Windows Live Service Open API 소개 – 개발자

매쉬업코리아2008 – 한국 Microsoft

한국 Microsoft 김대우

<http://www.uxkorea.net>

지금 시작하세요. Windows Live Service Open API 소개

<http://dev.live.com>

**Windows Live Service 소개**

Windows Live Service는 글로벌로 확장 가능한 플랫폼 요소로, 이 API들을 이용해 다양한 웹응용프로그램의 확장과 새로운 응용프로그램을 개발 가능하게 하는 기술입니다. 현재 웹에서 사용 가능한 Windows Live Service는 다음과 같습니다.



그림1. Windows Live 서비스

Windows Live 서비스들은 독립적인 응용프로그램을 그 목표로 하지 않습니다. 현재 대부분의 Windows Live Service가 개발자에게 사용 가능한 형태의 API로 제공되고 있으며, 개발자는 이 API를 이용해 다양한 형태의 커스텀 서비스 및 글로벌로 확장 가능한 응용프로그램 제작이 가능해 집니다.

이러한 다양한 서비스는 현재 <http://www.live.com> 에서 이용해 보실 수 있습니다.

**Windows Live Service Open API를 이용한 매쉬업**



그림2. 개발자가 사용 가능한 Windows Live Service Open API 적용 가능 서비스

현재 Windows Live Service Open API 리스트는 계속해서 확장되어 가고 있으며, Windows Live 서비스의 API를 이용해 개발자는 매쉬업을 진행하실 수 있습니다.

Windows Live Service Open API 개발자 웹사이트 : <http://dev.live.com/>

Windows Live Service가 제공하는 API들의 리스트를 위의 <http://dev.live.com> 에서 확인 가능하며 매쉬업을 위한 API 참조로 시작하실 수 있습니다. 또한, 다양한 API들에 대한 새로운 소식과 개발자 Forum(영문)을 이용해 여러 개발자와 커뮤니케이션도 가능합니다.

다음으로 Windows Live Service Open API에 대해서 소개해 드리도록 하겠습니다.

**1. 풍부한 미디어 기능 - Silverlight Streaming 서비스**

Windows Live Service가 제공하는 Silverlight Streaming 서비스를 이용하시면 개발자와 디자이너가 쉽게 확장 가능한 미디어를 Silverlight 기술로 전달 가능하게 합니다.

Silverlight 기술은 크로스플랫폼, 크로스 브라우져, 기술로 풍부한 미디어 경험과 인터렉티브 응용프로그램을 제작할 수 있는 기술입니다. 만들어진 Silverlight 미디어를 Live서비스를 통해 무료로 서비스 할 수 있습니다.

**개발 절차**

1) <http://dev.live.com/silverlight/> 접속

2) Windows Live 계정을 이용해 <https://silverlight.live.com/> 에 접속, Manage Account 접속

3) Account ID 및 Account Key를 발급

4) Silverlight Streaming SDK를 이용해 개발 진행.

**개발방식**

Silverlight Streaming은 REST API를 이용해 개발 가능합니다. 위에서 발급받은 Account ID와 Key를 통해 서비스 이용이 가능하며, Silverlight 미디어 업로드 및 리스트, 수정, 삭제가 가능합니다.

**샘플코드 – C#을 이용한 인증**

|  |
| --- |
| using System;  using System.Net;  using System.IO;  namespace SilverlightServicesApi  {  class Program  {  static void Main(string[] args)  {  // accountId = Silverlight Streaming Account ID  HttpWebRequest req =  (HttpWebRequest)HttpWebRequest.Create  ("https://silverlight.services.live.com/accountId");  byte[] userPass = Encoding.Default.GetBytes(accountId + ":" + accountsecret);  string basic = "Basic " + Convert.ToBase64String(userPass);  req.Headers["Authorization"] = basic;  HttpWebResponse resp = (HttpWebResponse)req.GetResponse();  Stream strm = resp.GetResponseStream();  StreamReader rdr = new StreamReader(strm);  string xmlResponse = rdr.ReadToEnd();  }  }  } |

**2. 개인화 가능한 검색 서비스 – Windows Live Search**

Live Search의 엔진을 그대로 이용해 SOAP 메세징으로 자사나 커뮤니티, 블로그 검색에 이용 가능합니다. 검색 상자를 이용한 방법과 Live Search API를 이용하는 방식이 있으며 여기서는 API를 이용하는 방식을 소개해 드리도록 하겠습니다.

