REST 웹서비스의 기술 언어 활용에 관한 연구(WADL)

이재만*
*포스코 ICT 정보제어기술연구소 SW 융합기술팀
e-mail : jaemanii.lee@poscoict.com

A Study on Description Language Usage for REST WebService(WADL)

Jae-Man Lee*
*SW Convergence Technology Team, POSCO ICT

요 약

인터넷 특히 웹의 등장과 더불어 응용 애플리케이션 환경도 인터넷을 중심으로 한 분산 네트워크 시스템이 자연스럽게 주를 이루고 있다. 이러한 환경속에서 서로 다른 기종간의 통신을 위하여 등장한 기술이 웹서비스인데 그 발전 과정에서 SOAP 기반의 웹서비스와 RESTful 기반의 웹서비스 형태로 제공되고 있다. 최근에는 RESTful 기반의 웹서비스가 가벼우면서 접근방법의 용이함으로 인해 많은 지지를 받으며 SOAP 에 비해 각광을 받고 있다. 하지만, REST 는 SOAP 에 비해 서비스에 대한 명세를 알릴 수 있는 표준의 부재와 관리가 어려운 단점이 있다. 그 중에서 서비스 명세에 대한 표준으로 SOAP 에서는 WSDL 이 존재하고, REST 에서는 WADL 이 존재한다. 본 논문에서는 WADL 를 실무에서 활용할 수 있도록 스펙에 따라 구현해보고 그 방법을 제시한다.

1. 서론

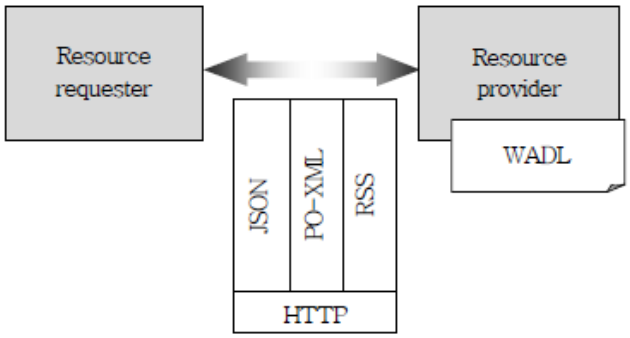
1989 년 Tim Berners-Lee 에 의해 개발된 웹은 HTTP, HTML, URL 등의 표준 기술을 사용하여 분산환경에서 정보자원들의 공유와 호환을 가능하게 해주었다. 이런한 웹 환경도 진화를 거듭하면서 개방,공유,참여의 특징을 갖는 웹 2.0 으로 표출되었고, 최근에는 지능형의 시맨틱 웹(Semantic Web), 모바일 웹(Mobile Web) 등으로 변화하고 있다. 하지만 최근의 이런 트렌드 이전에 1996 년 4 월 세계적인 IT 컨설팅업체인 Gartner 에 의해 처음 소개된 서비스 지향 아키텍처(Service-Oriented Architecture, SOA) 는 서비스를 공유하고 재사용할 수 있도록 정보시스템을 구축하는 소프트웨어 설계 방법론이다. 이러한 기술적 개념의 등장과 네트웍 중심의 애플리케이션 구조의 흐름속에서 자연스럽게 SOA 가 부각이 되었고, 이에 대한 구현 모델로써 CORBA, RMI, DCOM, XML-RPC 등에서부터 SOAP(Simple Obect Access Protocol) 로 이어져 왔다. 하지만, SOAP 은 기술적인 완성도의 부족과 소프트웨어 벤더들 간의 협력 부재, 그리고 지나치게 무거운 (heavy cost) 데이터 처리와 구현 방법의 복잡함으로 인해 공유, 개방의 플랫폼으로써는 어려운 면이 많았던 것이 사실이다.

