sent.xml?max\\_id=54321](http://twitter.com/direct_messages/sent.xml?max_id=54321)

- **count** : 선택 파라미터이다. 가져올 다이렉트 메시지의 개수를 명시하며 200개를 넘을 수 없다. 아래와 같이 사용하면 된다.

[http://twitter.com/direct\\_messages/sent.xml?count=5](http://twitter.com/direct_messages/sent.xml?count=5)

- **page** : 선택 파라미터이며 가져올 다이렉트 메시지의 페이지 인덱스를 명시한다.

[http://twitter.com/direct\\_messages/sent.xml?page=3](http://twitter.com/direct_messages/sent.xml?page=3)

---

---

## direct\_messages/new

[http://twitter.com/direct\\_messages/new.format](http://twitter.com/direct_messages/new.format)

- **user** : 필수 파라미터로 수신자의 아이디값이나 스크린명을 지정한다. 스크린명을 숫자형태로 사용하는 사용자의 경우 혼란을 방지하기 위해 **user** 파라미터 대신에 **screen\_name**이나 **user\_id**라는 파라미터도 허용된다.
  - **text**: 필수 파라미터로 여러분이 전송할 다이렉트 메시지의 내용을 지정한다. 반드시 URL 인코딩을 해야 하며 140자가 넘는 내용은 강제로 잘린다는 것을 잊지 말도록 하자.
- 
- 

## direct\_messages/destroy

[http://twitter.com/direct\\_messages/destroy/id.format](http://twitter.com/direct_messages/destroy/id.format)

- **id** : 필수 파라미터이며 삭제할 다이렉트 메시지의 id값을 지정한다. 아래와 같이 사용하면 된다.

[http://twitter.com/direct\\_messages/destroy/12345.xml](http://twitter.com/direct_messages/destroy/12345.xml)

---

---

- 100% 순수 자바로 구현되었으며 JDK 1.4.2 이후의 버전에서 동작한다.
- 구글 안드로이드 플랫폼도 지원한다.
- 추가적인 의존성이 없으므로 Twitter4J 관련 라이브러리만 설치하면 된다.
- OAuth(<http://oauth.net/> 참조)를 지원한다.
- 보다 빠른 퍼포먼스를 위해서 데이터를 gzip으로 압축해서 전송한다.

## B. 네이버 Open API 등록





## C. 네이버 실시간 급상승 검색어 API

<http://openapi.naver.com/search>




## Installing the Plugin

These instructions assume that you have already installed some flavor of Eclipse. If you have not, Eclipse can be downloaded from <http://www.eclipse.org/downloads/>.

### Update sites

If you are already familiar with installing Eclipse plugins, you can just use the update site URL below for your version of Eclipse.

#### Eclipse 3.5 (Galileo)

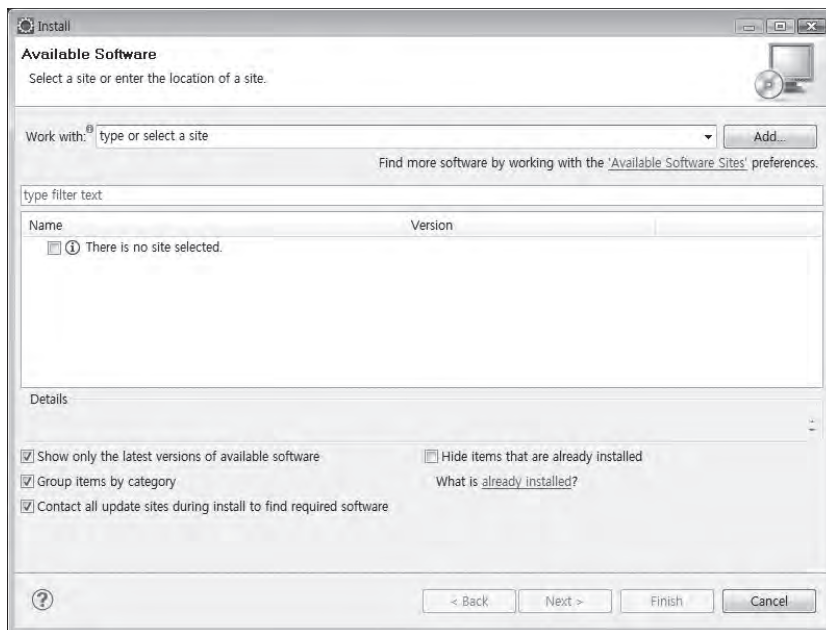
<http://dl.sourceforge.net/eclipse/plugin/3.5>

