

IN6AV

SubastAutos.com

Manual de Técnico

Una referencia a el programador.

Josué Alexander Baquix Baten.
Manual Técnico.
IN6AV





Josué Alexander Baquiaux Baten

Proyecto 2Unidad. Silverlight.

INTRODUCCION

Acá se le detallara el uso del remoting, utilizado en el sistema SubastAutos.com. El cual esta desarrollado en WCF ,utilizando PollingDuplex.



Josué Alexander Baquiaux Baten

Proyecto 2Unidad. Silverlight.

OBJETIVOS

Prevenir perdida de tiempo, en un futuro, pues el sistema puede ser modificado.

- Prevención de errores.
- Ayuda a aclarar el código.



Manual De Usuario

Un servicio WCF, puede tener muchas formas de comunicarse con un cliente, y una forma es utilizando PollingDuplex, o comunicación Cliente-Servidor.

```
using System;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Activation;
using System.Collections.Generic;
using MySql.Data.MySqlClient;
using SubastaAutos;

namespace SubastaAutos.Web
{
    //Defino una clase que servirá, como Callback ,para un
    contrato que lo aplique
    [ServiceContract]
    public interface IRemotingCallback
    {
        //Este es un método que se podrá ejecutan en el cliente,
        desde el servidor.
        [OperationContract(IsOneWay = true)]
        void ActualizarListaDeAutosParaSubasta(ModeloDeDatos.Subasta s);
    }

    //Contrato que usara el servidor, y que utiliza el metodo
    binding PollingDuplex

    [ServiceContract(Namespace = "", CallbackContract = typeof(IRemotingCallback))]
    [AspNetCompatibilityRequirements(RequirementsMode =
    AspNetCompatibilityRequirementsMode.Allowed)]
    [ServiceBehavior (InstanceContextMode = InstanceContextMode.Single)]
    public class Remoting
    {
        //Esta es una lista segura, que retendra los sockets con el
        cliente.

        SynchronizedCollection<IRemotingCallback> clientes = new
        SynchronizedCollection<IRemotingCallback>();
    }
}
```



//Método que ejecuta el cliente, para indicar que ha habido un cambio.

```
[OperationContract]
public void Reportar(int idSubasta, int idVehiculo, String login, String
fechaPropuesta, Double montoSugerido)
{
```

//Se ejecuta un método, que extrae información de la base de datos y se almacena en un variable.

```
ModeloDeDatos.Subasta sb = ModificarSubasta(idSubasta, idVehiculo,
login, fechaPropuesta, montoSugerido);
```

//Se recorren todos los canales, y se les ejecuta el metodo (CALLBACK) ActualizarListaDeAutosParaSubata(); y se le pasa como parametro los datos extraidos de la base de datos, claro para no tener mucho trafico hacia el servidor de DB.

```
foreach (IRemotingCallback cliente in clientes)
{
    try
    {
        cliente.ActualizarListaDeAutosParaSubasta(sb);
    }
    catch (Exception ex)
    {
        //this.clientes.Remove(cliente);
    }
}

}
[OperationContract(IsOneWay = true)]
public void Salir() {
    clientes.Remove(CurrentCallback);
}
public IRemotingCallback CurrentCallback
{
    get
    {
        return
OperationContext.Current.GetCallbackChannel<IRemotingCallback>();
    }
}
}
```

//Es uno de los metodos mas importantes,pues en el se registra un cliente.

```
[OperationContract]
public void Registrarse()
{
    this.clientes.Add(CurrentCallback);
}
```



```
[OperationContract]
public ModeloDeDatos.Subasta ModificarSubasta(int idSubasta, int idVehiculo,
String login, String fechaPropuesta, Double montoSugerido)
{
    SubastaAutos.DbSubASTAs conexionBd = new SubastaAutos.DbSubASTAs(new
SqlConnection("Database=dbsubastas;Data Source=localhost;User
Id=root;Password="));
    bool resultadoCorrecto = true;
    try
    {
        SubAsTA miSubasta = (from subastas in conexionBd.SubAsTA
                               where subastas.IDSubasta == idSubasta
                               select subastas).Single<SubAsTA>();
        miSubasta.IDVehiculo = idVehiculo;
        miSubasta.LogGIN = login;
        miSubasta.FechaPropuesta = fechaPropuesta;
        miSubasta.MontoSugerido = montoSugerido;
        conexionBd.SubmitChanges();
    }
    catch (Exception e)
    {
        resultadoCorrecto = false;
    }
    return VehiculosDeSubastaActual();
}
```

//Método que extrae información de la base de datos, la cual es retornada al método Report();

```
[OperationContract]
public ModeloDeDatos.Subasta VehiculosDeSubastaActual()
{
    SubastaAutos.DbSubASTAs conexionBd = new SubastaAutos.DbSubASTAs(new
SqlConnection("Database=dbsubastas;Data Source=localhost;User
Id=root;Password="));
    ModeloDeDatos.Subasta subastaNow = null;
    try
    {
        subastaNow = (from sub in conexionBd.SubAsTA
                       select new ModeloDeDatos.Subasta
                       {
                           Estado = sub.EstAdo,
                           fechaPropuesta = sub.FechaPropuesta,
                           hora = sub.HoRa,
                           IdSubasta = sub.IDSubasta,
                           IdVehiculo = sub.IDVehiculo,
                           LogginDueño = sub.LogGIN,
                           PrecioSugerido = sub.MontoSugerido
                       }).Single<ModeloDeDatos.Subasta>();

        if (subastaNow != null)
        {
            List<ModeloDeDatos.Vehiculo> lista = new
List<ModeloDeDatos.Vehiculo>();
            var vehiculos = from vehi in conexionBd.VEHICULO
```



```

    && vehi.IDEstado < 3

    join subasta in conexionBd.SubAsTA on
    vehi.IDVehiculo equals subasta.IDVehiculo

    orderby vehi.PLaCa
    where subasta.IDSubasta == subastaNow.IdSubasta

    select new
    {
        IDVehiculo = vehi.IDVehiculo,
        PLaCa = vehi.PLaCa,
        IDCombustible = vehi.IDCombustible,
        MarCa = vehi.MarCa,
        AÑO = vehi.AÑO,
        ModelO = vehi.ModelO,
        KiloMetRaJE = vehi.KiloMetRaJE,
        Color = vehi.Color,
        IDTipoVehiculo = vehi.IDTipoVehiculo,
        PrecioBase = subasta.MontoSugerido,
        IDEstado = vehi.IDEstado
    };
    foreach (var v in vehiculos)
    {
        lista.Add(new ModeloDeDatos.Vehiculo
        {
            idVehiculo = v.IDVehiculo,
            placa = v.PLaCa,
            idCombustible = v.IDCombustible,
            marca = v.MarCa,
            año = v.AÑO,
            modelo = v.ModelO,
            kilometraje = v.KiloMetRaJE,
            color = v.Color,
            idTipoVehiculo = v.IDTipoVehiculo,
            precioBase = v.PrecioBase,
            idEstado = v.IDEstado
        });
        subastaNow.VehiculosDeSubasta = lista;
    }
    }
    catch (Exception ex) { }

    return subastaNow;
}
}
```



Josué Alexander Baquiaux Baten

Proyecto 2Unidad. Silverlight.

CONCLUSION

Toda tecnología, tiene algo de complicado. Pero leyendo, se pueden resolver las dudas.



Josué Alexander Baquiaux Baten

Proyecto 2Unidad. Silverlight.

BIBLIOGRAFIA

<http://www.microsoft.com/silverlight/>

Migrating .NET Remoting to WCF (and even ASMX!)