

Event and Venue Retrieval as a Web Service

A WCF implementation using Eventful API

4/5/2013

Rochester Institute of Technology : CS Department

Karan Moodbidri

Table of Contents

Introduction	3
Technologies	3
Newtonsoft Json.NET	3
Usage of API	4
Venues Search.....	4
Categories Search.....	4
Events Search	5
Class Structure	5
Category.cs.....	5
Categories.cs	6
Performer.cs	6
Event.cs	7
Event.cs	8
IService1.cs.....	8
Service1.svc.cs.....	8

Introduction

The Music Information and Song Retrieval Web Service has been realized using the Windows Communication Foundation (WCF) version .NET 4.5.

Path to Create a WCF project:

1. Open Visual Studio
2. Click File
3. Click New
4. Click Project
5. Select WCF Service Application from list of available projects for Visual C#
6. Give Name for Project Below

This is how you create the WCF project.

In the Solution Explorer in the right hand side of Visual Studio you will see the files generated for the project.

Technologies

The public API i.e. Eventful API was called using the URL call. This was achieved by using the 'WebClient' class of System.Net. The values returned from the public API were in XML. But then the XML was converted into JSON i.e. (JavaScript Object Notation) format. To parse the JSON, a publically available JSON extension for .NET framework i.e. JSON.Net was used. Classes were created to represents segments of the JSON and were parsed and filled in the objects respectively.

Newtonsoft Json.NET

JSON.Net is used to parse the JSON String. It can be achieved by using the JObject notation or by creating an JArray which contains individual sections of the JSON string.

```
using (WebClient wc = new WebClient())
```

```

{
    string json = wc.DownloadString(apiCall);
    var jArray = JObject.Parse(json);
}

```

OR it can be done in the following manner :

```

JObject obj = JObject.Parse(json);
while (temp < count)
{
    category = new Category();
    category.categoryID = (string)obj["categories"]["category"][temp]["id"];
    category.categoryName = (string)obj["categories"]["category"][temp]["name"];
    temp++;
}

```

Usage of API

Venues Search

http://api.eventful.com/rest/venues/search?app_key=AppKey&keywords=kw&location=lw&within=w

Input Parameters :

Keywords : The Search parameter for a keyword based search submitted by the user.

APIKey : The key required to call the API

location : Which location is the point of interest

within : Sets a geographical radius for the search

This API call will return the venues belonging to the location which is being searched by the user using the location option.

Categories Search

http://api.evdb.com/rest/categories/list?app_key=key

Input Parameters :

APIKey : The key required to call the API

This API call will return all the categories under which events can be subclassed.

Events Search

```
http://api.eventful.com/rest/events/search?app_key=AppKey&keywords=kw&location=lw&within=w
```

Input Parameters :

Keywords : The Search parameter for a keyword based search submitted by the user.

APIKey : The key required to call the API

location : Which location is the point of interest

within : Sets a geographical radius for the search

Category: The category to which the event belongs

Date : The date as a query parameter specifying the time around which

This API call will return the events belonging to the location which is being searched by the user using the location option

Class Structure

Category.cs

Fields :

1. categoryID
2. categoryName

Use of the class :

The class is used to fill the data returned from the public API into its fields .

Categories.cs

Fields :

1. List<Category >()

Use of the class :

The class is used to fill the data i.e. information about the category returned from the public API into the fields of a Category object and in turn added to this list. This Categories will be then serialized and passed across as a customized JSON or XML.

Performer.cs

Fields :

1. performerID
2. performerUrl
3. performerName
4. performerBio

Use of the class :

The class is used to fill the data i.e. metadata about a particular performer returned from the public API into its fields .

Event.cs

Fields :

1. eventId;
2. eventTitle;
3. eventUrl;
4. eventStartTime;
5. eventStopTime;
6. venueUrl;
7. venueName;
8. venueAddress;
9. eventCityName;
10. eventState;
11. eventStateAbbr;
12. eventPostalCode;
13. eventCountryName;
14. eventCountryAbbr;
15. eventLongitude;
16. eventLatitude;
17. eventImageUrl;
18. List<Performer> performers = new List<Performer>();

Use of the class :

The class is used to fill the data i.e. metadata about a particular event returned from the public API into its fields . Also the List<Performer> is used to hold the various Performer objects.

Event.cs

Fields :

2. List<Events>()

Use of the class :

The class is used to fill the data i.e. metadata about event returned from the public API into the fields of a Events object and in turn added to this list. This Event object will be then serialized and passed across as a customized JSON .

IService1.cs

Contains the OperationContracts i.e. the definition of the possible set of operations that can be performed using this web service.

Service1.svc.cs

Contains the implantations of the OperationContracts i.e. the implementations of the possible set of operations that can be performed using this web service.