**개발절차**

1) <http://dev.live.com/livesearch/> 접속

2) Getting Started with the Live Search Web Service API 링크클릭

3) Live SDK 정보 및 Live Search 웹 서비스 WSDL(웹 참조) 정보, Application ID 생성을 확인

4) Create and Manage Application IDs 과정을 통해 Live API를 이용하기 위한 Application Name과 Application ID를 생성

5) <http://dev.live.com/livesearch/sdk/> 등 개발하기 위한 SDK를 확인

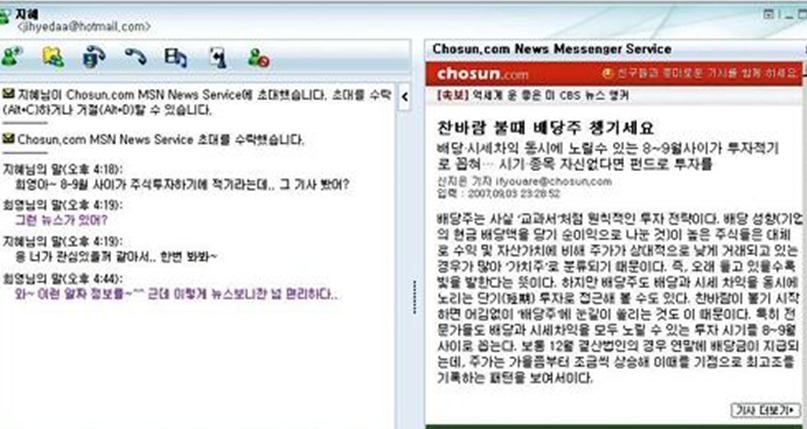
**샘플코드 – C#, Visual Studio를 이용해 개발**

|  |
| --- |
| // Visual Studio에서 웹참조를 추가합니다.  // 클래스 : http://soap.search.msn.com/webservices.asmx?wsdl  MSNSearchService s = new MSNSearchService();  SearchRequest searchRequest = new SearchRequest();  SourceRequest[] sr = new SourceRequest[1];  sr[0] = new SourceRequest();  sr[0].Source = SourceType.Web;  sr[0].ResultFields = ResultFieldMask.Description | ResultFieldMask.DisplayUrl | ResultFieldMask.Title | ResultFieldMask.Url;  searchRequest.Query = "Seattle Mariners";  searchRequest.Requests = sr;  // Application ID는 반드시 본인의 ID를 사용합니다.  searchRequest.AppID = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";  searchRequest.CultureInfo = "ko-KR"; //로케일을 필요하다면 변경합니다.(미국: en-US)  SearchResponse searchResponse = s.Search(searchRequest);  foreach (Result sourceResult in searchResponse.Responses[0].Results)  {  string title, description, url;  title = (!String.IsNullOrEmpty(sourceResult.Title)) ? sourceResult.Title : "";  description = (!String.IsNullOrEmpty(sourceResult.Description)) ? sourceResult.Description : "";  url = (!String.IsNullOrEmpty(sourceResult.Url)) ? sourceResult.Url : "";  } |

SOAP 방식 직접 요청 역시 가능합니다.

**3. Windows Live Messenger를 통한 서비스 확장 – Messenger API**

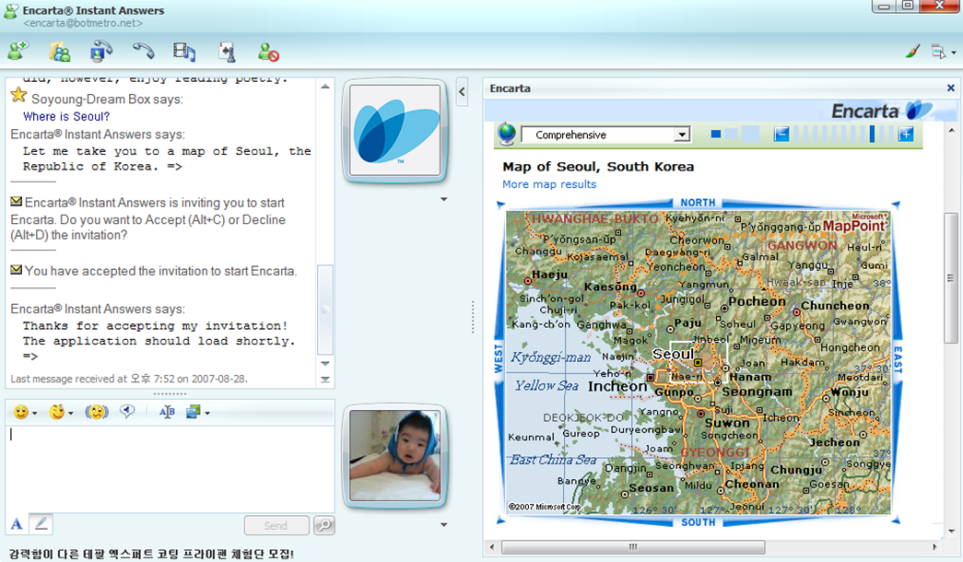
메신저 역시 Windows Live 플랫폼 서비스 입니다. 다양한 형태의 확장 기능을 사용 가능하며, Activity 서비스를 이용해 기능을 다양하게 이용 가능합니다.



