WCF

What do you know about WCF?

The most important piece of WCF is the endpoint.

Endpoint

Endpoint defines the interaction between a WCF service host and client. It consists of 3 essential components.

1. Address: Where a client can locate the service.
2. Contract: What a client can expect from the service.
3. Binding: Rules of the interaction between the client and the host.

Address

Address is where a client can find the service.

|  |  |
| --- | --- |
| Note: | The address can be changed if the contract is met by some other host. |

Contract

The contract defines exactly what a client can expect from the service.

Why do we need a contract when we have the address to the service?

We did not have this contract in the .asmx web services. There are 2 reasons to introduce this.

On the host side, having the contract takes away the responsibilities of contract definition from the developer and puts it in the lap of a designer or, if required, an architect. A developer need not put the [WebService] and [WebMethod] attributes and decide how such classes and methods will be exposed by setting the properties of those attributes. He or she can concentrate on making the class perfect. The designer or the architect then exposes the class as a [ServiceContract] (interface) with its [OperationContract]s (methods).

On the client side, there is a reason to have the contract as well, though a weak one. On the client side, the contract has to be exposed as some classes. Just as in the .asmx web service client, the WCF client has a proxy which is derived from this contract or interface, through which the client can interact with the host. The proxy is made as .wsdl, .xsd and the .cs or .vb files. Having this contract helps the runtime environment to connect the endpoint with this proxy right away.

Binding

This defines the rules of interaction between the host and the client. It consists of many parameters under different groups like:

1. Timeout
2. Encoding
3. Buffering
4. Security Credentials

Why binding?

In the.asmx web service world, everything was sent as HTTP. In the WCF world, the binding helps us in defining different protocols (including TCP, UDP and custom) and different parameters as mentioned above.

Also, these parameters can differ even for the contracts provided by the same service. For example, one service contract can have Text encoding, the other can have binary or MTOM encoding, even when they use the same underlying service.

Earlier, all the parameters had to be set in the service class but now this can be done using the configuration file of both the host and the client. (Of course, the more restrictive parameter of the two shall prevail).

DataContractSerializer

The .asmx web services used to be serialized using the XmlSerializer. You can learn more about this from the Serialization lecture. The WCF services are serialized using the DataContractSerializer. It carries a small disadvantage that we don’t have a lot of control over the serialization of a member. But this is compensated by big advantages. The very first is that this DataContractSerializer shows 10% more efficiency (may be because of standardization). The other is that we have a complete decoupling with the programming paradigm, that is, we can define what should be serialized, including the fields and the private members, something which could not be done by the XmlSerializer.

|  |  |
| --- | --- |
| Note: | The DataContractSerializer address can be changed if the contract is met by some other host. |

Properties

Generally, we have the WCF model which is per call. This can also be seen as calls to static methods only. So, even the properties have to be exposed as [OperationContract]s.

DataContract

If instances of a user-defined class are passed as parameters or return values, then they have to be exposed as [DataContract]s.

WCF Data Services

WCF Data Services help expose the database or any class which has one or more IQueryable members (which will correspond to the database tables).

ReST

ReST stands for Representational State Transfer. It is nothing but data returned in an XML format (just like SOAP). The advantage of ReST over SOAP is that we can query ReST with query strings. We can use some keywords to query the data like an ordinary database, for example:

$filter=PRICE lt 20&OrderBy=ID&$skip=50&$top=10

Here are the descriptions of these keywords:

|  |  |
| --- | --- |
| Keyword | Description |
| $expand | Directs that related records should be retrieved in the record or collection being retrieved. |
| $filter | Specifies an expression or function that must evaluate to ‘true’ for a record to be returned in the collection. |
| $orderby | Determines what values are used to order a collection of records. |
| $select | Specifies a sub set of properties to return. |
| $skip | Sets the number of records to skip before it retrieves records in a collection. |
| $top | Determines the maximum number of records to return. |

We can even have functions for WHERE ($filter) clauses which can be seen here:

<http://msdn.microsoft.com/en-us/library/gg309461.aspx>

References

http://msdn.microsoft.com/en-us/library/gg309461.aspx