최근 몇 년 사이에 많이 활용되고 있는 REST(REpresentational State Transfer) 는 이런 SOAP 의 단점을 보완하고자 등장한 아키텍처이다. REST 는 HTTP 를 통해 XML 또는 JSON 데이터를 교환하는 비교적 단순하고 가벼운(lightweight) 프로그래밍 모델 방식이다. 아마존 같은 경우 전체 오픈 API 의 95% 를 단순한 REST 스타일을 사용하여 공개하고 있다. 구글의 경우에도 2006 년 12 월 초에 SOAP 기반의 오픈 API 서비스를 중단하고, REST 스타일의 자바스크립트 기반의 AJAX Search API 를 대안으로 채택하였다. 하지만, REST 라고 하여 반드시 SOAP 보다 모든 면에서 우위에 있는 기술 또는 아키텍처라고 할 수는 없다. REST 접근하기 쉽고 웹의 장점을 고스란히 활용한다는 측면이 있지만, SOAP 의 WSDL(Web Service Description Language) 로 대표되는 제공하고자 하는 서비스의 명세를 기술하고 프로토콜화하는 명백한 구조를 지니고 있고, 어떤 서비스를 제공하는 지 검색하고 활용할 수 있는 Repository 인 UDDI 를 제시하고 있다. 이중 RESTful 구현 방식에서 SOAP 의 WSDL 이 없는 부분은 설계도 없이 집을 짓는 것과 유사하다고 할 수 있다. 그래서 또 다시 REST 의 이런 단점을 보완하고자 위해 2006 년에 Sun 에서 WADL(Web Application Description Language) 를 발표하였다. WADL 은 REST 방식의 웹서비스가 갖고 있던 서비스 명세에 대한 표준을 제시하기 위한 방법을 기술하기 위한 언어이다. 기존의 인터넷 기업들인 아마존, 구글, 야후, 플리커 등에서 REST 스타일의 API 의 기술(description) 을 HTML 텍스트 문서로 대신해 왔던 셈이다. 하지만, WADL 역시 POST, PUT, GET, DELETE 로 표현되는 Action 의 단순함 그리고 응답결과를 표현하는 Message 의 Format 을 기술하고 있지 않아 해외는 물론 국내에서 많이 활용되지 못하고 있다. 따라서, 본 논문에서는 REST 기술 언어의 단점

을 보완하는 방법을 소개하고자 한다.

2. REST ARCHITECTURE MODEL

REST 는 웹의 창시자 중 한 사람인 Roy Fielding 이 그의 박사 학위 논문에서 현재의 웹 아키텍처가 웹의 본래 우서성을 활용하지 못하므로 웹의 장점을 최대한 활용할 수 있는 네트워크 기반의 아키텍처를 제안했는데 이것이 REST 이다. REST 는 부수적인 레이어나 세션 관리를 추가하지 않고 HTTP 프로토콜을 활용하여 데이터를 교환하는 아키텍처이다. 여기에서 Resource 란 고유의 URI 로 대표되며, HTTP 의 METHOD 인 GET/PUT/POST/DELETE/HEAD/OPTION 로 행위를 정의한다. 그리고 WADL 은 웹서비스에서 제공하는 서비스의 명세를 정의하고 있다. 이처럼 RESTful 웹서비스는 리소스 중심의 표현 및 전달 방식의 특성으로 인하여 리소스 기반 아키텍처(ROA) 라고 한다.



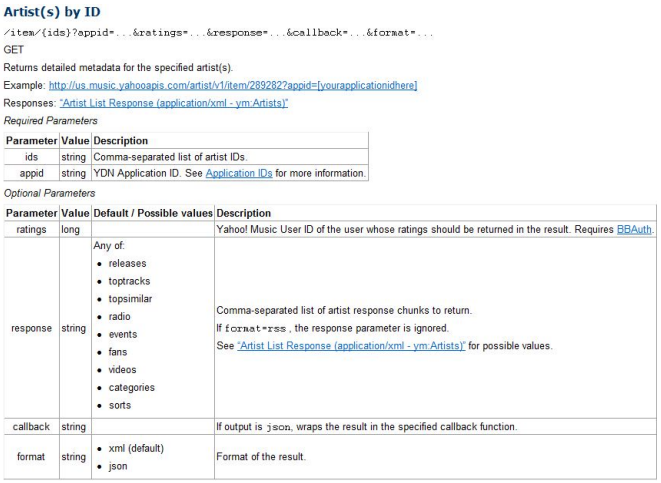
(그림 1) REST 아키텍처

그렇다면, REST 아키텍처를 이용하여 회원 정보 조회에 대해 구현해보면 다음과 같다.

