

The background features a decorative graphic consisting of three overlapping blue circles of varying sizes, arranged in a diagonal line from the top right towards the bottom right. Two thin blue lines intersect at the top left, forming a large 'V' shape that frames the circles.

# Weather Web Service

A WCF Implementation using weather wunderground  
API

**AMEY GAWDE**  
Rochester Institute Of technology  
Rochester (NY)  
5/22/2013

## Contents

Introduction.....	3
Technology .....	3
Functions available .....	4
Snippets .....	5
Notes: .....	8
Publishing Web Service .....	8

## Introduction

Weather information retrieval Web service has been created using the Windows Communication Foundation (WCF) version.NET4.5.

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.

(For More information about how to create web service please follow the link)

<http://vhost3.cs.rit.edu/application/documetations/How%20to%20create%20web%20service.pdf>

## Technology

The weather wunderground API by is very useful to create different functionality for weather web service. This API is publicly available for free. Please find documentation about how to use API's please click on the link below.

<http://www.wunderground.com/weather/api/>

## Functions available

- **getWeather**

This function provides the weather information for given state and city.  
Function provides the summarized weather information.

Input parameters: city name in string  
State name in string

Output: weather information in Json format.

- **getWeather\_hourly**

This function provides the weather information for given state and city.  
Function provides the hourly weather information.

Input parameters: city name in string  
State name in string

Output: weather information in Json format.

- **getWeather\_tenDays**

This function provides the weather information for given state and city.  
Function provides the 10 days weather information.

Input parameters: city name in string  
State name in string

Output: weather information in Json format.

## Snippets

### Function getWeather

```
/**
 * Provides the weather report for given city and state
 *
 * @param    city_name    (string) it is city name
 * @param    state_name  (string) it is state name
 *
 * @return    json object
 */
public string getWeather(string city_name, string state_name)
{
    string output;
    using (WebClient client = new WebClient())
    {
        string input = "http://api.wunderground.com/api/" + key + "/conditions/q/" + state_name+"/"+city_name+".json";
        output = client.DownloadString(input);

        //JsonValue value = JsonValue.Parse(output);
        var jss = new JavaScriptSerializer();
        var dict = jss.Deserialize<Dictionary<string, dynamic>>(output);
        return (jss.Serialize(dict));
    }
}
```

## Function getWeather\_hourly

```
/**
 * Provides the hourly weather report for given city and state
 *
 * @param    city_name    (string) it is city name
 * @param    state_name   (string) it is state name
 *
 * @return    json object
 */

public string getWeather_hourly(string city_name, string state_name, Boolean hourly)
{
    if (hourly)
    {
        string output;
        using (WebClient client = new WebClient())
        {

            string input = "http://api.wunderground.com/api/" + key + "/hourly/q/" + state_name + "/" + city_name + ".json";
            output = client.DownloadString(input);

            //JsonValue value = JsonValue.Parse(output);
            var jss = new JavaScriptSerializer();
            var dict = jss.Deserialize<Dictionary<string, dynamic>>(output);
            return (jss.Serialize(dict));
        }
    }
    return null;
}
```

## Function getWeather\_tenDays

```
/**
 * Provides the 10 days weather report for given city and state
 *
 * @param    city_name    (string) it is city name
 * @param    state_name  (string) it is state name
 *
 * @return    json object
 */

public string getWeather_tenDays(string city_name, string state_name, Boolean tenday)
{
    if (tenday)
    {
        string output;
        using (WebClient client = new WebClient())
        {

            string input = "http://api.wunderground.com/api/" + key + "/forecast10day/q/" + state_name + "/" + city_name + ".json";
            output = client.DownloadString(input);

            //JsonValue value = JsonValue.Parse(output);
            var jss = new JavaScriptSerializer();
            var dict = jss.Deserialize<Dictionary<string, dynamic>>(output);
            return (jss.Serialize(dict));
        }
    }
    return null;
}
```

---

## Notes:

Please make sure you have API key.

Also include following at the top

```
using System.Text;  
using System.Web;  
using System.Web.Script.Serialization;  
using System.Net;
```

## Publishing Web Service

Please click on following link and follow the instruction

<http://vhost3.cs.rit.edu/application/documetations/Publish%20web%20service%20to%20IIS.pdf>