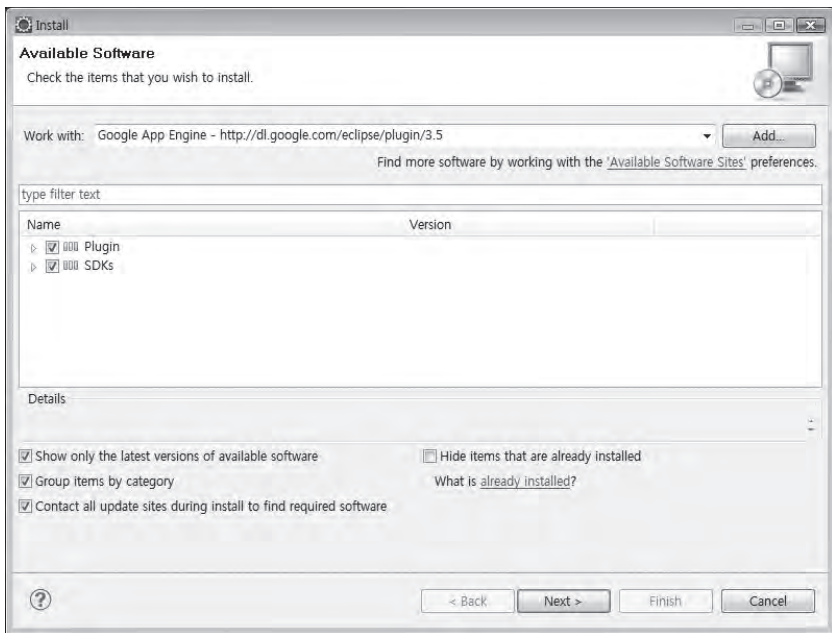
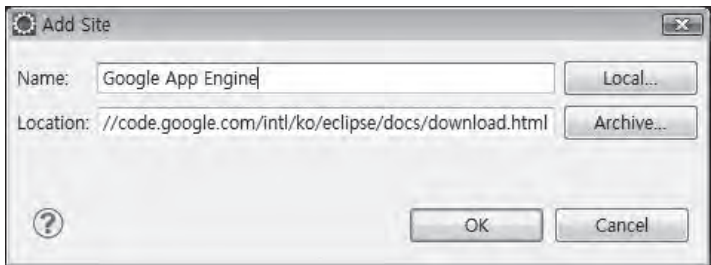
#### Eclipse 3.4 (Ganymede)

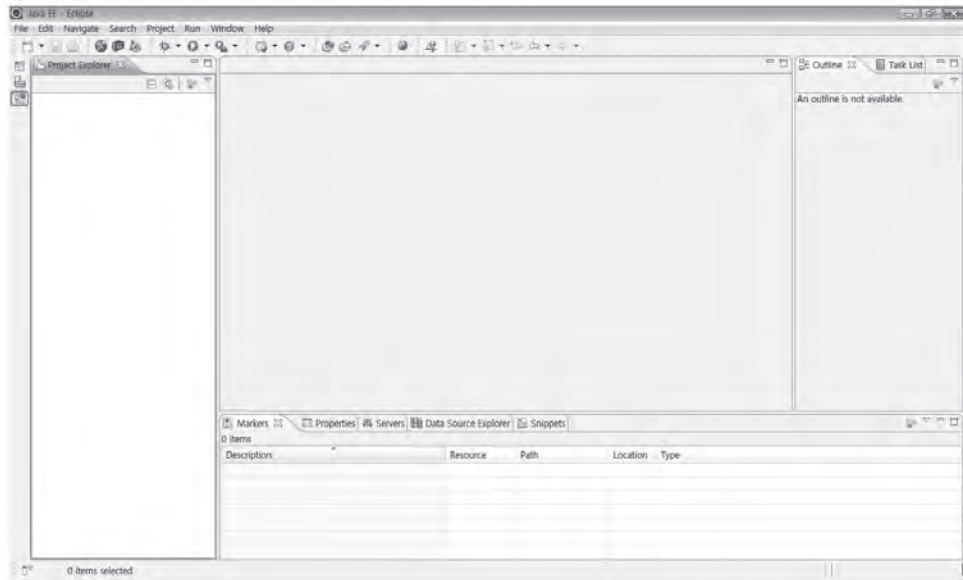
<http://dl.sourceforge.net/eclipse/plugin/3.4>

#### Eclipse 3.3 (Europa)

<http://dl.sourceforge.net/eclipse/plugin/3.3>









Google app engine Google 애플리케이션 엔진에 오신 것을 환영합니다.