- Action : 특정 회원 정보를 조회를 요청한다.
- URI : http://xxx.xxx.com/user/view/gildong
- Method : GET
- Response :

```
<user>
    <id><![CDATA[ gildong ]]></id>
    <name><![CDATA[ 고길동 ]]></name>
    <gender>M</gender>
    <age>33</age>
</user>
```

REST 의 최대 장점은 단순한 구현 방식을 지향하고 있는 점이다. 구글, 야후, 페이스북 등 인터넷 기업들이 오픈 API 를 REST 방식으로 제공하고 있는 점도 바로 이때문이다. 하지만 이런 장점에도 불구하고 REST 는 SOAP 의 WSDL 처럼 서비스 명세에 대해 명확한 표준을 제시하지 못한다. 따라서, 오픈 API 를 제공하고 있는 기업들도 텍스트 방식의 문서를 통해 배포하고 있으며 SOAP 과 같이 자동으로 stub 을 생성할 수도 없다. 표준이 없다는 부분은 개발에 있어서 관리가 어렵고, 큰 혼란이 발생할 수도 있다. 이로인해 REST 서비스에 대한 기계적인 처리가 가능하고 명세를 표현할 수 있는 기술 언어의 필요성이 증가하여 WADL 이 등장하였다.



Artist(s) by ID

/item/{ids}?appid=...&ratings=...&response=...&callback=...&format=...

GET

Returns detailed metadata for the specified artist(s).

Example: http://us.music.yahooapis.com/artist/v1/item/289282?appid=[yourapplicationidhere]

Responses: "Artist List Response (application/xml - ym:Artists)"

Required Parameters

Parameter	Value	Description
ids	string	Comma-separated list of artist IDs.
appid	string	YDN Application ID. See Application IDs for more information.

Optional Parameters

Parameter	Value	Default / Possible values	Description
ratings	long		Yahoo! Music User ID of the user whose ratings should be returned in the result. Requires BBAuth.
response	string	Any of: • releases • toptracks • topsimilar • radio • events • fans • videos • categories • sorts	Comma-separated list of artist response chunks to return. If `format=rss`, the response parameter is ignored. See "Artist List Response (application/xml - ym:Artists)" for possible values.
callback	string		If output is `json`, wraps the result in the specified callback function.
format	string	• xml (default) • json	Format of the result.

(그림 2) REST 기반의 Yahoo 오픈 API 설명 문서

3. WADL(Web Application Description Language)

WADL 은 2006 년 Sun 에서 발표한 HTTP 기반의 웹 애플리케이션을 기술하기 위해 설계된 언어이다. REST 는 Web 아키텍처를 충실히 따르고 있기 때문에 문서 방식의 웹 서비스에 대한 명세를 WADL 로 표현이 가능하다. 기계가 처리 가능한 포맷으로 기술되기위한 WADL 은 다음의 특징들을 갖는다.

- URI 로 표현되는 리소스의 집합
- 리소스들간의 관계를 기술
- 자원에 적용될 HTTP METHOD
- 지원되는 MIME type 과 데이터 포맷

이런 WADL 의 특징을 활용하여 애플리케이션 모델링과 시각화 및 자동으로 stub 과 skeleton 의 생성이 가능하도록 할 수 있다.
다음은 Amazon 상품 검색 애플리케이션의 WADL 예이다.

```
<application xmlns="http://research.sun.com/wadl/2006/07"
    xmlns:aws="http://webservices.amazon.com/AWSECom
merceService/2005-07-26"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<grammars>
   <include href="AWSECommerceService.xsd"/>
</grammars>

 <resources
base="http://webservices.amazon.com/onca/">
   <resource path="xml">
     <method href="#ItemSearch"/>
   </resource>
 </resources>

 <method name="GET" id="ItemSearch">
```

```
    <request>
      <param name="Service"
style="query"  fixed="AWSECommerceService"/>
      <param name="Version" style="query" fixed="2005-07-
26"/>
      <param name="Operation" style="query"
fixed="ItemSearch"/>
       <param name="SubscriptionId" style="query"
type="xsd:string" required="true"/>
      <param name="SearchIndex" style="query"
type="aws:SearchIndexType" required="true">
         <option value="Books"/>
         <option value="DVD"/>
         <option value="Music"/>
      </param>
      <param name="Keywords" style="query"
type="aws:KeywordList" required="true"/>
       <param name="ResponseGroup" style="query"
type="aws:ResponseGroupType" repeating="true">
          <option value="Small"/>
          <option value="Medium"/>
          <option value="Large"/>
          <option value="Images"/>
      </param>
    </request>
    <response>
      <representation mediaType="application/xml"
element="aws:ItemSearchResponse"/>
    </response>
 </method>
 </application>
```

이 WADL 에서 AWSECommerceService.xsd 는 xml 스키마에 대한 정의를 갖고 있고, 검색 서비를 제공하는 기본 URI 는 http://webservices.amazon.com/onca/ 이다. 이 Resource 에 대한 path 는 xml 이고, method 링크는 ItemSearch 이다. 따라서, Amazon 검색 서비스를 이용하기 위해서 클라이언트에서는 아래와 같은 REST 서비스에 대한 문서와 일치되는 Description 으로 볼수 있다.

