

# OpenAPI 를 이용한 위젯 팩토리 구현 및 설계

홍윤선, 이승원, 최훈

e-mail : i@egg.pe.kr, slsw8406@naver.com, hc@cnu.ac.kr

## The Design and Implementation of Widget Factory using Open

Yoon-Sun Hong, Seung-Won Lee, Hoon Choi  
Dept. of Computer Science, Chung-Nam University

### 요 약

최근 인터넷 상에서 블로그의 이용자가 급격히 증가함에 따라 이용자들 사이에 다양한 기능을 제공하는 위젯 기반 콘텐츠에 대한 인기도 함께 증가하고 있으며, 아울러 이러한 위젯 콘텐츠에 대한 갤러리 서비스를 제공하고자 하는 다양한 연구도 활발히 진행되고 있다. 그러나 지금까지 일반 이용자가 원하는 위젯 콘텐츠를 블로그나 웹페이지에 설치하기 위해서는 각 이용자가 어느 정도의 프로그래밍 지식을 가지고 있어야 하며, 위젯의 설치 절차도 복잡하였다. 따라서, 본 논문에서는 외부에서도 사용 가능한 openAPI를 설계 및 구현함으로써 사용자의 블로그에 위젯 콘텐츠를 설치하기 위한 절차들을 간소하여 일반 사용자들이 보다 효율적으로 위젯 콘텐츠를 이용할 수 있게 하고자 하였다.

### 1. 서론

위젯 갤러리 서비스인 위젯팩토리[1]는 블로그나 웹 서비스를 대상으로 위젯 콘텐츠를 유통시키기 위해 개발되었다. 그러나 지금까지 개발된 위젯팩토리는 일반 사용자가 원하는 위젯 콘텐츠를 자신의 블로그나 웹 페이지에 설치하기 위해서 어느 정도의 프로그래밍 지식과 여러 단계의 절차를 거쳐야만 했다. 다시 말하자면, 위젯 콘텐츠를 블로그나 웹 페이지에 설치하기 위해서는 우선 위젯 갤러리 서비스에 접속하여 원하는 위젯 콘텐츠를 선택하고 설정 값을 입력한 후, 코드를 복사하거나 편집하는 일련의 과정들을 거쳐야만 하였다. 또한, 기존의 위젯팩토리는 단순히 위젯 콘텐츠를 모아 갤러리 형태로 서비스함에 따라 외부에서도 위젯 갤러리 서비스에 제공되는 위젯들을 자유롭게 볼 수 있고 코드 발급과정을 위한 프로세스를 줄이는 도구가 필요하게 되었다. 따라서, 본 논문에서는 지금까지 개발된 위젯팩토리에서 제공하는 위젯 콘텐츠를 사용자가 보다 쉽게 사용할 수 있도록 위젯 콘텐츠에 대한 정보를 외부에 제공하는 openAPI[3]를 설계 및 구현하였으며, 구현한 openAPI는 타 서비스에서도 위젯팩토리와 동일한 위젯 갤러리 서비스를 제공할 수 있다. 또한, 구현한 openAPI를 이용하여 현재 국내에서 많이 사용되고 있는 설치형 블로그 서비스인 텍스트큐브의 플러그인을 개발할 수 있는 장점을 제공한다.

### 2. 관련기술

#### 2.1. 웹 2.0

웹 2.0은 특정 신기술을 지칭하기 보다는 기존의 기술과 사용에 대한 접근 방식을 지칭하며 그 개념적인 속성은 사용자 중심, 개방성, 단순성 및 경량화, 분산화 및 참여 등의 속성을 지닌 것으로 정의된다[2].

<표 1> 웹 1.0 과 웹 2.0 의 비교

웹 1.0	웹 2.0
DoubleClick	→ Google AdSense
Ofoto	→ Flickr
Akamai	→ BitTorrent
mp3.com	→ Napster
Britannica Online	→ Wikipedia
personal websites	→ Blogging
evite	→ Upcoming.org and EVDB
domain name speculation	→ Search engine optimization
page views	→ cost per click
Screen scraping	→ web services
Publishing	→ Participation

본 연구는 지식경제부의 IT R&D 지원으로 수행되었습니다.[2008-F-048, 웨어러블 컴퍼니언 개발사업]

Content management systems	→ Wikis
Directories (taxonomy)	→ Tagging (folksonomy)
Stickiness	→ syndication

## 2.2. 위젯

‘위젯’(widget)[3] 의 사전적인 의미는 실용적인 목적으로 사용되는 작은 기계 또는 전기 장치를 말하는 것으로, 주로 새롭게 만들어졌거나 신기한 장치, 또는 이름을 알 수 없거나 생각나지 않는 소형 장치, 부품, 도구를 일컫는 단어로 사용된다. 또한 동의어로 ‘가젯’(gadget)이란 단어가 자주 사용되며 이 두 단어의 정확한 어원은 없으나 웹스터 사전에 등록된 시기를 보면 꽤 오래 전부터 사용된 단어로 추정된다[4].