**Google 인프라에서 웹 응용 프로그램을 실행합니다.**

Google 애플리케이션 엔진을 이용하는 개발자는 Google의 최첨단 프로그램을 운영하는 데 필요한 인프라를 걱정하지 않고 애플리케이션을 개발할 수 있습니다.

**여정부터가 중요하지 않습니다.**

Google 애플리케이션 엔진은 완벽하게 통합된 개발 환경을 제공합니다.

**작업이 편리합니다.**

Google 애플리케이션 엔진을 사용하면 인프라에 대한 걱정 없이 한 곳에서 수백만 명의 사용자 지원이 가능한 애플리케이션 프로그램을 쉽게 개발할 수 있습니다.

**무료로 시작 가능**

모든 Google 애플리케이션 엔진 애플리케이션은 무료로 사용할 수 있는 개인용 개발 계정, 대역폭 및 저장공간을 제공합니다. 추가 비용이 발생하는 경우에는 Google의 가격으로 리소스를 구입하여 사용할 리소스에 대해서만 지불할 수 있습니다.

**이 서비스는 Google 애플리케이션 엔진의 브랜드 버전입니다.** 시작하려면 Google 계정을 사용하여 Google 애플리케이션 엔진에 로그인하거나 호스팅 확인하여 SDK를 다운로드한 다음 Google의 지침을 따라주세요.

Google 애플리케이션 엔진에 Google 애플리케이션 계정으로 로그인하려면 다음 URL을 참조하십시오.

<https://appengine.google.com/yourdomain.com/>

**Google 계정**

다른 계정으로 로그인

비밀번호

로그인

로그인

로그인

Google 계정이 없으신가요?

가입하기

## Welcome to Google App Engine

Before getting started, you want to learn more about developing and deploying applications. Learn more about Google App Engine by reading the [Getting Started Guide](#), the [FAQ](#), or the [Developer's Guide](#).

[Create an Application](#)

## Verify Your Account by SMS

To create applications with Google App Engine, you need a verification code. Select the country and carrier for your mobile phone and enter your mobile phone number. The verification code will be sent to it via SMS. Note you will only need to verify your account once.

Country and Carrier:

Other (Not Listed) ▾

If your country and carrier are not on the list, select Other (Not Listed). [What carriers are supported?](#)

Mobile Number:

Include your [country code](#) and full phone number. eg. +1 650 555 1212

[Send](#)

## Create an Application

### Application Identifier:

.appspot.com

[Check Availability](#)

You can map this application to your own domain later. [Learn more](#)

### Application Title:

Displayed when users access your application.

### Authentication Options (Advanced): [Learn more](#)

Google App Engine provides an API for authenticating your users. If you choose not to use this, anyone in the world will be able to access your application. However, if you choose to use this, you'll need to specify now who can sign in to your application.

### Open to all Google Accounts users (default)

If your application uses authentication, anyone with a valid Google Account may sign in. (This includes all Gmail Accounts, but does not include accounts on any Google Apps domains.)

[Edit](#)

### Terms of Service:

1.1. Your use of the Google App Engine service (the "Service") is governed by this agreement (the "Terms"). "Google" means Google Inc., located at 1600 Amphitheatre Parkway, Mountain View, CA 94043, United States, and its subsidiaries or affiliates involved in providing the Service.

1.2. In order to use the Service, you must first agree to the Terms. You can agree to the Terms by actually using the Service. You understand and agree that Google will treat your use of the Service as acceptance of the Terms

☐ I accept these terms.

[Save](#)

[Cancel](#)