- Action : 아마존에서 특정 조건의 상품을 검색한다.
- URI : http://xxx.xxx.com/user/view/gildong
- Method : GET
- Parameter : Service={Service}&Version={Version}&Operation={Operation}&SearchIndex={SearchIndex}
- Response : aws:ItemSearchResponse

여기에서 응답은 aws 네임스페이스의 ItemSearchResponse 로 받도록 정의가 되어있는데, 응답에 대한 스펙은 다음의 URL 을 통해 제공이 된다.

http://webservices.amazon.com/AWSECommerceService/2020-07-26/AWSECommerceService.xsd

아마존에서 사용하는 응답 데이터 포맷이 정의가 되어 있는데, 이 중 ItemSearchResponse 관련 부분을 보면 다음과 같다.

```
<xs:element name="ItemSearchResponse">
…
<xs:complexType>
        <xs:sequence>
          <xs:element ref="tns:OperationRequest"
   minOccurs="0" />
          <xs:element ref="tns:Items" minOccurs="0"
   maxOccurs="unbounded" />
        </xs:sequence>
      </xs:complexType>
</xs:element>
…
   <xs:element name="Items">
     <xs:complexType>
       <xs:sequence>
                  <xs:element ref="tns:Request"
   minOccurs="0" />
                  <xs:element ref="tns:CorrectedQuery"
   minOccurs="0" />
                  <xs:element name="TotalResults"
   type="xs:nonNegativeInteger" minOccurs="0" />
                  <xs:element name="TotalPages"
    type="xs:nonNegativeInteger" minOccurs="0" />
                  <xs:element
ref="tns:SearchResultsMap" minOccurs="0" />
                  <xs:element ref="tns:Item"
    minOccurs="0" maxOccurs="unbounded" />
                  <xs:element ref="tns:SearchBinSets"
    minOccurs="0" />
          </xs:sequence>
      </xs:complexType>
</xs:element>
```

응답에 대한 데이터 포맷을 제공함으로써 마샬링과 언마샬링을 처리할 수 있는 프로그래밍 코드를 자동으로 생성하는 것이 가능하다. 그래서 이러한 WADL 처리를 위한 라이브러리를 http://wadl.java.net 에서 wadl2java 클래스를 제공한다.

4. 결론

RESTful 웹서비스는 SOAP 이 가지고 있던 복잡함과 무겁다는 단점을 보완하여 등장한 아키텍처 모델이다. 하지만, 서비스에 대한 명세를 텍스트 문서에 의존해야 하므로 자동 코드 생성 및 서비스 공개에 대한 어려움이 있던 것이 사실이다. REST 에서 가장 아쉬웠던 서비스 명세에 대해 기계가 처리 가능하도록 명세를 기술하기 위한 WADL 이 Sun 에 의해 발표됨으로써 REST 의 부족한 부분이 보완이 되었지만, 아직 업계에서는 존재를 알지 못할 정도로 활용되지 않는 것이 현실이다. REST 라는 인터넷 환경에서 제공할 수 있는 최선의 아키텍처를 WADL 이라는 기술언어를 통하여 서비스의 명세를 체계화하고 개발에 이용될 수 있기를 바란다.

[참고문헌]

[1] 박유미외 4 명, “SOAP 기반 웹서비스와 RESTful 웹서비스 기술 비교, 전자통신동향분석 제 25 권 제 2 호 2010 년 4 월.

[2] 김창환, “웹서비스 개념 및 응용 서비스 동향,” 정보통신산업진흥원, 주간기술동향 1395 호, 2009. 5.

[3] Roy Fielding, “Architectural Styles and the Design of Network-based Software Architectures,” Dissertation of Doctor of Philosophy in Information and Computer Science, University of California, IRVINE, 2000.

[4] Marc J. Hadley "Web Application Description Language (WADL)", Sun Microsystems Inc, November 9, 2006