

Investigación Técnica 1

“First Steps Visual Studio with C#”

Derek Jimenez Quesada

Jose Alejandro Gomez Castro

Jose Andres Ramirez Ocampo

Jasson Calderón Villalobos

Introducción

En esta pequeña investigación se busca ampliar los conocimientos tanto lógicos como técnicos acerca de los Servicios Web y las comunicaciones entre cliente y servidor en la herramienta Visual Studios, los temas que serán investigados en este documento serán: JSON, REST y WFC. Y generando de estas 3 tecnologías una pequeña aplicación similar a una calculadora que pueda sumar, restar, multiplicar y dividir dos números.

Temas

1. JSON
   1. ¿Que es?

JSON (JavaScript Object Notation) es un formato para el intercambios de datos, básicamente JSON describe los datos con una sintaxis dedicada que se usa para identificar y gestionar los datos. JSON nació como una alternativa a XML, el fácil uso en javascript ha generado un gran número de seguidores de esta alternativa. Una de las mayores ventajas que tiene el uso de JSON es que puede ser leído por cualquier lenguaje de programación. Por lo tanto, puede ser usado para el intercambio de información entre distintas tecnologías.

* 1. ¿Para que se utiliza?

Para guardar e intercambiar información en texto plano. Una de las mayores ventajas que tiene el uso de JSON es que puede ser leído por cualquier lenguaje de programación. Por lo tanto, puede ser usado para el intercambio de información entre distintas tecnologías.

* 1. Tipo de datos:
     1. Números: permite también números negativos y fraccionarios.
     2. Cadenas: permite uno o más caracteres y entre comillas “”.
     3. Booleanos: permite banderas de true o false.
     4. Null: permite valores nulos.
     5. Vector: permite listas ordenadas con uno o más valores de cualquier tipo, se separan por comas y cada conjunto de valores se agrupa dentro de corchetes.
     6. Objetos: permite listas no ordenadas de valores en pares con el formato nombre: valor, separados por coma y metidos en llaves. El nombre tiene que ser una cadena, y el valor puede ser cualquier tipo.
  2. Estructuras:
     1. Un conjunto de pares (clave,valor) encerrado entre los caracteres “{” y “}”, separando la clave del valor por el símbolo “:”, y separando cada par del siguiente con el carácter “,”.
     2. Un conjunto ordenado de valores encerrado entre los caracteres “[” y “]”, y separando cada valor del siguiente con el carácter “,”.

{

"nombre" : "Raúl",

"fecha\_nacimiento": "1995-17-08",

"Aficiones": ["Música", "Cine", "Tenis"],

"Residencia": "Madrid"

}

1. REST
   1. ¿Que es ?

REST es el acrónimo de Representational State Transfer (o Transferencia de estado representacional), es un estilo de arquitectura de software que puede ser usado con muchos tipos de formatos de mensajes. Los servicios que siguen los estándares de REST son llamados servicios RESTful. Tiene como visión brindar:

- Escalabilidad: un sistema basado en REST debería trabajar bien con muchos clientes y muchas transacciones.

- Generalización de interfaces: que significa que el sistema puede ser adaptado a diferentes procesos de negocios.

- Despliegue independiente de los componentes.

En REST la comunicación entre el cliente y el servidor no tiene que ser bajo el mismo sistema operativo o mismo lenguaje de programación y este puede depender de componentes intermediarios.

* 1. ¿Como se utiliza?

i. Se crea un ASP.NET Web API y en sus “Controllers” en el archivo “ValuesController.cs” se genera la ruta y el metodo por el cual se le va a pasar los datos. Tambien aca se procesa la operación que va a ser ejecutada con el/los dato(s) que se le pasaron.

ii. Se crea una comunicación entre lo que va a funcionar como cliente y el servicio por medio de clases como HttpWebRequest, WebClient, Ajax, etc. En esta se va a definir el tipo de dato que se le envía, la ruta, y el tipo de método.

iii. El cliente invoca la ruta y es quien procesa la clase enviando datos en formato JSON en el caso de enviar objetos. También es quien recibe la respuesta de la conexión e imprime el resultado.

c. Métodos REST y su relación con CRUD:

i. Create = PUT Con un nuevo URI / POST Con un URI base que retorna un nuevo URI.

ii. Read = GET.

iii. Update = PUT Con un URI ya existente.

iv. Delete = DELETE.