## Application Registered Successfully

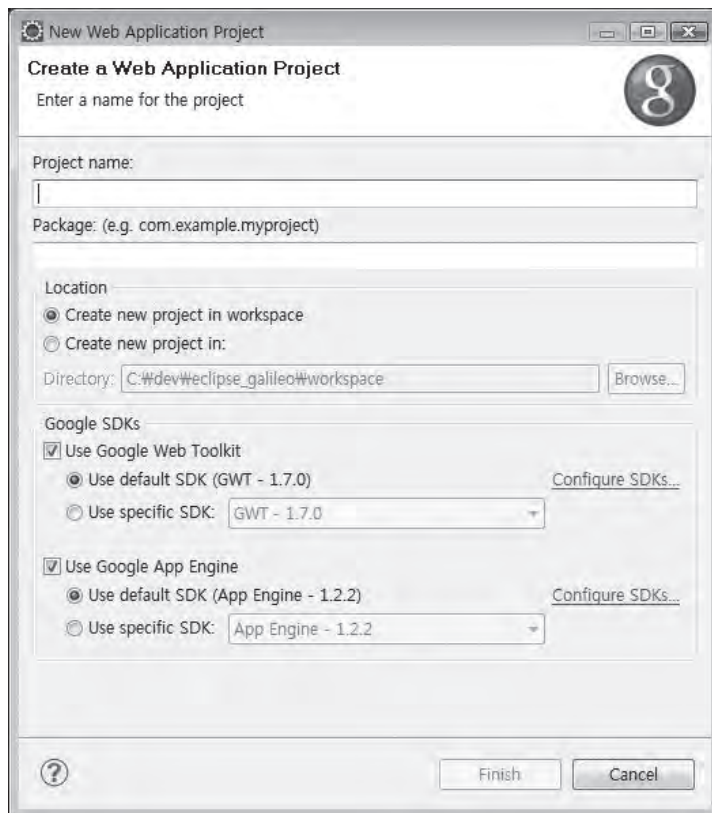
The application will use **tweetandy** as an identifier. This identifier belongs in your application's app.yaml file as well. [Learn more](#)  
Note that this identifier cannot be changed.

If you use Google authentication for your application, **tweetandy** will be displayed on Sign In pages when users access your application.

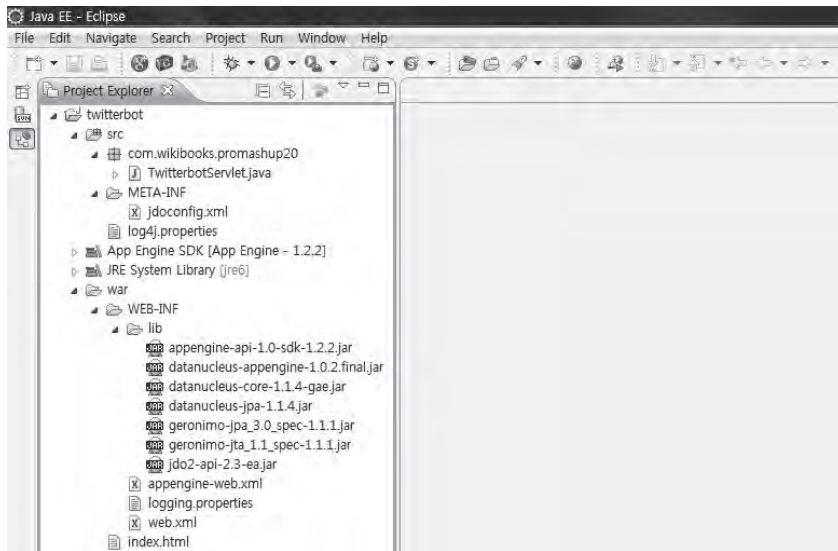
Choose an option below:

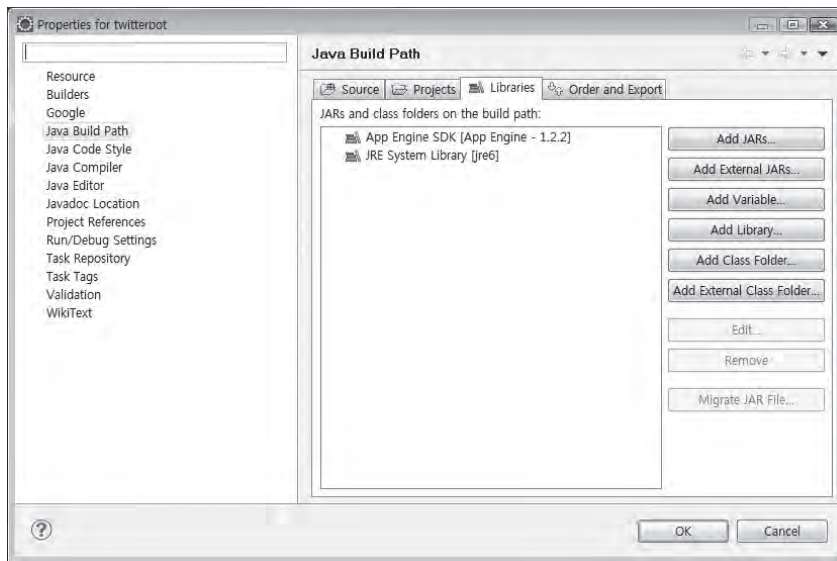
- View the [dashboard](#) for tweetandy.
- Use [appcfg](#) to upload and deploy your application code.
- Add [developers](#) to collaborate on this application.