<표 2> 위젯의 종류

기준	구분	특징
작동 환경	데스크탑 위젯 (desktop widget)	컴퓨터의 바탕 화면에서 작동하는 위젯
	웹 위젯 (web widget)	웹 페이지 또는 블로그에서 작동하는 위젯
	모바일 위젯 (mobile widget)	모바일 기기의 대기 화면에서 작동하는 위젯
용도	액세서리 위젯 (accessory widget)	간단한 기능을 수행하는 위젯
	애플리케이션 위젯 (application widget)	제대로된 기능을 수행하는 위젯
	정보성 위젯 (information widget)	웹 콘텐츠를 보여주기 위한 위젯
구현 방식	HTML 위젯	표준 웹 기술 기반의 위젯
	플러그인 위젯 (plugin widget)	플러그인 기반의 위젯
기능	리스트 위젯 (list widget)	RSS 피드 방식으로 콘텐츠 리스트를 제공하는 위젯
	검색 위젯 (search widget)	특정 웹 사이트의 검색 기능을 제공하는 위젯
	뷰어 위젯 (viewer widget)	이미지 파일을 보여주는 위젯
	플레이어 위젯 (player widget)	음악이나 동영상을 재생하는 위젯
	광고 위젯 (advertising widget)	광고 내용을 전달하는 위젯

## 2.3. OpenAPI

Open API[4]는 ‘자사의 데이터 또는 서비스를 외부에서 사용할 수 있도록 API 를 공개한 것’을 말한다. 위키피디아에서는 API 를 ‘응용 프로그램에서 사용할 수 있도록 운영체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있도록 만든 인터페이스’ 라고 정의하고 있다. 즉, 원래는 운영체제나 언어가 제공하는 기능을 제어할 수 있는 인터페이스였으나 웹 서비스에서는 특정 서비스를 사용할 수 있도록 제공하는 인터페이스를 API 라 지칭하며, 이를 외부에서 사용할 수 있도록 공개한 것이 OpenAPI 이다.

일반적으로 OpenAPI 는 SOAP 과 같이 복잡한 프로토콜보다는 XML-RPC, REST 등의 프로토콜을 사용하여 API 공개를 통한 개방과 참여를 유도하는 특성을 가지며, 이를 통해 여러 서비스 혹은 데이터를 이용하여 별도의 웹페이지나 서비스를 제공할 수 있다[5].

## 2.4. 텍스트큐브

텍스트큐브(textcube)는 오픈 소스 블로그 소프트웨어인 ‘태터툴즈(tattertools)’를 기반으로 만들어진 온라인 출판 플랫폼으로서, 개인 또는 기업이 블로깅 서비스를 구축할 수 있는 도구이다. 확장이 용이한 플러그인 구조와 테마 시스템 및 다국어 지원을 제공하며, 외부 개발자가 자유롭게 이용할 수 있도록 관련 문서와 완성된 플러그인을 등록 및 배포 가능한 갤러리 서비스를 제공한다.

플러그인 서비스를 제공하기 위해서는 일반적으로 다음과 같은 두 개의 파일이 필요하다.

- ✓ **index.xml**: 텍스트큐브 코어와 플러그인의 상관관계 및 동작 위치 등을 결정
- ✓ **index.php**: 실제 실행되는 플러그인 코드

텍스트큐브의 종류는 크게 사이드바에 추가되는 위젯을 생성할 수 있는 ‘사이드바 플러그인’, 블로그 주소를 입력했을 때 나타나는 페이지로 사용되거나 별도 페이지를 작성하는 ‘포지 플러그인’ 및 관리자 화면의 센터에 출력되는 위젯을 만드는 ‘센터 플러그인’이 있으며, 공통적인 기능으로는 사용자의 선택에 따라 다른 동작을 수행할 수 있도록 환경설정패널 및 환경 저장을 위한 API 를 제공한다[6].