조선일보의 Activity 서비스 – 웹 페이지에서 Messenger 친구와 조선일보의 기사를 공유 가능



웹 페이지의 영상을 메신저를 이용해 공유 가능



엔카르타 백과 사전 자동 응답 Agent 서비스 - [encarta@botmetro.net](mailto:encarta@botmetro.net)

자동 응답 Agent 기능을 이용한 서비스 예시 - [wilma@live.com](mailto:wilma@live.com)

이 서비스들은 메신저 확장 기능을 이용해 개발 가능합니다. 특히, 미디어 사이트의 경우 자사의 다양한 웹 콘텐트에 “메신저로 친구와 공유” 기능을 이용해 자사의 서비스를 확장하는 새로운 사용자 접속 채널을 적용 가능합니다.

Messenger의 기능을 확장하는 이 기능은 현재 조선일보(<http://www.chosun.com/>), MSNBC(<http://www.msnbc.com/>) 등에서 확인 가능합니다.

**개발방안**

1) 개발 절차는 <http://msdn2.microsoft.com/ko-kr/library/aa905675.aspx> 정보를 확인 가능합니다.

2) Messenger Activity 및 Add-in 서비스를 구현하시기 위해서는 Microsoft와 협력 및 계약 절차가 필요하며 [sylee@microsoft.com](mailto:sylee@microsoft.com) 에 문의 하시면 서비스에 대한 안내를 받으실 수 있습니다.

3) 서비스 계약이 완료되시면, Application ID 또는 Add-in 서비스 구현을 위한 SDK를 받으실 수 있습니다.

4) 개발 언어는 JavaScript 및 다양한 언어를 이용해 개발하실 수 있습니다.

**4. 정보의 즉각적인 알림 Windows Live Alert 서비스**

Alert 서비스는 특정 이벤트 등을 Messenger 토스트(화면 우측 하단의 Messenger 알림 창)을 통해 전달할 수 있는 서비스 입니다. 즉, 블로그나 커뮤니티의 게시판에 새로운 글이 올라오면 자동으로 알림을 받거나, 내가 문의한 글에 답변 글이나 댓글이 추가되면 알림을 받을 수 있는 형태의 전달 서비스 입니다.



이렇게 커뮤니티의 특정 게시판에 새글이 올라올 경우 Alert 토스트 등을 받으실 수 있습니다.

**개발방안**

1. Alert 서비스는 SOAP 방식을 이용하며 <http://msdn2.microsoft.com/ko-kr/library/bb259752.aspx> 입니다.

2. Alert 서비스를 구현하시기 위해서는 Microsoft와 협력 및 계약 절차가 필요하며 [sylee@microsoft.com](mailto:sylee@microsoft.com) 에 문의 하시면 서비스에 대한 안내를 받으실 수 있습니다.

3. 서비스 계약이 완료되시면, 웹 서비스 주소와 Application ID를 받으실 수 있습니다.

4. 개발 언어는 SOAP이므로 다양한 언어를 이용해 개발하실 수 있습니다.

**샘플코드 – SOAP 메세징**

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  <soapenv:Body>  <ns1:InitiateSignup soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:ns1="signup.alerts.msn.com">  <h xsi:type="ns1:RecServicesHeader">  <messageID xsi:type="xsd:string">2003-03-26.testInitiateSignup.1</messageID>  <timestamp xsi:type="xsd:string">2003-03-26T04:36:25-08:00</timestamp>  <version xsi:type="xsd:string">1.0</version>  </h>  <ID xsi:type="ns1:RecServicesIdentification">  <PINID xsi:type="xsd:int">00</PINID>  <PW xsi:type="xsd:string">password</PW>  </ID>  <partnerUID xsi:type="xsd:string">junit testInitiateSignup</partnerUID>  <returnURL xsi:type="xsd:string">http://alerts.msn.com</returnURL>  <transportType xsi:type="xsd:string">MSNA</transportType>  </ns1:InitiateSignup>  </soapenv:Body>  </soapenv:Envelope> |

몇 가지 Windows Live Service에 대해서 소개해 드렸습니다.

이외에도 Virtual Earth 및 다양한 Open API를 이용하실 수 있으며 이 정보들은 <http://dev.live.com> 에서 확인 가능합니다.

감사합니다.