

WCF 의 Contract1

WCF 의 기본 중에서 가장 먼저 알려드리고 싶어 꺼내 든 주제는 **Contract** 입니다.

Contract 의 사전적 의미는,, "**계약**" 이죠.

잘 아시겠지만 계약은 약속이랑 비슷하지만 약속 보다는 좀 더 강력한 의미로 쓰입니다.

다시, 본론으로 돌아와서,, 그럼, WCF 에서의 Contract 는 무엇을 말하는 걸까요?

네, 다들 예상 하셨겠지만, WCF 에서의 Contract 역시 사전적 의미와 비슷하게 쓰이며, WCF 서비스와 클라이언트 사이에 어떤 계약을 정의할 때 사용하는 것을 말합니다.

좀 더 자세하게 설명을 하자면, 이 Contract 라는 것은 **서비스와 클라이언트가 서로 통신할 때 사용하는 메시지의 명세(specification)를 정의하는 것**을 말합니다.

WCF 는 총 세가지 타입의 Contract 가 있으며, 다음은 이 타입들에 대한 정의입니다.

- **Service contract** : 서비스에 의해 구현되는 기능들에 대해 설명하며, 서비스 계약(service contract)으로 정의된 .NET 타입의 클래스는 WSDL 의 services, port types 의 엘리먼트로 매핑됩니다. 서비스 계약과 함께 Operation contract는 서비스 계약 내에서 정의되며, 서비스의 동작(operation)을 설명합니다.
- **Data contract** : 서비스가 통신(communication)을 하는데 사용하는 데이터 구조를 나타냅니다. 이는 CLR 타입을 XSD(XML Schema Definitions) 로 매핑해주는 역할을 수행하는데, WCF 가 통신할 때 사용되는 데이터들을 어떻게 직렬화(serialization) 또는 역직렬화(deserialization)를 수행하는지를 설명합니다.
- **Message contract** : 메시지 계약은 CLR 타입을 SOAP 메시지로 매핑해주며, SOAP 메시지의 포맷을 설명합니다. 메시지 계약은 SOAP 헤더에서 바디까지 컨트롤할 수 있게 해줍니다.

WCF 서비스는 많은 시스템과의 상호 운용성(interoperability)을 높이기 위해 WSDL 을 사용 합니다.

위에서 설명한 세 가지의 Contract 들은 **WCF 서비스나 서비스에서 사용하는 여러 데이터들을 WSDL 또는 SOAP 의 요소들로 매핑 시키는 역할을 수행**합니다.

이 말은 곧, 서비스와 통신하기 위한 메시지의 포맷을 정의한다는 말이기도 하지요.

그럼, 이러한 Contract 들이 어떻게 쓰이는지 하나씩 살펴보겠습니다.

우선, Service Contract 와 Operation Contract 에 대해 알아보구요, 한 두번에 걸쳐서 Data Contract 와 Message Contract 에 대해서도 알아보도록 하겠습니다.

1. ServiceContract & Operation Contract

다시 한번 더 얘기하자면, 서비스 계약은 **서비스가 제공하는 여러 기능(동작)들을 정의하는 인터페이스이며, 서비스 자체를 정의**한다고 생각하면 될 듯 합니다.

그리고, 서비스에서 제공하는 기능들을 외부로 노출하기 위해 사용하는 Operation Contract 가 있습니다. 이는 Service Contract 내부에서 정의됩니다.

```
[ServiceContract]
interface IProductService
{
    [OperationContract]
    string GetFirstName(string empID);
}
```

만들었던 첫 WCF 서비스의 코드 중 일부입니다.

서비스는 ServiceContract 특성을 인터페이스에서 선언하고, 그 메소드에 OperationContract 특성을 선언해줌으로써, 서비스 계약을 정의해주었습니다. 기억하시겠지만, 실제 구체적인 서비스의 기능은 Service Contract 가 선언된 인터페이스를 상속 받아 구현해야 했었죠.

그럼, 이렇게 서비스가 정의됐을 때 WSDL 이 어떻게 만들어지는지 살펴보겠습니다.

우리가 작성했었던 첫 WCF 서비스를 동작시킨 상태에서 웹 브라우저를 이용해 <http://localhost:8000/ProductService?wsdl> 로 접근해 보면 다음과 같은 화면을 확인할 수 있습니다.

```
<?xml version="1.0" encoding="utf-8" ?>
- <wsdl:definitions name="ProductService" targetNamespace="http://tempuri.org/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:tns="http://tempuri.org/"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy"
  xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
  xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract"
  xmlns:wsa10="http://www.w3.org/2005/08/addressing"
  xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
  xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata">
- <wsdl:types>
  - <xsd:schema targetNamespace="http://tempuri.org/Imports">
    <xsd:import schemaLocation="http://localhost:8000/ProductService?xsd=xsd0"
      namespace="http://tempuri.org/" />
  -
```

이것이 바로 WCF 서비스가 제공하는 WSDL 입니다.

앞에서 언급했듯이 **Service Contract 로 정의 되어진 정보는 WSDL 의 "service", "port type" 엘리먼트로 매핑** 된다고 하였습니다. 또한 **Operation Contract 로 정의된 부분은 "operation" 엘리먼트로 매핑** 된다고 했었죠,, 이것 역시 다음과 같이 확인할 수 있었습니다.

```
- <wsdl:portType name="IProductService">
- <wsdl:operation name="GetFirstName">
  <wsdl:input wsaw:Action="http://tempuri.org/IProductService/GetFirstName"
    message="tns:IProductService_GetFirstName_InputMessage" />
  <wsdl:output wsaw:Action="http://tempuri.org/IProductService/GetFirstNameResponse"
    message="tns:IProductService_GetFirstName_OutputMessage" />
</wsdl:operation>
</wsdl:portType>
```

```

- <wsdl:service name="ProductService">
- <wsdl:port name="BasicHttpBinding_IProductService"
  binding="tns:BasicHttpBinding_IProductService">
  <soap:address location="http://localhost:8000/ProductService" />
</wsdl:port>
</wsdl:service>

```

이렇게 WCF 를 이용하여 만들어진 서비스는 WSDL 로 제공되어지며, 클라이언트에서 서비스를 사용할 수 있게 되는 것입니다.

그리고, ServiceContract 특성 클래스에는 **Namespace 속성**을 제공합니다. 이 속성은 WSDL 과 SOAP 메시지의 Namespace 의 값을 명시적으로 설정할 수 있게 합니다.

다음과 같이 기존의 서비스를 조금 수정해 보았습니다.

```

[ServiceContract(Namespace="http://RuAAService.co.kr/")]
interface IProductService
{
    [OperationContract]
    string GetFirstName(string empID);
}

```

그리고, 다시 이 서비스의 WSDL 을 확인해보면, 서비스의 네임스페이스가 다음과 같이 바뀌어져있는 것을 확인할 수 있습니다.

```

<wsdl:import namespace="http://RuAAService.co.kr/"
  location="http://localhost:8000/ProductService?wsdl=wsdl0" />

```