URI = Es el acrónimo para Uniform Resource Identifier (o Identificador de recursos uniforme) que es una cadena de caracteres que identifica un recurso en la red de forma que este no pueda equivocarse ni confundirse. El URI se compone de las siguientes partes: Esquema, Autoridad, Ruta, Consulta, Fragmento.

1. WCF
   1. ¿Que es?

Windows Communication Foundation (WCF) es un marco de trabajo para la creación de aplicaciones orientadas a servicios. Con WCF, es posible enviar datos como mensajes asincrónicos de un extremo de servicio a otro. Un extremo de servicio puede formar parte de un servicio disponible continuamente hospedado por IIS, o puede ser un servicio hospedado en una aplicación. Un extremo puede ser un cliente de un servicio que solicita datos de un extremo de servicio. Los mensajes pueden ser tan simples como un carácter o una palabra que se envía como XML, o tan complejos como una secuencia de datos binarios.

WCF es una plataforma para la construcción distribuida e interoperable (una aplicación distribuida se le conoce también como sistemas conectados, es cuando una aplicación en la que parte de ella se ejecutan 2 o más nodos de computación).

No se limita solo a los Web Service(WS), sino que a todo tipo de servicios distribuidos de la plataforma **.net**.

* 1. ¿Como se utiliza? (Solo un Binding)
     1. Se crea un WCF Service y se referencia en el cliente.
     2. Un extremo de servicio puede formar parte de un servicio disponible continuamente hospedado por IIS, o puede ser un servicio hospedado en una aplicación.
     3. Realizar peticiones en internet o dentro de una red (el cliente es quien hace una petición y recibe una respuesta).
  2. Ventajas:
     1. Seguridad.
     2. Calidad.
     3. Escalabilidad.
     4. Robustez.
     5. Portabilidad.
  3. Ejemplos:
     1. Un servicio seguro para procesar transacciones comerciales.
     2. Un servicio que proporciona datos actualizados a otras personas, como un informe sobre tráfico u otro servicio de supervisión.
     3. Un servicio de chat que permite a dos personas comunicarse o intercambiar datos en tiempo real.
     4. Una aplicación de panel que sondea los datos de uno o varios servicios y los muestra en una presentación lógica.
     5. Exponer un flujo de trabajo implementado utilizando Windows Workflow Foundation como un servicio WCF.
     6. Una aplicación de Silverlight para sondear un servicio en busca de las fuentes de datos más recientes.

Enlace al video tutorial (REST):

<https://www.youtube.com/watch?v=TAAlk975FGQ>

Código fuente servicio REST:

https://mega.nz/#!8IQFWLJD

Código fuente servicio WCF:

https://mega.nz/#!QZ41FAIR

Bibliografía

* <http://www.c-sharpcorner.com/UploadFile/dhananjaycoder/four-steps-to-create-first-wcf-service-for-beginners/>
* <https://msdn.microsoft.com/es-es/library/ms731082(v=vs.110).aspx>
* <https://www.dotnetfoundation.org/blog/wcf-is-open-source>
* <http://www.lynda.com/OData-tutorials/Foundations-Programming-Web-Services/126131-2.html>
* http://johnnycode.com/2012/02/23/consuming-your-own-asp-net-web-api-rest-service/
* http://stackoverflow.com/questions/9145667/how-to-post-json-to-the-server
* <http://stackoverflow.com/questions/24843264/how-to-add-custom-methods-to-asp-net-webapi-controller>
* http://stackoverflow.com/questions/6201529/turn-c-sharp-object-into-a-json-string-in-net-4