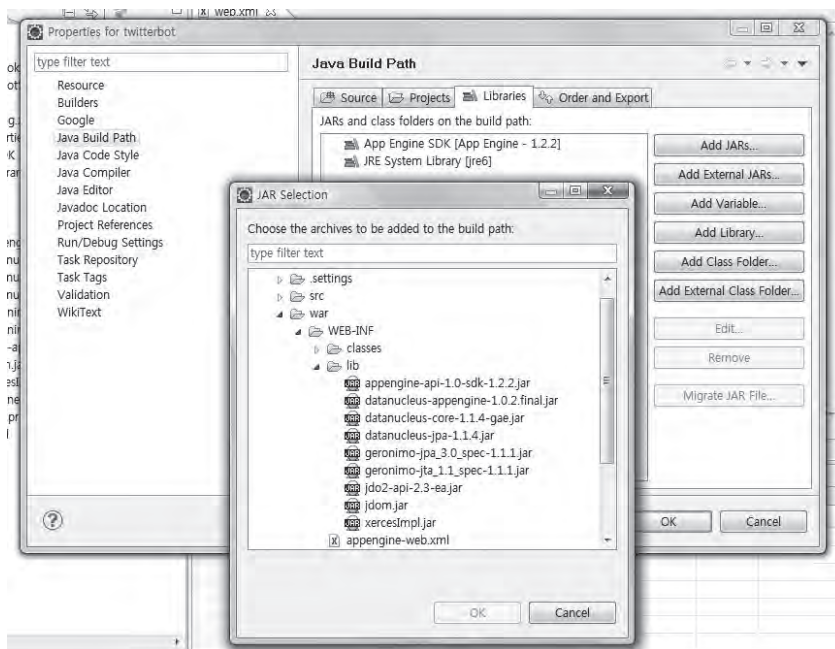
## 예제 코드 작성







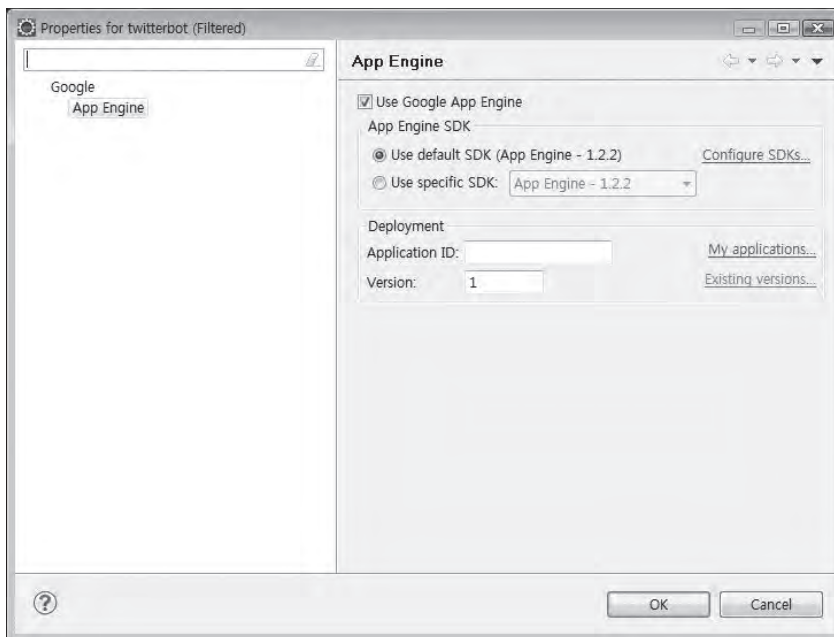
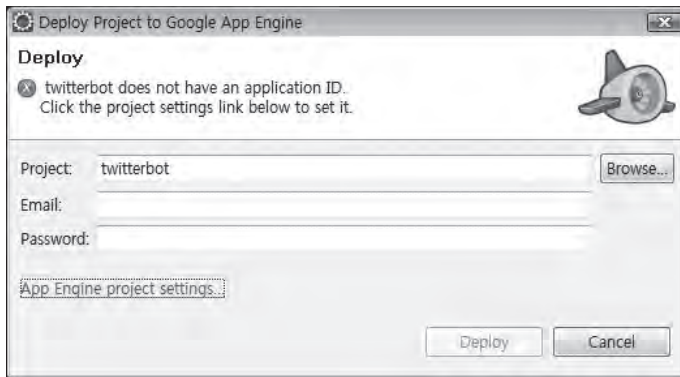












맺음말