using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.ServiceModel;

using System.Text;

using System.ServiceModel.Web;

namespace WcfService3

{

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the interface name "IRestService1" in both code and config file together.

[ServiceContract]

public interface IRestService1

{

[OperationContract]

[WebInvoke(Method = "GET", ResponseFormat = WebMessageFormat.Json,BodyStyle=WebMessageBodyStyle.Wrapped, UriTemplate = "Addition/{a}/{b}")]

string Addition(string a,string b);

[OperationContract]

[WebInvoke(Method = "GET", ResponseFormat = WebMessageFormat.Json,BodyStyle=WebMessageBodyStyle.Wrapped, UriTemplate = "Subtraction/{a}/{b}")]

string Subtraction(string a,string b);

[WebInvoke(Method = "GET", ResponseFormat = WebMessageFormat.Json,BodyStyle=WebMessageBodyStyle.Wrapped, UriTemplate = "Multiplication/{a}/{b}")]

string Multiplication(string a,string b);

[WebInvoke(Method = "GET", ResponseFormat = WebMessageFormat.Json,BodyStyle=WebMessageBodyStyle.Wrapped,UriTemplate = "Division/{a}/{b}")]

string Division(string a,string b);

}

}

<%@ ServiceHost Language="C#" Debug="true" Service="WcfService3.RestService1" CodeBehind="RestService1.svc.cs"

Factory="System.ServiceModel.Activation.WebServiceHostFactory"

%>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.ServiceModel;

using System.Text;

namespace WcfService3

{

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name "RestService1" in code, svc and config file together.

public class RestService1 : IRestService1

{

public string Addition(string a, string b)

{

int a1 = Convert.ToInt32(a);

int a2 = Convert.ToInt32(b);

int a3 = a1 + a2;

return a3.ToString();

}

public string Subtraction(string a, string b)

{

int a1 = Convert.ToInt32(a);

int a2 = Convert.ToInt32(b);

int a3 = a1 - a2;

return a3.ToString();

}

public string Multiplication(string a, string b)

{

int a1 = Convert.ToInt32(a);

int a2 = Convert.ToInt32(b);

int a3 = a1 \* a2;

return a3.ToString();

}

public string Division(string a, string b)

{

int a1 = Convert.ToInt32(a);

int a2 = Convert.ToInt32(b);

int a3 = a1 + a2;

return a3.ToString();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.ServiceModel;

using System.Text;

using System.ServiceModel.Web;

namespace WcfService3

{

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the interface name "IRestService1" in both code and config file together.

[ServiceContract]

public interface IRestService1

{

[OperationContract]

[WebInvoke(Method = "GET", ResponseFormat = WebMessageFormat.Json,BodyStyle=WebMessageBodyStyle.Wrapped, UriTemplate = "Addition/{a}/{b}")]

string Addition(string a,string b);

[OperationContract]

[WebInvoke(Method = "GET", ResponseFormat = WebMessageFormat.Json,BodyStyle=WebMessageBodyStyle.Wrapped, UriTemplate = "Subtraction/{a}/{b}")]

string Subtraction(string a,string b);

[WebInvoke(Method = "GET", ResponseFormat = WebMessageFormat.Json,BodyStyle=WebMessageBodyStyle.Wrapped, UriTemplate = "Multiplication/{a}/{b}")]

string Multiplication(string a,string b);

[WebInvoke(Method = "GET", ResponseFormat = WebMessageFormat.Json,BodyStyle=WebMessageBodyStyle.Wrapped,UriTemplate = "Division/{a}/{b}")]

string Division(string a,string b);

}

}