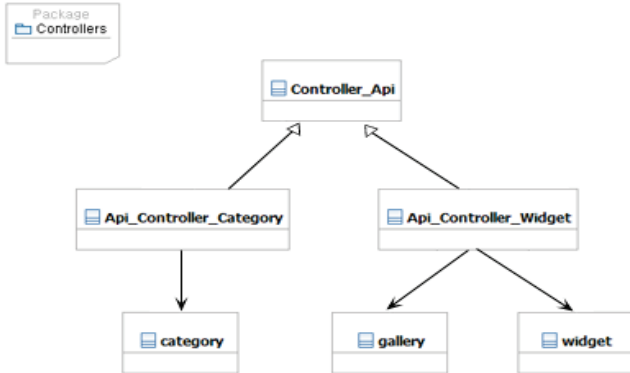
## 3. 설계 및 구현

### 3.1. API 기본 구조

#### 3.1.1. 컨트롤러

사용자로부터 전달된 값을 기반으로 관련 모델(Model)을 이용하여 필요한 데이터를 전달 받아 이를 가공 후 결과물을 출력하는 패키지이다. 일반적인

MVC 패턴의 컨트롤러의 경우, 모델에서 전달 받은 데이터에 대한 비즈니스 로직(logic)만을 담당하는 경우가 대부분이나 별도로 HTML 에 대한 가공 없이 일정한 형식의 데이터만을 출력하기 때문에 뷰 패키지 없이 직접 데이터를 출력한다.



(그림 1) Controllers Package's Class Diagram

Controller\_Api, Api\_Controller\_Category 및 Api\_Controller\_Widget 은 일반적인 MVC 패턴에서 사용되는 Controller 와는 달리 사용자의 요청을 처리하기 위한 액션(action)이 제외되어 있는 클래스이다. 액션은 별도의 category, gallery 및 widget 클래스에 포함되지만, 개발 언어의 특성 및 프레임워크의 구조적 특성을 고려하여 실제로는 각각의 폴더로 구성되어 요청을 처리하는 파일 단위의 액션으로 구분되어 있다.

### 3.1.1.1. Controller\_API

API 에서 공통으로 사용되는 기본적인 모듈을 모아놓은 클래스로서, 이를 상속받아 각각의 서비스에 맞게 재정의한다.

<표 3> Controller\_Api's Operation List

Name	Return	Explanation
api_get	String	item 을 output type 에 맞게 변환된 string 반환
api_list	String	item 을 output type 에 맞게 변환된 string 반환
apikey	String	apikey 를 반환
auth_apikey	Boolean	인증 여부 반환. 필요 시 인증 작업 수행
cache_get	String	item 을 output type 에 맞게 변환하여 지정된 cachename 으로 cachefile 을 생성하여 경로를 반환
cache_list	String	items 을 output type 에 맞게 변환하여 지정된 cachename 으로 cachefile 을 생성하여 경로를 반환.
callback	String	지정된 callback 값 반환.
filter	Array	미리 저장된 값만 선별하여 반환.

### 3.1.1.2. Api\_Controller\_Category

카테고리 정보에 맞춰 Controller\_API 를 재정의한 클래스이다.

<표 4> gallery's Action List

Name	return	Explanation
category	void	지정된 카테고리 해당하는 위젯을 지정된 페이지에 해당하는 위젯 정보를 반환.
list	void	모든 위젯을 대상으로 지정된 페이지에 해당하는 위젯 정보를 반환.
widget	void	지정된 위젯 정보 반환

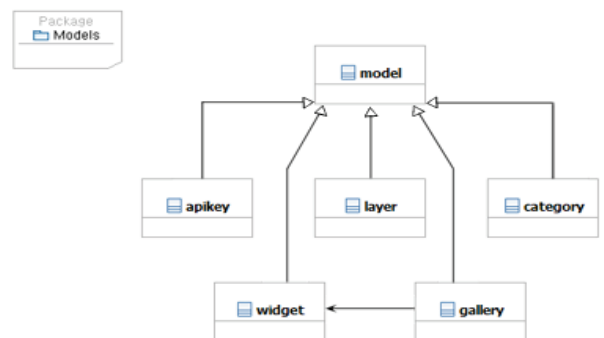
### 3.1.1.3. widget

위젯의 생성, 정보 수정 및 생성된 위젯 목록 등, 위젯과 관련된 요청을 처리하는 액션들의 모음으로서 클래스 다이어그램으로 표현되었으나 실제 해당 operation 은 파일 단위의 서버스크립트 형태이다.

<표 5> widget's Action List

Name	return	Explanation
create	void	지정된 위젯 생성
delete	void	지정된 위젯코드에 해당하는 위젯 삭제.
get	void	지정된 위젯 코드에 해당하는 생성된 위젯 정보와 설정 값 반환.
list	void	지정된 apikey 에 해당하는 위젯 목록 또는 지정된 위젯 코드에 해당하는 위젯 정보 반환.
update	void	지정된 위젯코드에 해당하는 위젯의 설정 값 변경.

### 3.1.2. 모델



(그림 2) Models Package's Class Diagram

컨트롤러에 의해 호출되며 프레임워크에서 제공하는 라이브러리를 이용해 데이터베이스에 접근한 후,

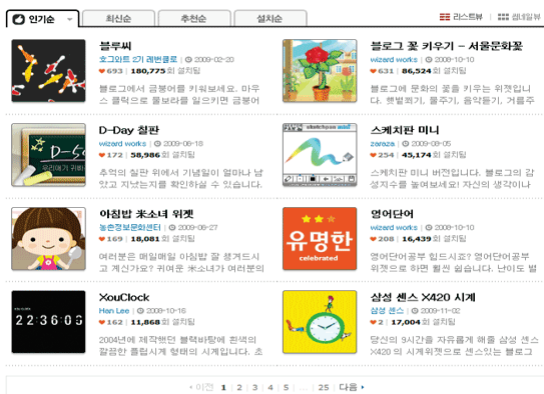
데이터에 대한 입출력을 담당하는 패키지이다. 포함하는 클래스로는 apikey, layer, category, widget 및 gallery 가 있다. <표 6>는 gallery 클래스의 특성을 나타낸다.

<표 6> gallery's Operation List

Name	return	Explanation
get	widget	새로운 위젯 반환
getList	array	지정된 apikey 를 통해 사용 가능한 위젯 목록 반환.
getListByCategory	array	지정된 apikey 를 통해 사용 가능한 위젯 중 선택된 카테고리에 해당하는 위젯 목록 반환.
getTotalcount	int	지정된 apikey 를 통해 사용 가능한 위젯의 개수 반환.
getTotalcountByCategory	int	지정된 apikey 를 통해 사용 가능한 위젯 중 선택된 카테고리에 해당하는 위젯 개수 반환.

4. 구현결과

다음 OpenAPI 를 이용하여 구현한 위젯 갤러리 서비스를 그림 3 과 같이 볼 수 있다. 카테고리별, 인기순, 최신순, 추천순, 설치순으로 정렬 기능이 제공되고, 사용자의 스타일에 따라 원하는 위젯을 고르면 그림 4 와 같이 볼 수 있다. 그림 4 처럼 사용자는 자신의 블로그 주소만 알면 미리보기 기능이 실행이 되면서 위치와 크기를 자유자재로 변경 할 수 있고 바로 자신의 블로그에 적용 시킬 수 있다.



(그림 3) 위젯팩토리 - 위젯 갤러리 페이지 UI



(그림 4) 위젯팩토리 - 위젯 상세 페이지

5. 결론

본 논문에서는 블로그 플러그인 뿐만 아니라 외부 제휴 혹은 제작 의뢰를 통해 위젯팩토리가 아닌 외부에서 만들어진 위젯 퍼가기 페이지에서도 구현한 openAPI 인 갤러리 API 를 이용하여 위젯 콘텐츠를 쉽고 간결하게 이용할 수 있게 하였다. 이를 통해 위젯팩토리 입장에서 위젯 콘텐츠의 유통량과 개발 편의성이 증가될 수 있으며, 제휴 업체 측면에서는 자신들의 서비스 내에서도 위젯을 퍼갈 수 있도록 하여 사용자 하여금 거부감 없이 위젯을 생성하거나 효과적인 이벤트를 진행할 수 있다.

참고 문헌

- [1] 위젯팩토리, <http://wzdfactory.com/gallery/all>.
- [2] O'ReillyTim, "What is Web 2.0-Design patterns and business models for the next generation of software", 2005/09/30.
- [3] Widgets 1.0, "W3C working Draft.", <http://www.w3.org/TR/2008/WD-widgets-20080414/>.
- [4] 노주환, "웹 패러다임을 바꾸는 위젯", 멘토르 2008, p.4.
- [5] 네이버 Open API, <http://openapi.naver.com>.
- [6] textcube.com, "플러그인 제작하기." <http://www.textcube.org/manual/entry/PluginIntroduction>.